NATO Science for Peace and Security Series
D: Information and Communication Security - Vol. 19

Mining Massive Data Sets for Security

Advances in Data Mining, Search, Social Networks and Text Mining, and their Applications to Security

Edited by
Françoise Fogelman-Soulié
Domenico Perrotta
Jakub Piskorski
Ralf Steinberger

IOS
Press

NATO
OTAN

This publication is supported by: The NATO Science for Peace and Security Programme

MINING MASSIVE DATA SETS FOR SECURITY

NATO Science for Peace and Security Series

This Series presents the results of scientific meetings supported under the NATO Programme: Science for Peace and Security (SPS).

The NATO SPS Programme supports meetings in the following Key Priority areas: (1) Defence Against Terrorism; (2) Countering other Threats to Security and (3) NATO, Partner and Mediterranean Dialogue Country Priorities. The types of meeting supported are generally "Advanced Study Institutes" and "Advanced Research Workshops". The NATO SPS Series collects together the results of these meetings. The meetings are co-organized by scientists from NATO countries and scientists from NATO's "Partner" or "Mediterranean Dialogue" countries. The observations and recommendations made at the meetings, as well as the contents of the volumes in the Series, reflect those of participants and contributors only; they should not necessarily be regarded as reflecting NATO views or policy.

Advanced Study Institutes (ASI) are high-level tutorial courses to convey the latest developments in a subject to an advanced-level audience.

Advanced Research Workshops (ARW) are expert meetings where an intense but informal exchange of views at the frontiers of a subject aims at identifying directions for future action.

Following a transformation of the programme in 2006 the Series has been re-named and re-organised. Recent volumes on topics not related to security, which result from meetings supported under the programme earlier, may be found in the NATO Science Series.

The Series is published by IOS Press, Amsterdam, and Springer Science and Business Media, Dordrecht, in conjunction with the NATO Public Diplomacy Division.

Sub-Series

A.	Chemistry and Biology	Springer Science and Business Media
B.	Physics and Biophysics	Springer Science and Business Media
C.	Environmental Security	Springer Science and Business Media
D.	Information and Communication Security	IOS Press
E.	Human and Societal Dynamics	IOS Press

http://www.nato.int/science
http://www.springer.com
http://www.iospress.nl

Sub-Series D: Information and Communication Security – Vol. 19 ISSN 1874-6268

Mining Massive Data Sets for Security

Advances in Data Mining, Search, Social Networks and Text Mining, and their Applications to Security

Edited by

Françoise Fogelman-Soulié
KXEN

Domenico Perrotta
European Commission – Joint Research Centre

Jakub Piskorski
European Commission – Joint Research Centre

and

Ralf Steinberger
European Commission – Joint Research Centre

IOS
Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

Published in cooperation with NATO Public Diplomacy Division

Proceedings of the NATO Advanced Study Institute on Mining Massive Data Sets for Security,
Gazzada (Varese), Italy
10–21 September 2007

ISBN 978-1-58603-898-4
Library of Congress Control Number: 2008935667

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

LEGAL NOTICE
The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

Mining Massive Data Sets for Security

The ever growing flood of data arises from many different sources: huge databases store increasing amounts of data on customers, sales, expenses, traveling, credit card usage, tax payments and the like; the Web gives us access to basically unlimited information in the form of text, images, videos and sound. Everything we do leaves an *information trail* which can potentially be stored and used. Most of the times, the uses are beneficial (providing us with a list of our expenses, information on our next trip or the delivery of some book we bought on the Web …), but sometimes this information can also be exploited by rogues to put us at risk (identity theft, credit card fraud, intrusion on a sensitive computer network, terrorism …). Information security breach is thus becoming an every-day threat for the citizens of modern societies and – when linked to terrorism – it is becoming more and more dangerous. However, discriminating good uses from bad ones is not easy if we want to maintain our standards of democracy and individual freedom. We therefore need to develop different – both efficient and non-invasive – security applications that can be deployed across a wide range of activities in order to deter and to detect the bad uses. The main research challenges of such security applications are: to *gather and share large amounts of data* or even just sample them when the volume is too big (data streams), to *fuse data* from different origins (such as numerical, text), to *extract the relevant information* in the correct context, to develop *effective user interfaces* from which people can obtain quick interpretations and security-alerts, and to *preserve people's privacy*.

All security issues share common traits: one has to handle enormous amounts of data (*massive data sets*), heterogeneous in nature, and one has to use a variety of techniques to store and analyze this data to achieve security. This requires an interdisciplinary approach, combining techniques from computer science, machine learning and statistics, computational linguistics, Web search, social networks and aggregation techniques to present results easily interpretable by the user. Academic research in all of the above areas has made great progress in recent years, while commercial applications now exist all around us (Amazon, eBay, Google, Yahoo!, …). The real power for security applications will come from the synergy of academic and commercial research focusing on the specific issue of security.

Special constraints apply to this domain, which are not always taken into consideration by academic research, but are critical for successful security applications:

- *Large volumes*: techniques must be able to handle huge amounts of data, and perform 'on-line' computation;
- *Scalability:* algorithms must have processing times that scale well with ever growing volumes;
- *Automation:* the analysis process must be automated so that information extraction can 'run on its own';
- *Ease of use*: every-day citizens should be able to extract and assess the necessary information;
- *Robustness*: systems must be able to cope with data of poor quality (missing or erroneous data).

The NATO Advanced Study Institute (ASI) on *Mining Massive Data Sets for Security*, held in Villa Cagnola, Gazzada, Varese (Italy) from 10 to 21 September 2007, brought together around 90 participants to discuss these issues. The scientific program consisted of invited lectures, oral presentations and posters from participants. The present volume includes the most important contributions, but can of course not entirely reflect the lively interactions which allowed the participants to exchange their views and share their experience.

The book is organized along the five themes of the workshop, providing both introductory reviews and state-of-the-art contributions, thus allowing the reader a comprehensive view of each of the themes. The bridge between academic methods and industrial constraints is systematically discussed throughout. This volume will thus serve as a reference book for anyone interested in understanding the techniques for handling very large data sets and how to apply them in conjunction for solving security issues.

Section 1 on *Data Mining* brings together contributions around algorithms for learning large data sets. Section 2 on *Search* highlights the problems of scale and threats of the web. Section 3 on *Social Networks* presents the theoretical tools and various issues around very large network structures. Section 4 on *Text Mining* focuses on techniques to extract structured information from multilingual and very large text collections. Finally, Section 5 presents various *applications* of the mentioned techniques to security: fraud, money laundering, intelligence, terrorism, geolocalization, intrusion.

The ASI event and the publication of this book have been co-funded by the NATO Program ‘Science for Peace and Security’, by the European Commission’s (EC) Enlargement Program and the EC-funded PASCAL Network of Excellence. All lectures have been filmed and are now freely accessible via the website http://videolectures.net/.

The NATO ASI was made possible through the efforts of the four co-Directors Clive Best (JRC, Ispra, Italy), Françoise Fogelman-Soulié (Kxen, France), Patrick Gallinari (University of Paris 6 – LIP6, France) and Naftali Tishby (Hebrew University, Israel), as well as of the other members of the program committee: Léon Bottou (NEC Labs America, USA), Lee Giles (Pennsylvania State University, USA), Domenico Perrotta (JRC, Ispra, Italy), Jakub Piskorski (JRC, Ispra, Italy) and Ralf Steinberger (JRC, Ispra, Italy). We would like to thank the additional reviewers, whose valuable feedback helped to improve the quality of this book: Spyros Arsenis, Thierry Artières, Carlo Ferigato, Nicholas Heard, Hristo Tanev, Cristina Versino, Dennis Wilkinson and Roman Yangarber. Our special thanks go to Sabine Gross (Public Relations, JRC Ispra) for her dedication and the endless hours she spent on organizing the ASI. Without her, the event would not have taken place.

June 2008 Clive Best and Françoise Fogelman-Soulié
Domenico Perrotta, Jakub Piskorski and Ralf Steinberger

Contents

Data Mining

Search

Social Networks

Text Mining

Security Applications

This page intentionally left blank

Data Mining

This page intentionally left blank

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-3

Learning using hidden information: Master-class learning

Vladimir Vapnik[a], Akshay Vashist[a] and Natalya Pavlovitch[b]
[a] *4 Independence Way, Princeton, NJ 08540 USA*
[b] *Institute of Russian Language, Volhonka 18/2, Moscow 121019 Russia*

Abstract. The classical setting of the supervised learning problem is to learn a decision rule from labeled data where data is points in some space $\mathcal{X}$ and labels are in $\{+1, -1\}$. In this paper we consider an extension of this supervised learning setting: given training vectors in space $\mathcal{X}$ along with labels and description of this data in another space $\mathcal{X}^*$, find in space $\mathcal{X}$ a decision rule better than the one found in the classical setting [1]. Thus, in this setting we use two spaces for describing the training data but the test data is given only in the space $\mathcal{X}$. In this paper, using SVM type algorithms, we demonstrate the potential advantage of the new setting.

Keywords. Kernel methods, SVM, Hidden Information, Learning in multiple spaces and SVM+, master-class learning.

Introduction

In the classical supervised learning paradigm, training examples are represented by vectors of attributes, an oracle (teacher) supplies labels for each training example; the goal of learning is to find a decision rule using this data.

In many cases, however, a teacher can supplement training data with some additional information (comments) which will not be available at the test stage.

For example, consider the case of learning a decision rule for prognosis of a disease in a year, given the current symptoms of a patient. In this problem, for the training data described by current symptoms and outcome in a year, one can also obtain additional information about symptoms in half a year. Can this additional information help to predict the outcome of disease in a year?

Consider another example: find a rule that can classify biopsy images, into two categories cancer and non-cancer. Here the problem is given images described in pixel space find the classification rule. However, along with a picture the doctor has a report written by a pathologist which describes the picture using a high level holistic language. The problem is to use pictures along with information given in the reports (which will not be available at the test stage) to find a better classification rule in pixel space.

In these two examples at the training stage, for every input vector $\mathbf{x}_i$ in space $\mathcal{X}$ we are also given an additional information $\mathbf{x}_i^*$ in another space $\mathcal{X}^*$. Since the additional data is not available (hidden) at the test stage, we call it *hidden information* and call the above problem *learning using hidden information or master-class learning*[1].

[1] In human master-class learning teachers comments play the most important role.

In the classical setting there are no good or bad teachers since they just provide labels. In the new learning setting, however, a good teacher provides good additional descriptions (comments) of training examples which can help to construct a better decision rule in space $\mathcal{X}$. So in the classical setting we are given pairs $(\mathbf{x}_i, y_i)$, in the new setting we are given the triplet $(\mathbf{x}_i, \mathbf{x}_i^*, y_i)$, where $\mathbf{x}_i^*$ is the teacher's holistic description (comments) of example $\mathbf{x}_i$.

Below, we consider a pattern recognition algorithm for learning from hidden information and demonstrate its potential advantage relative to classical SVM. In particular, in digit recognition problem for any digit $\mathbf{x}_i$ of training set defined in the pixel space we will give its poetic description $\mathbf{x}_i^*$ given in holistic space. We will show that the holistic description helps to improve the decision rule in the pixel space.

The organization of this text is as follows. Section 1 reviews SVM and its extension SVM+ is presented in section 2. Section 3 investigates one specific mechanism of modeling slacks using hidden information. Section 4 is devoted to master-class learning of the digit recognition problem. Section 5 is our conclusion.

1. Background: SVM

SVM [2,3] is a supervised learning algorithm which learns a decision rule, $y = f(\mathbf{x})$, from an admissible set of functions given training data

$$(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}, \qquad y_i \in \{-1, 1\}.$$

To construct a decision rule $y = f(\mathbf{x})$, SVM maps vectors $\mathbf{x} \in \mathbf{X}$ to $\mathbf{z} \in \mathbf{Z}$ and finds a hyperplane that separates the images $\mathbf{z}_i$'s of training vectors $\mathbf{x}_i$'s in the two classes with minimal number of errors. Among the many possible hyperplanes, SVM chooses the optimal one that does this separation with maximum margin [3]. The hyperplane is specified by a weight vector $\mathbf{w}$ and a threshold b which are found by solving the quadratic optimization problem. Minimize over $\mathbf{w}$, b, and ξ the functional

$$R(\mathbf{w}, b, \xi) = \frac{1}{2}\mathbf{w}^2 + C\sum_{i=1}^{\ell}\xi_i \tag{1}$$

subject to constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{z}_i + b) &\geq 1 - \xi_i, \\ \xi_i &\geq 0, i = 1, \dots, \ell \quad , \end{aligned} \tag{1a}$$

where C is fixed and ξ_i's are the slack variables. SVM finds the optimal hyperplane by solving the optimization problem (1) in the dual space. SVM does not calculate the inner product in $\mathcal{Z}$ space. Instead, it uses the "kernel trick". According to Mercer's theorem, for every inner product in $\mathcal{Z}$ space, there exists a positive definite function $K(\mathbf{x}_i, \mathbf{x}_j)$ (kernel function) such that $\mathbf{z}_i \cdot \mathbf{z}_j = K(\mathbf{x}_i, \mathbf{x}_j)$, for all $i, j = 1, \dots, \ell$. So, we only need to specify the kernel function for learning a nonlinear decision rule.

Decision rule for SVM has a form

$$y = f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right), \tag{2}$$

where the coefficients α_i's are obtained by maximizing the functional

$$R(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i y_i \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

subject to constraints

$$\begin{gathered} \sum_{i=1}^{\ell} \alpha_i y_i = 0, \\ 0 \le \alpha_i \le C, \quad i = 1, \dots, \ell. \end{gathered} \tag{3a}$$

For universal kernels (for example, Gaussian kernel), with increasing training data SVM solution converges to the Bayesian solution. Also SVM has been demonstrated to perform well on datasets with limited size [4,5].

2. Supervised learning using hidden information

Consider the problem of learning a decision rule where at the training stage, we are given triplets

$$(\mathbf{X}, \mathbf{X}^*, \mathbf{Y}) = \{(\mathbf{x}_i, \mathbf{x}_i^*, y_i)\}_{i=1}^{\ell}, \quad y_i \in \{-1, 1\}.$$

Here $\mathbf{X}^*$ is additional information from the vector space $\mathcal{X}^*$, generally speaking, different from space $\mathcal{X}$. The goal is to use the additional information $\mathbf{X}^*$ to find a decision rule $y = f(\mathbf{x})$ (in space $\mathcal{X}$), which is better than the decision rule obtained without using the additional information [1]. The space $\mathcal{X}$ in which the decision rule is constructed is called the *decision space*, and the space $\mathcal{X}^*$ is called the *correction space*.

Compared to the classical supervised learning setting in which an oracle provides labels for training vectors, the new setting expands a teacher's role who can supply additional descriptions for training vectors. One can realize this idea using the SVM+ algorithm [1].

2.1. SVM+

SVM+ is a generalization of SVM. It allows us to model relationships[2] between the slack variables, ξ_i, in SVM i.e,

$$\xi_i = \psi(\mathbf{x}_i^*, \delta), \qquad \delta \in \mathcal{D},$$

where $\psi(\mathbf{x}_i^*, \delta)$ belongs to some set of admissible functions in $\mathcal{X}^*$, called the *correcting functions*. This is a generalization of SVM because for SVM, $\mathbf{X}^* = \mathbf{X}$ and $\psi(\mathbf{x}_i, \delta)$ is the

[2] These relationships can be non-linear (not just correlations); in fact, we will model non-linear relationships.

set of all possible functions. In SVM+ slacks are no longer variables in the optimization problem, so SVM+ can depend on less parameters than SVM, and consequently the decision rule found by SVM+ is chosen from a set with smaller capacity than SVM which can lead to a better generalization.

Similar to the mapping of vectors $\mathbf{x}_i \in \mathcal{X}$ to $\mathbf{z}_i \in \mathcal{Z}$ in the decision space, the vectors $\mathbf{x}_i^* \in \mathcal{X}^*$ in correction space are mapped to $\mathbf{z}_i^* \in \mathcal{Z}^*$. To do this we use two different kernels - the kernel for the decision rule (represented by $K(,)$) and the kernel for the correcting function (represented by $K^*(,)$). In space $\mathcal{Z}^*$, the correcting function has the form

$$\psi(\mathbf{x}_i^*, \delta) = \mathbf{w}^* \cdot \mathbf{z}_i^* + d; \quad \mathbf{w}^* \in \mathcal{Z}^*,\ d \in \mathbb{R}. \tag{4}$$

Using this mapping, the slacks can be written as $\xi_i = \mathbf{w}^* \cdot \mathbf{z}_i^* + d$. This leads to the following problem formulation: minimize over $\mathbf{w}$, b, $\mathbf{w}^*$, and d the functional

$$R(\mathbf{w}, b, \mathbf{w}^*, d) = \frac{1}{2}\mathbf{w}^2 + \frac{\gamma}{2}\mathbf{w}^{*2} + C\sum_{i=1}^{\ell}(\mathbf{w}^* \cdot \mathbf{z}_i^* + d) \tag{5}$$

subject to constraints

$$\begin{array}{c} y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - (\mathbf{w}^* \cdot \mathbf{z}_i^* + d), \\ (\mathbf{w}^* \cdot \mathbf{z}_i^* + d) \geq 0, \ \ i = 1, \ldots, \ell\,, \end{array} \tag{5a}$$

where γ and C are parameters. The Lagrangian for (5) is

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta) = \tfrac{1}{2}\mathbf{w}^2 + \tfrac{\gamma}{2}\mathbf{w}^{*2} + C\sum_{i=1}^{\ell}(\mathbf{w}^* \cdot \mathbf{z}_i^* + d) - \sum_{i=1}^{\ell}\beta_i(\mathbf{w}^* \cdot \mathbf{z}_i^* + d) \\ - \sum_{i=1}^{\ell}\alpha_i\left[y_i(\mathbf{w} \cdot \mathbf{z}_i + b) - 1 + (\mathbf{w}^* \cdot \mathbf{z}_i^* + d)\right], \end{aligned} \tag{6}$$

where $\alpha \geq \mathbf{0}$ and $\beta \geq \mathbf{0}$ are the Lagrange multipliers. To obtain the minimum of $L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta)$, over $\mathbf{w}$, $\mathbf{w}^*$, b and d, we equate the corresponding partial derivatives to 0 and obtain the following:

$$\begin{array}{lcl} \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta) = 0 & \Rightarrow & \mathbf{w} = \sum_{i=1}^{\ell}\alpha_i y_i \mathbf{z}_i\,, \\ \frac{\partial}{\partial \mathbf{w}^*} L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta) = 0 & \Rightarrow & \mathbf{w}^* = \frac{1}{\gamma}\sum_{i=1}^{\ell}(\alpha_i + \beta_i - C)\mathbf{z}_i^*\,, \\ \frac{\partial}{\partial b} L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta) = 0 & \Rightarrow & \sum_{i=1}^{\ell}\alpha_i y_i = 0\,, \\ \frac{\partial}{\partial d} L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta) = 0 & \Rightarrow & \sum_{i=1}^{\ell}(\alpha_i + \beta_i - C) = 0\,. \end{array} \tag{7}$$

Substituting (7) in (6) and using the kernel trick, we obtain the dual of (5) as: maximize over α and β the functional

$$R(\alpha,\beta) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i y_i \alpha_j y_j K(\mathbf{x}_i,\mathbf{x}_j) - \frac{1}{2\gamma}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\alpha_i+\beta_i-C)(\alpha_j+\beta_j-C)K^*(\mathbf{x}_i^*,\mathbf{x}_j^*) \tag{8}$$

subject to constraints

$$\begin{aligned} &\sum_{i=1}^{\ell}\alpha_i y_i = 0\,, \\ &\sum_{i=1}^{\ell}(\alpha_i+\beta_i) = \ell C\,, \\ &\alpha_i \geq 0,\ \ \beta_i \geq 0, \qquad i=1,\ldots,\ell\,. \end{aligned} \tag{8a}$$

The decision rule for this algorithm, called SVM+, has the same form as the SVM decision rule (2). It differs in the way it determines the coefficients α_i's. The coefficients β_i's are used only in the correcting function which is given by

$$\psi(\mathbf{x}^*) = \mathbf{w}^*\cdot\mathbf{z}^* + d = \frac{1}{\gamma}\sum_{i=1}^{\ell}(\alpha_i+\beta_i-C)K^*(\mathbf{x}_i^*,\mathbf{x}^*) + d\,. \tag{9}$$

The quadratic optimization related to SVM+ (8) is different from the classical SVM but can be solved using the generalized sequential minimal optimization procedure for SVM+ described in [6] similar to SMO [7] used for SVM. So, the computational effort in SVM and SVM+ are comparable.

SVM+ can be used for learning from hidden information by modeling slacks using information in $\mathbf{X}^*$.

3. Idea of estimating slacks

In this section, we explore the potential advantage of knowing the values of slacks in SVM and hypothesize one of the possible mechanisms of how hidden information is used in SVM+ for modeling slacks to produce a better decision rule. Accordingly, we present three cases:
The first case (I1) uses knowledge of slacks in SVM.
The second case (I2) uses information similar to slacks in SVM+.
The third case (I3) uses hidden information to estimate correcting function $\psi(\mathbf{x}^*)$.

I1. Suppose that along with the training data we are given the values of slacks,

$$\xi_i^{bst} = \left[1 - y_i(\mathbf{w}_{bst}\cdot\mathbf{z}_i + b_{bst})\right]_+\,,\ i=1,\ldots,\ell\,,$$

for the best decision rule (in the admissible set) with parameters $(\mathbf{w}_{bst}, b_{bst})$ and where $a_+ = a$ if $a > 0$, and 0, otherwise.
In this case, we can use training data and these values of slacks in SVM-like algorithm to find a decision rule by solving the following optimization problem: minimize over $\mathbf{w}$ and b the functional

$$R(\mathbf{w}, b) = \frac{\mathbf{w}^2}{2}$$

subject to constraints

$$y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq r_i, \text{ where } r_i = 1 - \xi_i^{bst}, \quad i = 1, \ldots, \ell\ .$$

I2. We now consider a modification of the previous case where we are given training data and the set of so called, *deviation values* (dev. vals.),

$$d_i^{bst} = 1 - y_i(\mathbf{w}_{bst} \cdot \mathbf{z}_i + b_{bst}), \quad i = \{1, \ldots, \ell\}.$$

In this case, we use SVM+ with input $(\mathbf{X}, \mathbf{X}^*, \mathbf{Y})$, where $\mathbf{X}^* = \mathbf{d}^{bst}$, is the set of deviation values.

I3. Suppose there exist a large number of triplets $(\mathbf{X}_o, \mathbf{X}_o^*, \mathbf{Y}_o)$ but we are given a fixed set of training data $(\mathbf{X}, \mathbf{X}^*, \mathbf{Y})$ along with a large number of data $(\mathbf{X}_2^*, \mathbf{Y}_2)$. Both of these sets are taken from $(\mathbf{X}_o, \mathbf{X}_o^*, \mathbf{Y}_o)$.

We first use SVM for classifying $(\mathbf{X}_2^*, \mathbf{Y}_2)$ and obtain the decision rule

$$\hat{f}(\mathbf{x}^*) = sgn\left(\sum_{j=1}^{|\mathbf{X}_2^*|} \alpha_j y_j K(\mathbf{x}_j^*, \mathbf{x}^*) + b\right).$$

Second, for each training vector $\mathbf{x}_i^*$, we compute the estimate of deviation values as

$$d_i = 1 - y_i\left(\sum_{j=1}^{|\mathbf{X}_2^*|} \alpha_j y_j K(\mathbf{x}_j^*, \mathbf{x}_i^*) + b\right), \text{ where } \mathbf{x}_i^* \in \mathbf{X}^*.$$

Finally we use SVM+ with the input $(\mathbf{X}, \mathbf{d}, \mathbf{Y})$.

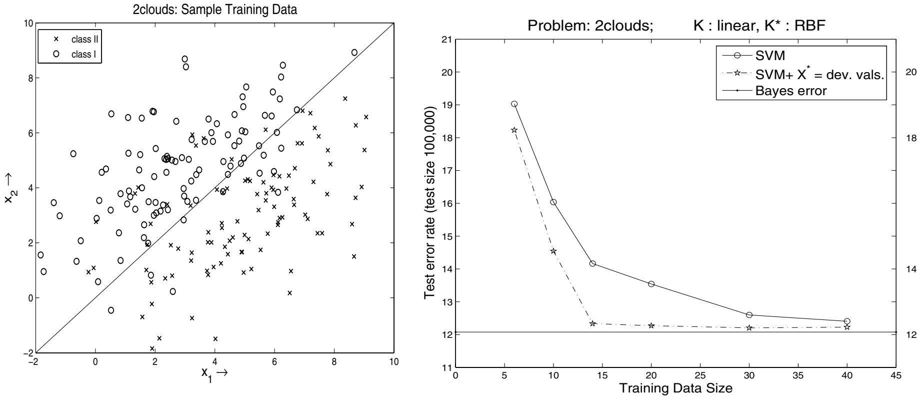
Below we test our methods of modeling slacks and deviation values, as described in I1-I3, to find the limit of possible improvement that can be achieved by using the knowledge of slacks (or their estimates). Note that the first two cases are not practical since they require knowledge of the best decision rule in the admissible set. These cases are used only to test the conjecture. The third case, however, is practical and can be very useful.

3.1. Synthetic Example 1

Our first example, defined by two clouds shown in Fig. 1(a), involves separating points generated by two 2D Gaussians[3].

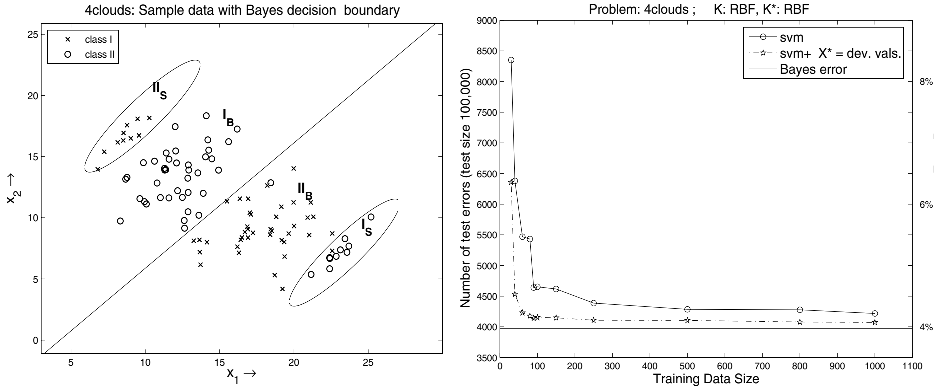
[3] The prior probabilities for the two classes are equal, i.e, $p_1 = p_2 = 0.5$. Their mean vectors are $\mu_\mathbf{1} = [3\ \ 5]$ and $\mu_\mathbf{2} = [5\ \ 3]$, respectively. Both classes have equal covariance $\Sigma = [5\ \ 3;\ 2\ \ 4]$. Linear kernel was used in the decision space, and RBF kernel in the correction space. Two independent sets were used for tuning the parameters (C for SVM; C and kernel parameter for SVM+) and testing. We performed 20 trials for each experiment.

For this problem, Bayes decision rule yields 12.078% error rate on the test set. The best decision rule is linear in $\mathcal{X}$ and has parameters $\mathbf{w}_{bst} = (1, -1)$ and $b_{bst} = 0$ (see Fig.1(a)). The decision rules obtained by using the methods I1, I2 and I3 (for I3, $\mathbf{X}^*$ was from the same space as $\mathbf{X}$) have almost the same performance. To maintain legibility in the plot, we show (by the dashed line marked with stars in Fig. 1(b)) only the performance of SVM+ using estimate of deviation values, d^i. All three methods led to better decision rules than the decision rule obtained by using the classical SVM (line with circles).



(a) (b)

Figure 1. 2clouds problem. (a) Sample data with Bayes decision boundary (diagonal line). (b) Performance comparison of SVM+ and SVM.



(a) (b)

Figure 2. 4clouds problem. (a) Sample data with Bayes decision boundary (ellipses and line). Class **I** is shown by circles; it is the union of one big subclass I_B and a small subclass I_S. Class **II** is shown by crosses; it is the union of the big subclass II_B and the small subclass II_S. Each subclass is a normal distribution with parameters shown in footnote 4. (b) Performance comparison of SVM+ and SVM.

3.2. Synthetic Example 2

Consider another synthetic problem[4] (4clouds) shown in Fig. 2(a). For this problem, Bayes decision rule yields 3.971% error rate.

In this example, we tested the method I3. Here $\mathbf{X}^*$ was from the same space as $\mathbf{X}$ and we used ten thousand training examples (as additional input $(\mathbf{X}_2^*, \mathbf{Y}_2)$) to approximate the best decision rule. Performance of SVM+ using deviation values from this decision rule is depicted by the dashed line marked with stars in Fig. 2(b). It performs close to the Bayes decision rule.

4. Using hidden information (master-class learning)

Usually, we do not have information about deviation values in explicit form but a teacher can provide information about deviation values by describing training examples in a second (holistic) space, say $\mathcal{X}^*$. The idea is to evaluate deviation values in holistic space and use them as elements $\mathbf{x}^*$ of triplets in SVM+ algorithm to estimate slacks in the decision space.

Below we test this idea on the problem of classifying digits 5 and 8 using MNIST data which is presented in 28x28 pixel space.

Classification of digits 5 and 8 using 28x28 pixel images is an easy problem. To make this problem more difficult we resized them to 10x10 pixel images. We used 100 training examples (first 50 instances of digits 5 and 8 in MNIST data). For every training vector (digit), we created its poetic description. A sample of digits is shown in Fig. 3 followed by a sample of poetic descriptions.

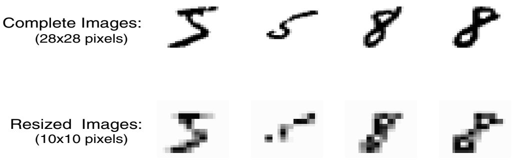


Figure 3. Sample digits along with their resized images.

Poetic description for the first image of 5 (see Fig. 3):

Not absolute two-part creature. Looks more like one impulse. As for two-partness the head is a sharp tool and the bottom is round and flexible. As for tools it is a man with a spear ready to throw it. Or a man is shooting an arrow. He is firing the bazooka. He swung his arm, he drew back his arm and is ready to strike. He is running. He is flying. He is looking ahead. He is swift. He is throwing a spear ahead. He is dangerous. It is slanted to the right. Good snaked-ness. The snake is attacking. It is going to jump and bite.

[4] The parameters (using the symbols shown in Fig. 2(a)) for the clouds are prior probabilities: $p_{I_B} = p_{II_B} = 0.4$, $p_{I_S} = p_{II_S} = 0.1$; mean vectors: $\mu_{I_B} = [12\ 13]$, $\mu_{II_B} = [18\ 9]$, $\mu_{I_S} = [23\ 7]$, $\mu_{II_S} = [9\ 17]$, and covariances matrices: $\Sigma_B = [\frac{16}{3}\ 1;\ 1\ 4]$, $\Sigma_S = [1\ 1;\ 1\ \frac{4}{3}]$. Two independent sets were used as validation set (for tuning C and the kernel hyper-parameter for SVM; C, γ, and the two kernel hyper-parameters for SVM+) and test set. Parameters were tuned using recursive grid search. For each training size, 12 trials were performed to obtain average error rate.

> *It is free and absolutely open to anything. It shows itself, no kidding. Its bottom only slightly (one point!) is on earth. He is a sportsman and in the process of training. The straight arrow and the smooth flexible body. This creature is contradictory - angular part and slightly roundish part. The lashing whip (the rope with a handle). A toe with a handle. It is an outside creature, not inside. Everything is finite and open. Two open pockets, two available holes, two containers. A piece of rope with a handle. Rather thick. No loops, no saltire. No hill at all. Asymmetrical. No curlings.*

Poetic description for the first image of 8 (see Fig. 3):

> *Two-part creature. Not very perfect infinite way. It has a deadlock, a blind alley. There is a small right-hand head appendix, a small shoot. The right-hand appendix. Two parts. A bit disproportionate. Almost equal. The upper one should be a bit smaller. The starboard list is quite right. It is normal like it should be. The lower part is not very steady. This creature has a big head and too small bottom for this head. It is nice in general but not very self-assured. A rope with two loops which do not meet well. There is a small upper right-hand tail. It does not look very neat. The rope is rather good - not very old, not very thin, not very thick. It is rather like it should be. The sleeping snake which did not hide the end of its tail. The rings are not very round - oblong - rather thin oblong. It is calm. Standing. Criss-cross. The criss-cross upper angle is rather sharp. Two criss-cross angles are equal. If a tool it is a lasso. Closed absolutely. Not quite symmetrical (due to the horn).*

Poetic descriptions were translated into 21-dimensional feature vectors[5]. A subset of these features (with range of possible values) is: `two-part-ness` (0 - 5); `tilting to the right` (0 - 3); `aggressiveness` (0 - 2); `stability` (0 - 3); `uniformity` (0 - 3), and so on. The values of these features (in the order they appear above) for the first 5 and 8 are [2, 1, 2, 0, 1], and [4, 1, 1, 0, 2], respectively.

Our goal was to construct a decision rule for classifying 10x10 pixel images [6] using the 100 dimensional pixel space $\mathcal{X}$ and the corresponding 21-dimensional vectors in the space $\mathcal{X}^*$. This idea was realized using the SVM+ algorithm in the following two settings.

A. Poetic descriptions were used as vectors in 21-dimensional correction space. Decision rule was obtained by using $(\mathbf{X}, \mathbf{X}^*, \mathbf{Y})$ in SVM+ as input and solving (8).

B. Poetic descriptions were used to estimate the deviation values which were then used as one dimensional correction space in SVM+. The procedure to do this is described below.

1. Use SVM with RBF kernel to classify $(\mathbf{X}^*, \mathbf{Y})$, and obtain the decision rule,

$$f(\mathbf{x}^*) = \text{sgn}(\sum_{j=1}^{|\mathbf{X}^*|} \alpha_j y_j K(\mathbf{x}_j^*, \mathbf{x}^*) + b).$$

[5] Other encodings of the poetic descriptions can possibly improve results, however, we experimented only with the above described 21-dimensional encoding.

[6] MNIST training data has 5,222 and 5,652 (28x28 pixel) images of 5 and 8, respectively. As we just mentioned, classifying digits 5 and 8 using 28x28 images is an easy problem. Therefore, the problem was made more difficult by resizing digits to 10x10 pixel images. We used 4,000 digits as validation set for tuning the parameters in SVM(C and RBF kernel parameter) and SVM+ (C, γ and the parameters of two RBF kernels).

2. Find the values,

$$d_i^p = 1 - y_i(\sum_{j=1}^{|\mathbf{X}^*|} \alpha_j y_j K(\mathbf{x}_j^*, \mathbf{x}_i^*) + b),$$

where $\mathbf{x}_i^*$ is the poetic description for the training vector $\mathbf{x}_i$.

3. Obtain decision rule by using SVM+ with input $(\mathbf{X}, \mathbf{d}^p, \mathbf{Y})$.

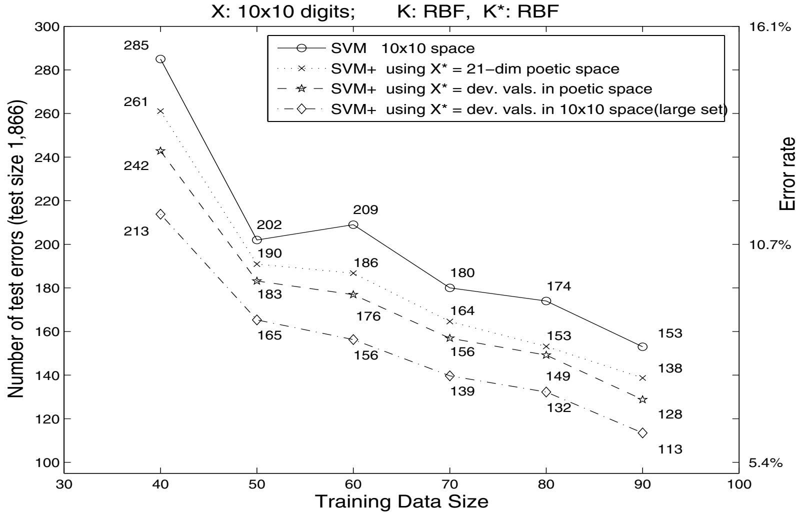


Figure 4. Performance of SVM+ on the digit recognition task.

Results of using different correction spaces (21-dimensional poetic space and 1-dimensional space of deviation values) in SVM+ are shown in Fig. 4. Performance of SVM trained and tested on 10x10 digits is shown by the line marked with circles and the average[7] of test errors are shown by numbers. Performance using 21-dimensional poetic space as correction space (setting A above) is shown by the line marked with crosses. Performance using deviation values in the poetic space as correction space (setting B) is shown by the line with stars. In both cases use of hidden information improves performance.

To evaluate the limit of possible improvement over SVM, we used deviation values from a quasi-optimal decision rule in 10x10 pixel space. We constructed this decision rule by using SVM with about 6,500 10x10 pixel digits. Then, for training vectors we computed their deviation values from the obtained decision rule (see I3). Performance of SVM+ using deviation values is shown by the dashed line with diamonds.

A good master-class teacher can, probably, develop descriptions of hidden information that allow one to be close to this performance. Our first experience with master-class teaching of computers to recognize digits yielded only 60% of possible improvement.

[7] For every training data size, we solved this problem 12 times with different random samples of training data. The average of test errors is reported.

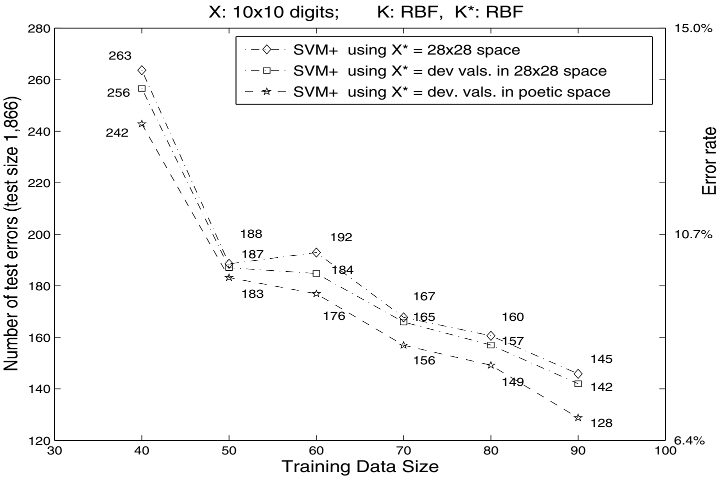


Figure 5. Comparison between information obtained 28x28 pixel space and poetic description in SVM+.

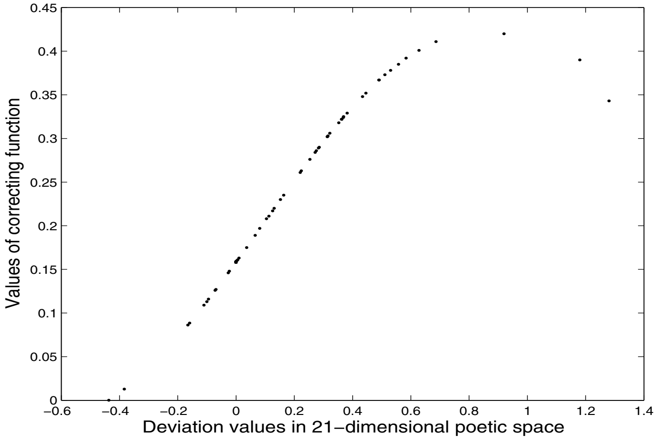


Figure 6. Plot between deviation values from the decision rule in the poetic space and corresponding correcting function values. This representative plot was generated for a sample of training data size 70.

Fig. 6 shows functional relationship between the deviation values defined in the poetic space and the values of the correcting function.

To understand how much information is contained in poetic descriptions, we conducted the following experiment. We used 28x28 pixel digits (784 dimensional space) instead of the 21-dimensional poetic descriptions in settings A and B in SVM+ (results shown in Fig. 5). In both the settings, using 28x28 pixel description of digits SVM+ performs worse than SVM+ that uses poetic descriptions.

5. Conclusion

Our results on digit recognition using poetic descriptions indicate that there is information other than technical (pixel space for digits) that plays an important role in learning. Based on results, it appears that non-technical (poetic descriptions) was more relevant to the digit recognition task than information in 28x28 pixel description of digits. It looks like, sometimes using non-technical descriptions can achieve results that are not easy to achieve just by using technical descriptions.

The digit recognition problem using poetic descriptions is not an isolated problem that benefits from information in multiple languages (spaces). The scenario with information in multiple languages (technical information and teachers comments) is very general and decision making problems involving such scenarios arise often. For instance, in the stock market prediction problem information is not only in various quantitative indices but also in afterward verbal analysis of market analysts. Incorporating descriptions from these experts during training stage can improve rule for prediction. In genomics, for the protein sequences homology detection problem, using 3D structure and functional classification of proteins at the training stage can improve the rule for prediction of remote homology using primary sequences only.

References

[1] Vapnik, V. N. (2006) Estimation of Dependences Based on Empirical Data: Empirical Inference Science, Springer, .

[2] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992) In Proceedings of the fifth annual workshop on Computational learning theory volume **5**, Pittsburgh, PA, USA: . pp. 144–152.

[3] Cortes, C. and Vapnik, V. N. (1995) *Machine Learning* **20**, 273–297.

[4] Bernhard Scölkopf, Christopher Burges, and Alexander Smola, (ed.) (1998) Advances in Kernel Methods: Support Vector Learning, MIT Press, .

[5] Bennett, K. P. and Campbell, C. (2000) *SIGKDD Explorations* **2(2)**, 1–13.

[6] Izmailov, R., Vashist, A., and Vapnik, V. Generalized sequential minimal optimization for SVM+ computations Technical Report 2007-L048 NEC Labs. America Princeton, NJ, USA (2007).

[7] Platt, J. C. (1998) Sequential minimal optimization: A fast algorithm for training support vector machines In Bernhard Schölkopf, Christopher Burges, and Alexander Smola, (ed.), Advances in Kernel Methods: Support Vector Learning, MIT Press.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-15

Learning using Large Datasets

Léon Bottou [a] and Olivier Bousquet [b]
[a] *NEC Laboratories America, Princeton, NJ08540, USA*
[b] *Google Zürich, 8002 Zürich, Switzerland*

Abstract. This contribution develops a theoretical framework that takes into account the effect of approximate optimization on learning algorithms. The analysis shows distinct tradeoffs for the case of small-scale and large-scale learning problems. Small-scale learning problems are subject to the usual approximation–estimation tradeoff. Large-scale learning problems are subject to a qualitatively different tradeoff involving the computational complexity of the underlying optimization algorithms in non-trivial ways. For instance, a mediocre optimization algorithms, stochastic gradient descent, is shown to perform very well on large-scale learning problems.

Keywords. Large-scale learning. Optimization. Statistics.

Introduction

The computational complexity of learning algorithms has seldom been taken into account by the learning theory. Valiant [1] states that a problem is "learnable" when there exists a probably approximatively correct learning algorithm *with polynomial complexity*. Whereas much progress has been made on the statistical aspect (e.g., [2,3,4]), very little has been told about the complexity side of this proposal (e.g., [5].)

Computational complexity becomes the limiting factor when one envisions large amounts of training data. Two important examples come to mind:

- Data mining exists because competitive advantages can be achieved by analyzing the masses of data that describe the life of our computerized society. Since virtually every computer generates data, the data volume is proportional to the available computing power. Therefore one needs learning algorithms that scale roughly linearly with the total volume of data.
- Artificial intelligence attempts to emulate the cognitive capabilities of human beings. Our biological brains can learn quite efficiently from the continuous streams of perceptual data generated by our six senses, using limited amounts of sugar as a source of power. This observation suggests that there are learning algorithms whose computing time requirements scale roughly linearly with the total volume of data.

This contribution finds its source in the idea that approximate optimization algorithms might be sufficient for learning purposes. The first part proposes new decomposition of the test error where an additional term represents the impact of approximate optimization. In the case of small-scale learning problems, this decomposition reduces to the well known tradeoff between approximation error and estimation error. In the case of large-scale learning problems, the tradeoff is more complex because it involves the computational complexity of the learning algorithm. The second part explores the asymptotic properties of the large-scale learning tradeoff for various prototypical learning algorithms under various assumptions regarding the statistical estimation rates associated with the chosen objective functions. This part clearly shows that the best optimization algorithms are not necessarily the best learning algorithms. Maybe more surprisingly, certain algorithms perform well regardless of the assumed rate for the statistical estimation error. Finally, the final part presents some experimental results.

1. Approximate Optimization

Following [6,2], we consider a space of input-output pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ endowed with a probability distribution $P(x, y)$. The conditional distribution $P(y|x)$ represents the unknown relationship between inputs and outputs. The discrepancy between the predicted output $\hat{y}$ and the real output y is measured with a loss function $\ell(\hat{y}, y)$. Our benchmark is the function f^* that minimizes the expected risk

$$E(f) = \int \ell(f(x), y)\, dP(x, y) = \mathbb{E}\,[\ell(f(x), y)],$$

that is,

$$f^*(x) = \arg\min_{\hat{y}} \mathbb{E}\,[\ell(\hat{y}, y)|\, x].$$

Although the distribution $P(x, y)$ is unknown, we are given a sample $\mathcal{S}$ of n independently drawn training examples (x_i, y_i), $i = 1 \ldots n$. We define the empirical risk

$$E_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) = \mathbb{E}_n[\ell(f(x), y)].$$

Our first learning principle consists in choosing a family $\mathcal{F}$ of candidate prediction functions and finding the function $f_n = \arg\min_{f \in \mathcal{F}} E_n(f)$ that minimizes the empirical risk. Well known combinatorial results (e.g., [2]) support this approach provided that the chosen family $\mathcal{F}$ is sufficiently restrictive. Since the optimal function f^* is unlikely to belong to the family $\mathcal{F}$, we also define $f^*_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} E(f)$. For simplicity, we assume that f^*, $f^*_{\mathcal{F}}$ and f_n are well defined and unique.

We can then decompose the excess error as

$$\begin{aligned} \mathbb{E}\,[E(f_n) - E(f^*)] &= \underbrace{\mathbb{E}\,[E(f^*_{\mathcal{F}}) - E(f^*)]}_{\mathcal{E}_{\text{app}}} + \underbrace{\mathbb{E}\,[E(f_n) - E(f^*_{\mathcal{F}})]}_{\mathcal{E}_{\text{est}}}, \\ &= \qquad \mathcal{E}_{\text{app}} \qquad + \qquad \mathcal{E}_{\text{est}} \end{aligned} \tag{1}$$

where the expectation is taken with respect to the random choice of training set. The *approximation error* $\mathcal{E}_{\text{app}}$ measures how closely functions in $\mathcal{F}$ can approximate the optimal solution f^*. The *estimation error* $\mathcal{E}_{\text{est}}$ measures the effect of minimizing the empirical risk $E_n(f)$ instead of the expected risk $E(f)$. The estimation error is determined by the number of training examples and by the capacity of the family of functions [2]. Large families[1] of functions have *smaller approximation errors* but lead to *higher estimation errors*. This tradeoff has been extensively discussed in the literature [2,3] and lead to excess error that scale between the inverse and the inverse square root of the number of examples [7,8].

1.1. Optimization Error

Finding f_n by minimizing the empirical risk $E_n(f)$ is often a computationally expensive operation. Since the empirical risk $E_n(f)$ is already an approximation of the expected risk $E(f)$, it should not be necessary to carry out this minimization with great accuracy. For instance, we could stop an iterative optimization algorithm long before its convergence.

Let us assume that our minimization algorithm returns an approximate solution $\tilde{f}_n$ that minimizes the objective function up to a predefined tolerance $\rho \geq 0$.

$$E_n(\tilde{f}_n) < E_n(f_n) + \rho$$

We can then decompose the excess error $\mathcal{E} = \mathbb{E}\big[E(\tilde{f}_n) - E(f^*)\big]$ as

$$\begin{aligned} \mathcal{E} &= \underbrace{\mathbb{E}\left[E(f^*_{\mathcal{F}}) - E(f^*)\right]}_{\textstyle \mathcal{E}_{\mathrm{app}}} + \underbrace{\mathbb{E}\left[E(f_n) - E(f^*_{\mathcal{F}})\right]}_{\textstyle \mathcal{E}_{\mathrm{est}}} + \underbrace{\mathbb{E}\big[E(\tilde{f}_n) - E(f_n)\big]}_{\textstyle \mathcal{E}_{\mathrm{opt}}} . \end{aligned} \tag{2}$$

We call the additional term $\mathcal{E}_{\mathrm{opt}}$ the *optimization error*. It reflects the impact of the approximate optimization on the generalization performance. Its magnitude is comparable to ρ (see section 2.1.)

1.2. The Approximation–Estimation–Optimization Tradeoff

This decomposition leads to a more complicated compromise. It involves three variables and two constraints. The constraints are the maximal number of available training example and the maximal computation time. The variables are the size of the family of functions $\mathcal{F}$, the optimization accuracy ρ, and the number of examples n. This is formalized by the following optimization problem.

$$\min_{\mathcal{F},\rho,n} \quad \mathcal{E} = \mathcal{E}_{\mathrm{app}} + \mathcal{E}_{\mathrm{est}} + \mathcal{E}_{\mathrm{opt}} \quad \text{subject to} \begin{cases} n \leq n_{\max} \\ T(\mathcal{F},\rho,n) \leq T_{\max} \end{cases} \tag{3}$$

The number n of training examples is a variable because we could choose to use only a subset of the available training examples in order to complete the optimization within the alloted time. This happens often in practice. Table 1 summarizes the typical evolution of the quantities of interest with the three variables $\mathcal{F}$, n, and ρ increase.

The solution of the optimization program (3) depends critically of which budget constraint is active: constraint $n < n_{\max}$ on the number of examples, or constraint $T < T_{\max}$ on the training time.

[1] We often consider nested families of functions of the form $F_c = \{f \in \mathcal{H},\ \Omega(f) \leq c\}$. Then, for each value of c, function f_n is obtained by minimizing the regularized empirical risk $E_n(f) + \lambda\Omega(f)$ for a suitable choice of the Lagrange coefficient λ. We can then control the estimation-approximation tradeoff by choosing λ instead of c.

Table 1. Typical variations when $\mathcal{F}$, n, and ρ increase.

		$\mathcal{F}$	n	ρ
$\mathcal{E}_{\mathrm{app}}$	(approximation error)	$\searrow$		
$\mathcal{E}_{\mathrm{est}}$	(estimation error)	$\nearrow$	$\searrow$	
$\mathcal{E}_{\mathrm{opt}}$	(optimization error)	$\cdots$	$\cdots$	$\nearrow$
T	(computation time)	$\nearrow$	$\nearrow$	$\searrow$

- We speak of *small-scale learning problem* when (3) is constrained by the maximal number of examples n_{max}. Since the computing time is not limited, we can reduce the optimization error $\mathcal{E}_{\mathrm{opt}}$ to insignificant levels by choosing ρ arbitrarily small. The excess error is then dominated by the approximation and estimation errors, $\mathcal{E}_{\mathrm{app}}$ and $\mathcal{E}_{\mathrm{est}}$. Taking $n = n_{\mathrm{max}}$, we recover the approximation-estimation tradeoff that is the object of abundant literature.
- We speak of *large-scale learning problem* when (3) is constrained by the maximal computing time T_{max}. Approximate optimization, that is choosing $\rho > 0$, possibly can achieve better generalization because more training examples can be processed during the allowed time. The specifics depend on the computational properties of the chosen optimization algorithm through the expression of the computing time $T(\mathcal{F}, \rho, n)$.

2. The Asymptotics of Large-scale Learning

In the previous section, we have extended the classical approximation-estimation tradeoff by taking into account the optimization error. We have given an objective criterion to distiguish small-scale and large-scale learning problems. In the small-scale case, we recover the classical tradeoff between approximation and estimation. The large-scale case is substantially different because it involves the computational complexity of the learning algorithm. In order to clarify the large-scale learning tradeoff with sufficient generality, this section makes several simplifications:

- We are studying upper bounds of the approximation, estimation, and optimization errors (2). It is often accepted that these upper bounds give a realistic idea of the actual convergence rates [9,10,11,12]. Another way to find comfort in this approach is to say that we study guaranteed convergence rates instead of the possibly pathological special cases.
- We are studying the asymptotic properties of the tradeoff when the problem size increases. Instead of carefully balancing the three terms, we write $\mathcal{E} = \mathcal{O}(\mathcal{E}_{\mathrm{app}}) + \mathcal{O}(\mathcal{E}_{\mathrm{est}}) + \mathcal{O}(\mathcal{E}_{\mathrm{opt}})$ and only need to ensure that the three terms decrease with the same asymptotic rate.
- We are considering a fixed family of functions $\mathcal{F}$ and therefore avoid taking into account the approximation error $\mathcal{E}_{\mathrm{app}}$. This part of the tradeoff covers a wide spectrum of practical realities such as choosing models and choosing features. In the context of this work, we do not believe we can meaningfully address this without discussing, for instance, the thorny issue of feature selection. Instead we focus on the choice of optimization algorithm.
- Finally, in order to keep this paper short, we consider that the family of functions $\mathcal{F}$ is linearly parametrized by a vector $w \in \mathbb{R}^d$. We also assume that x, y and w

are bounded, ensuring that there is a constant B such that $0 \leq \ell(f_w(x), y) \leq B$ and $\ell(\cdot, y)$ is Lipschitz.

We first explain how the uniform convergence bounds provide convergence rates that take the optimization error into account. Then we discuss and compare the asymptotic learning properties of several optimization algorithms.

2.1. Convergence of the Estimation and Optimization Errors

The optimization error $\mathcal{E}_{\text{opt}}$ depends directly on the optimization accuracy ρ. However, the accuracy ρ involves the empirical quantity $E_n(\tilde{f}_n) - E_n(f_n)$, whereas the optimization error $\mathcal{E}_{\text{opt}}$ involves its expected counterpart $E(\tilde{f}_n) - E(f_n)$. This section discusses the impact on the optimization error $\mathcal{E}_{\text{opt}}$ and of the optimization accuracy ρ on generalization bounds that leverage the uniform convergence concepts pioneered by Vapnik and Chervonenkis (e.g., [2].)

In this discussion, we use the letter c to refer to any positive constant. Multiple occurences of the letter c do not necessarily imply that the constants have identical values.

2.1.1. Simple Uniform Convergence Bounds

Recall that we assume that $\mathcal{F}$ is linearly parametrized by $w \in \mathbb{R}^d$. Elementary uniform convergence results then state that

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} |E(f) - E_n(f)|\right] \leq c\sqrt{\frac{d}{n}}\,,$$

where the expectation is taken with respect to the random choice of the training set.[2] This result immediately provides a bound on the estimation error:

$$\begin{aligned}\mathcal{E}_{\text{est}} &= \mathbb{E}\left[\ \big(E(f_n) - E_n(f_n)\big) + \big(E_n(f_n) - E_n(f^*_{\mathcal{F}})\big) + \big(E_n(f^*_{\mathcal{F}}) - E(f^*_{\mathcal{F}})\big)\ \right] \\ &\leq 2\,\mathbb{E}\left[\ \sup_{f\in\mathcal{F}} |E(f) - E_n(f)|\ \right]\ \leq\ c\sqrt{\frac{d}{n}}.\end{aligned}$$

This same result also provides a combined bound for the estimation and optimization errors:

$$\begin{aligned}\mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}} &= \mathbb{E}\big[E(\tilde{f}_n) - E_n(\tilde{f}_n)\big] + \mathbb{E}\big[E_n(\tilde{f}_n) - E_n(f_n)\big] \\ &\quad + \mathbb{E}\left[E_n(f_n) - E_n(f^*_{\mathcal{F}})\right] + \mathbb{E}\left[E_n(f^*_{\mathcal{F}}) - E(f^*_{\mathcal{F}})\right] \\ &\leq c\sqrt{\frac{d}{n}} + \rho + 0 + c\sqrt{\frac{d}{n}}\ =\ c\left(\rho + \sqrt{\frac{d}{n}}\right).\end{aligned}$$

Unfortunately, this convergence rate is known to be pessimistic in many important cases. More sophisticated bounds are required.

[2] Although the original Vapnik-Chervonenkis bounds have the form $c\sqrt{\frac{d}{n}\log\frac{n}{d}}$, the logarithmic term can be eliminated using the "chaining" technique (e.g., [10].)

2.1.2. Faster Rates in the Realizable Case

When the loss functions $\ell(\hat{y}, y)$ is positive, with probability $1 - e^{-\tau}$ for any $\tau > 0$, relative uniform convergence bounds state that

$$\sup_{f \in \mathcal{F}} \frac{E(f) - E_n(f)}{\sqrt{E(f)}} \leq c\sqrt{\frac{d}{n}\log\frac{n}{d} + \frac{\tau}{n}}\,.$$

This result is very useful because it provides faster convergence rates $\mathcal{O}(\log n/n)$ in the *realizable case*, that is when $\ell(f_n(x_i), y_i) = 0$ for all training examples (x_i, y_i). We have then $E_n(f_n) = 0$, $E_n(\tilde{f}_n) \leq \rho$, and we can write

$$E(\tilde{f}_n) - \rho \leq c\sqrt{E(\tilde{f}_n)}\,\sqrt{\frac{d}{n}\log\frac{n}{d} + \frac{\tau}{n}}\,.$$

Viewing this as a second degree polynomial inequality in variable $\sqrt{E(\tilde{f}_n)}$, we obtain

$$E(\tilde{f}_n) \leq c\left(\rho + \frac{d}{n}\log\frac{n}{d} + \frac{\tau}{n}\right)\,.$$

Integrating this inequality using a standard technique (see, e.g., [13]), we obtain a better convergence rate of the combined estimation and optimization error:

$$\mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}} = \mathbb{E}\left[E(\tilde{f}_n) - E(f^*_{\mathcal{F}})\right] \leq \mathbb{E}\left[E(\tilde{f}_n)\right] = c\left(\rho + \frac{d}{n}\log\frac{n}{d}\right)\,.$$

2.1.3. Fast Rate Bounds

Many authors (e.g., [10,4,12]) obtain fast statistical estimation rates in more general conditions. These bounds have the general form

$$\mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} \leq c\left(\;\mathcal{E}_{\text{app}} + \left(\frac{d}{n}\log\frac{n}{d}\right)^{\alpha}\;\right) \quad \text{for } \frac{1}{2} \leq \alpha \leq 1\,. \tag{4}$$

This result holds when one can establish the following variance condition:

$$\forall f \in \mathcal{F} \quad \mathbb{E}\left[\Big(\ell(f(X), Y) - \ell(f^*_{\mathcal{F}}(X), Y)\Big)^2\right] \;\leq\; c\;\left(\;E(f) - E(f^*_{\mathcal{F}})\;\right)^{2-\frac{1}{\alpha}}\,. \tag{5}$$

The convergence rate of (4) is described by the exponent α which is determined by the quality of the variance bound (5). Works on fast statistical estimation identify two main ways to establish such a variance condition.

- Exploiting the strict convexity of certain loss functions [12, theorem 12]. For instance, Lee et al. [14] establish a $\mathcal{O}(\log n/n)$ rate using the squared loss $\ell(\hat{y}, y) = (\hat{y} - y)^2$.
- Making assumptions on the data distribution. In the case of pattern recognition problems, for instance, the “Tsybakov condition” indicates how cleanly the posterior distributions $P(y|x)$ cross near the optimal decision boundary [11,12]. The realizable case discussed in section 2.1.2 can be viewed as an extreme case of this.

Despite their much greater complexity, fast rate estimation results can accomodate the optimization accuracy ρ using essentially the methods illustrated in sections 2.1.1 and 2.1.2. We then obtain a bound of the form

$$\mathcal{E} = \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}} = \mathbb{E}\left[E(\tilde{f}_n) - E(f^*)\right] \leq c\left(\mathcal{E}_{\text{app}} + \left(\frac{d}{n}\log\frac{n}{d}\right)^{\alpha} + \rho\right). \quad (6)$$

For instance, a general result with $\alpha = 1$ is provided by Massart [13, theorem 4.2]. Combining this result with standard bounds on the complexity of classes of linear functions (e.g., [10]) yields the following result:

$$\mathcal{E} = \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}} = \mathbb{E}\left[E(\tilde{f}_n) - E(f^*)\right] \leq c\left(\mathcal{E}_{\text{app}} + \frac{d}{n}\log\frac{n}{d} + \rho\right). \quad (7)$$

See also [15,4] for more bounds taking into account the optimization accuracy.

2.2. *Gradient Optimization Algorithms*

We now discuss and compare the asymptotic learning properties of four gradient optimization algorithms. Recall that the family of function $\mathcal{F}$ is linearly parametrized by $w \in \mathbb{R}^d$. Let $w^*_{\mathcal{F}}$ and w_n correspond to the functions $f^*_{\mathcal{F}}$ and f_n defined in section 1. In this section, we assume that the functions $w \mapsto \ell(f_w(x), y)$ are convex and twice differentiable with continuous second derivatives. Convexity ensures that the empirical const function $C(w) = E_n(f_w)$ has a single minimum.

Two matrices play an important role in the analysis: the Hessian matrix H and the gradient covariance matrix G, both measured at the empirical optimum w_n.

$$H = \frac{\partial^2 C}{\partial w^2}(w_n) \;=\; \mathbb{E}_n\left[\frac{\partial^2 \ell(f_{w_n}(x), y)}{\partial w^2}\right], \quad (8)$$

$$G = \mathbb{E}_n\left[\left(\frac{\partial \ell(f_{w_n}(x), y)}{\partial w}\right)\left(\frac{\partial \ell(f_{w_n}(x), y)}{\partial w}\right)'\right]. \quad (9)$$

The relation between these two matrices depends on the chosen loss function. In order to summarize them, we assume that there are constants $\lambda_{\max} \geq \lambda_{\min} > 0$ and $\nu > 0$ such that, for any $\eta > 0$, we can choose the number of examples n large enough to ensure that the following assertion is true with probability greater than $1 - \eta$:

$$\operatorname{tr}(G\,H^{-1}) \leq \nu \qquad \text{and} \qquad \text{EigenSpectrum}(H) \subset [\,\lambda_{\min}\,,\,\lambda_{\max}\,] \quad (10)$$

The condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ is a good indicator of the difficulty of the optimization [16].

The condition $\lambda_{\min} > 0$ avoids complications with stochastic gradient algorithms. Note that this condition only implies strict convexity around the optimum. For instance, consider the loss function ℓ is obtained by smoothing the well known hinge loss $\ell(z, y) = \max\{0, 1-yz\}$ in a small neighborhood of its non-differentiable points. Function $C(w)$ is then piecewise linear with smoothed edges and vertices. It is not strictly convex. However its minimum is likely to be on a smoothed vertex with a non singular Hessian. When we have strict convexity, the argument of [12, theorem 12] yields fast estimation rates $\alpha \approx 1$ in (4) and (6). This is not necessarily the case here.

The four algorithm considered in this paper use information about the gradient of the cost function to iteratively update their current estimate $w(t)$ of the parameter vector.

- **Gradient Descent (GD)** iterates

$$w(t+1) \;=\; w(t) - \eta \frac{\partial C}{\partial w}(w(t)) \;=\; w(t) - \eta \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w} \ell\big(f_{w(t)}(x_i), y_i\big)$$

where $\eta > 0$ is a small enough gain. GD is an algorithm with linear convergence [16]. When $\eta = 1/\lambda_{\max}$, this algorithm requires $\mathcal{O}(\kappa \log(1/\rho))$ iterations to reach accuracy ρ. The exact number of iterations depends on the choice of the initial parameter vector.

- **Second Order Gradient Descent (2GD)** iterates

$$w(t+1) \;=\; w(t) - H^{-1} \frac{\partial C}{\partial w}(w(t)) \;=\; w(t) - \frac{1}{n} H^{-1} \sum_{i=1}^{n} \frac{\partial}{\partial w} \ell\big(f_{w(t)}(x_i), y_i\big)$$

where matrix H^{-1} is the inverse of the Hessian matrix (8). This is more favorable than Newton's algorithm because we do not evaluate the local Hessian at each iteration but simply assume that we know in advance the Hessian at the optimum. 2GD is a superlinear optimization algorithm with quadratic convergence [16]. When the cost is quadratic, a single iteration is sufficient. In the general case, $\mathcal{O}(\log\log(1/\rho))$ iterations are required to reach accuracy ρ.

- **Stochastic Gradient Descent (SGD)** picks a random training example (x_t, y_t) at each iteration and updates the parameter w on the basis of this example only,

$$w(t+1) \;=\; w(t) - \frac{\eta}{t} \frac{\partial}{\partial w} \ell\big(f_{w(t)}(x_t), y_t\big).$$

Murata [17, section 2.2], characterizes the mean $\mathbb{E}_{\mathcal{S}}[w(t)]$ and variance $\mathbb{V}\mathrm{ar}_{\mathcal{S}}[w(t)]$ with respect to the distribution implied by the random examples drawn from the training set $\mathcal{S}$ at each iteration. Applying this result to the discrete training set distribution for $\eta = 1/\lambda_{\min}$, we have $\delta w(t)^2 = \mathcal{O}(1/t)$ where $\delta w(t)$ is a shorthand notation for $w(t) - w_n$.
We can then write

$$\begin{aligned} \mathbb{E}_{\mathcal{S}}[\, C(w(t)) - \inf C \,] &= \mathbb{E}_{\mathcal{S}}\big[\mathrm{tr}\big(H\, \delta w(t)\, \delta w(t)'\big)\big] + \mathrm{o}\big(\tfrac{1}{t}\big) \\ &= \mathrm{tr}\,\big(\, H\, \mathbb{E}_{\mathcal{S}}[\delta w(t)]\, \mathbb{E}_{\mathcal{S}}[\delta w(t)]' + H\, \mathbb{V}\mathrm{ar}_{\mathcal{S}}[w(t)] \,\big) + \mathrm{o}\big(\tfrac{1}{t}\big) \\ &\leq \tfrac{\mathrm{tr}(GH)}{t} + \mathrm{o}\big(\tfrac{1}{t}\big) \;\leq\; \tfrac{\nu\kappa^2}{t} + \mathrm{o}\big(\tfrac{1}{t}\big)\,. \end{aligned} \tag{11}$$

Therefore the SGD algorithm reaches accuracy ρ after less than $\nu\kappa^2/\rho + \mathrm{o}(1/\rho)$ iterations on average. The SGD convergence is essentially limited by the stochastic noise induced by the random choice of one example at each iteration. Neither the initial value of the parameter vector w nor the total number of examples n appear in the dominant term of this bound! When the training set is large, one could reach the desired accuracy ρ measured on the whole training set without even visiting all the training examples. This is in fact a kind of generalization bound.

Table 2. Asymptotic results for gradient algorithms (with probability 1). Compare the second last column (time to optimize) with the last column (time to reach the excess test error ϵ).
Legend: n number of examples; d parameter dimension; κ, ν see equation (10).

Algorithm	**Cost of one iteration**	**Iterations to reach ρ**	**Time to reach accuracy ρ**	**Time to reach $\boldsymbol{\mathcal{E} \leq c\,(\mathcal{E}_{\mathrm{app}} + \varepsilon)}$**
GD	$\mathcal{O}(nd)$	$\mathcal{O}\Big(\kappa \log \frac{1}{\rho}\Big)$	$\mathcal{O}\Big(nd\kappa \log \frac{1}{\rho}\Big)$	$\mathcal{O}\Big(\frac{d^2\,\kappa}{\varepsilon^{1/\alpha}} \log^2 \frac{1}{\varepsilon}\Big)$
2GD	$\mathcal{O}\big(d^2 + nd\big)$	$\mathcal{O}\Big(\log\log \frac{1}{\rho}\Big)$	$\mathcal{O}\Big(\big(d^2 + nd\big) \log\log \frac{1}{\rho}\Big)$	$\mathcal{O}\Big(\frac{d^2}{\varepsilon^{1/\alpha}} \log \frac{1}{\varepsilon} \log\log \frac{1}{\varepsilon}\Big)$
SGD	$\mathcal{O}(d)$	$\frac{\nu\kappa^2}{\rho} + \mathrm{o}\Big(\frac{1}{\rho}\Big)$	$\mathcal{O}\Big(\frac{d\nu\kappa^2}{\rho}\Big)$	$\mathcal{O}\Big(\frac{d\,\nu\,\kappa^2}{\varepsilon}\Big)$

The first three columns of table 2 report for each algorithm the time for a single iteration, the number of iterations needed to reach a predefined accuracy ρ, and their product, the time needed to reach accuracy ρ. These asymptotic results are valid with probability 1, since the probability of their complement is smaller than η for any $\eta > 0$.

The fourth column bounds the time necessary to reduce the excess error $\mathcal{E}$ below $c\,(\mathcal{E}_{\mathrm{app}} + \varepsilon)$ where c is the constant from (6). This is computed by observing that choosing $\rho \sim \big(\frac{d}{n} \log \frac{n}{d}\big)^{\alpha}$ in (6) achieves the fastest rate for ε, with minimal computation time. We can then use the asymptotic equivalences $\rho \sim \varepsilon$ and $n \sim \frac{d}{\varepsilon^{1/\alpha}} \log \frac{1}{\varepsilon}$. Setting the fourth column expressions to $T_{\max}$ and solving for ϵ yields the *best excess error achieved by each algorithm* within the limited time $T_{\max}$. This provides the asymptotic solution of the Estimation–Optimization tradeoff (3) for large scale problems satisfying our assumptions.

These results clearly show that the generalization performance of *large-scale learning systems* depends on both the statistical properties of the estimation procedure and the computational properties of the chosen optimization algorithm. Their combination leads to surprising consequences:

- *The SGD result does not depend on the estimation rate α.* When the estimation rate is poor, there is less need to optimize accurately. That leaves time to process more examples. A potentially more useful interpretation leverages the fact that (11) is already a kind of generalization bound: its fast rate trumps the slower rate assumed for the estimation error.
- *Superlinear optimization brings little asymptotical improvements in ε.* Although the superlinear 2GD algorithm improves the logarithmic term, the learning performance of all these algorithms is dominated by the polynomial term in $(1/\varepsilon)$. This explains why improving the constants d, κ and ν using preconditioning methods and sensible software engineering often proves more effective than switching to more sophisticated optimization techniques [18].
- *The SGD algorithm yields the best generalization performance despite being the worst optimization algorithm.* This had been described before [19] in the case of a second order stochastic gradient descent and observed in experiments.

In contrast, since the optimization error $\mathcal{E}_{\mathrm{opt}}$ of *small-scale learning systems* can be reduced to insignificant levels, their generalization performance is solely determined by the statistical properties of their estimation procedure.

Table 3. Results with linear SVM on the RCV1 dataset.

Model	Algorithm	Training Time	Objective	Test Error
Hinge loss, $\lambda = 10^{-4}$ *See [21,22].*	SVMLight	23,642 secs	0.2275	6.02%
	SVMPerf	66 secs	0.2278	6.03%
	SGD	**1.4 secs**	0.2275	6.02%
Logistic loss, $\lambda = 10^{-5}$ *See [23].*	LibLinear ($\rho = 10^{-2}$)	30 secs	0.18907	5.68%
	LibLinear ($\rho = 10^{-3}$)	44 secs	0.18890	5.70%
	SGD	**2.3 secs**	0.18893	5.66%

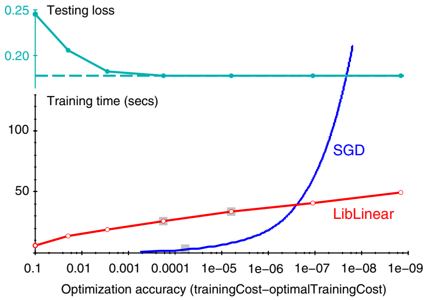


Figure 1. Training time and testing loss as a function of the optimization accuracy ρ for SGD and LibLinear [23].

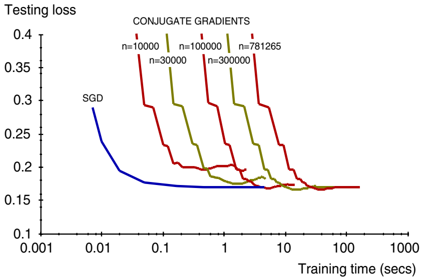


Figure 2. Testing loss versus training time for SGD, and for Conjugate Gradients running on subsets of the training set.

3. Experiments

This section empirically compares the SGD algorithm with other optimization algorithms on a well-known text categorization task, the classification of documents belonging to the CCAT category in the RCV1-v2 dataset [20]. Refer to `http://leon.bottou.org/projects/sgd` for source code and for additional experiments that could not fit in this paper because of space constraints.

In order to collect a large training set, we swap the RCV1-v2 official training and test sets. The resulting training sets and testing sets contain 781,265 and 23,149 examples respectively. The 47,152 TF/IDF features were recomputed on the basis of this new split. We use a simple linear model with the usual hinge loss SVM objective function

$$\min_{w} \; C(w,b) \;=\; \frac{\lambda}{2} + \frac{1}{n}\sum_{i=1}^{n} \ell(y_t(wx_t+b)) \quad \text{with} \;\; \ell(z) = \max\{0, 1-z\}\,.$$

The first two rows of table 3 replicate earlier results [21] reported for the same data and the same value of the hyper-parameter λ.

The third row of table 3 reports results obtained with the SGD algorithm

$$w_{t+1} = w_t - \eta_t \left(\lambda w + \frac{\partial \ell(y_t(wx_t+b))}{\partial w}\right) \quad \text{with} \;\; \eta_t = \frac{1}{\lambda(t+t_0)}\,.$$

The bias b is updated similarly. Since λ is a lower bound of the smallest eigenvalue of the hessian, our choice of gains η_t approximates the optimal schedule (see section 2.2).

The offset t_0 was chosen to ensure that the initial gain is comparable with the expected size of the parameter w. The results clearly indicate that SGD offers a good alternative to the usual SVM solvers. Comparable results were obtained in [22] using an algorithm that essentially amounts to a stochastic gradient corrected by a projection step. Our results indicates that the projection step is not an essential component of this performance.

Table 3 also reports results obtained with the logistic loss $\ell(z) = \log(1 + e^{-z})$ in order to avoid the issues related to the nondifferentiability of the hinge loss. Note that this experiment uses a much better value for λ. Our comparison points were obtained with a state-of-the-art superlinear optimizer [23], for two values of the optimization accuracy ρ. Yet the very simple SGD algorithm learns faster.

Figure 1 shows how much time each algorithm takes to reach a given optimization accuracy. The superlinear algorithm reaches the optimum with 10 digits of accuracy in less than one minute. The stochastic gradient starts more quickly but is unable to deliver such a high accuracy. However the upper part of the figure clearly shows that the testing set loss stops decreasing long before the moment where the superlinear algorithm overcomes the stochastic gradient.

Figure 2 shows how the testing loss evolves with the training time. The stochastic gradient descent curve can be compared with the curves obtained using conjugate gradients[3] on subsets of the training examples with increasing sizes. Assume for instance that our computing time budget is 1 second. Running the conjugate gradient algorithm on a random subset of 30000 training examples achieves a much better performance than running it on the whole training set. How to guess the right subset size a priori remains unclear. Meanwhile running the SGD algorithm on the full training set reaches the same testing set performance much faster.

4. Conclusion

Taking in account budget constraints on both the number of examples and the computation time, we find *qualitative differences* between the generalization performance of small-scale learning systems and large-scale learning systems. The generalization properties of large-scale learning systems depend on both the statistical properties of the estimation procedure and the computational properties of the optimization algorithm. We illustrate this fact by deriving asymptotic results on gradient algorithms supported by an experimental validation.

Considerable refinements of this framework can be expected. Extending the analysis to regularized risk formulations would make results on the complexity of primal and dual optimization algorithms [21,24] directly exploitable. The choice of surrogate loss function [7,12] could also have a non-trivial impact in the large-scale case.

Acknowledgments

Part of this work was funded by NSF grant CCR-0325463.

[3] This experimental setup was suggested by Olivier Chapelle (personal communication). His specialized variant of the conjugate gradients algorithm works nicely in this context because it converges superlinearly with very limited overhead.

References

[1] Leslie G. Valiant. A theory of learnable. *Proc. of the 1984 STOC*, pages 436–445, 1984.

[2] Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer Series in Statistics. Springer-Verlag, Berlin, 1982.

[3] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[4] Peter L. Bartlett and Shahar Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334, 2006.

[5] J. Stephen Judd. On the complexity of loading shallow neural networks. *Journal of Complexity*, 4(3):177–192, 1988.

[6] Richard O. Duda and Peter E. Hart. *Pattern Classification And Scene Analysis.* Wiley and Son, 1973.

[7] Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32:56–85, 2004.

[8] Clint Scovel and Ingo Steinwart. Fast rates for support vector machines. In Peter Auer and Ron Meir, editors, *Proceedings of the 18th Conference on Learning Theory (COLT 2005)*, volume 3559 of *Lecture Notes in Computer Science*, pages 279–294, Bertinoro, Italy, June 2005. Springer-Verlag.

[9] Vladimir N. Vapnik, Esther Levin, and Yann LeCun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

[10] Olivier Bousquet. *Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms.* PhD thesis, Ecole Polytechnique, 2002.

[11] Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statististics*, 32(1), 2004.

[12] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, March 2006.

[13] Pascal Massart. Some applications of concentration inequalities to statistics. *Annales de la Faculté des Sciences de Toulouse*, series 6, 9(2):245–303, 2000.

[14] Wee S. Lee, Peter L. Bartlett, and Robert C. Williamson. The importance of convexity in learning with squared loss. *IEEE Transactions on Information Theory*, 44(5):1974–1980, 1998.

[15] Shahar Mendelson. A few notes on statistical learning theory. In Shahar Mendelson and Alexander J. Smola, editors, *Advanced Lectures in Machine Learning*, volume 2600 of *Lecture Notes in Computer Science*, pages 1–40. Springer-Verlag, Berlin, 2003.

[16] John E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods For Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[17] Noboru Murata. A statistical study of on-line learning. In David Saad, editor, *Online Learning and Neural Networks.* Cambridge University Press, Cambridge, UK, 1998.

[18] Yann Le Cun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag, 1998.

[19] Léon Bottou and Yann Le Cun. Large scale online learning. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16.* MIT Press, Cambridge, MA, 2004.

[20] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[21] Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference*, Philadelphia, PA, August 2006. ACM Press.

[22] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated subgradient solver for SVM. In Zoubin Ghahramani, editor, *Proceedings of the 24th International Machine Learning Conference*, pages 807–814, Corvallis, OR, June 2007. ACM.

[23] Chih-Jen Lin, Ruby C. Weng, and S. Sathiya Keerthi. Trust region newton methods for large-scale logistic regression. In Zoubin Ghahramani, editor, *Proceedings of the 24th International Machine Learning Conference*, pages 561–568, Corvallis, OR, June 2007. ACM.

[24] Don Hush, Patrick Kelly, Clint Scovel, and Ingo Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. *Journal of Machine Learning Research*, 7:733–769, 2006.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-27

Practical Feature Selection: from Correlation to Causality

Isabelle Guyon [1]

Abstract. Feature selection encompasses a wide variety of methods for selecting a restricted number of input variables or "features", which are "relevant" to a problem at hand. In this chapter, we guide practitioners through the maze of methods, which have recently appeared in the literature, particularly for supervised feature selection. Starting from the simplest methods of feature ranking with correlation coefficients, we branch in various direction and explore various topics, including "conditional relevance", "local relevance", "multivariate selection", and "causal relevance". We make recommendations for assessment methods and stress the importance of matching the complexity of the method employed to the available amount of training data. Software and teaching material associated with this tutorial are available [12].

Keywords. Feature selection, variable selection, correlation, causality, filters, wrappers.

Introduction

Feature selection is a problem pervasive in all domains of application of machine learning and data mining: engineering applications, robotics and pattern recognition (speech, handwriting, face recognition), Internet applications (text categorization), econometrics and marketing applications and medical applications (diagnosis, prognosis, drug discovery). Restricting the input space to a (small) subset of available input variables has obvious economical benefits in terms of data storage, computational requirements, and cost of future data collection. It often also provides better data or model understanding and even better prediction performance. This chapter is directed to practitioners who seek a brief review of feature selection techniques and practical recommendations. We adopt a rather informal style to avoid introducing a lot of notations. For more formal developments and an extensive bibliography the readers will be referred, in the course of the chapter, to selected readings from two recently published books on the subject [11,19]. We begin by reviewing the goals and expectations of feature selection and the cases in which it may not be needed. Then, starting from the simplest feature ranking methods with correlation coefficients, we branch in various directions to introduce refinements, when then are needed. We finish by a discussion of the problem of model selection, emphasizing that many refinements come at the price of increasing the risk of overfitting.

[1] Address: 955 Creston Road, Berkeley, CA 94708, USA; E-mail: isabelle@clopinet.com.

1. Statement of the problem and motivations

In machine learning and data mining, there are many possible statements of the problem of feature selection. We place ourselves in this chapter in the context of *supervised learning*, in which data samples or "patterns" are recorded as vectors $\mathbf{x}$ of dimension N and a target variable y must be predicted. We will slightly generalize that framework in the course of the chapter and extend it to non-vectorial data representations and non-scalar targets. We will allude to the case of unsupervised feature selection, but not discuss it in details. In our simplified setting, pairs of training examples or training "patterns" $\{\mathbf{x}_i, y_i\}, i = 1, ...M$ are given in the form of a data matrix $\mathbf{X} = [x_{ij}](i = 1, ...M, j = 1, ...N)$ of dimensions (M, N), with column vectors X_j, and a target matrix $Y = [y_i]$ of dimension (M, 1). With a certain abuse of notation, we will also call Y the random variable (RV) whose realizations in training data are the y_i and we will call X_j the RV for feature j. Until Section 3.4 we will assume that data are i.i.d. (identically and independently distributed).

The primary goal we are pursuing is to make good predictions of the target variable y for new patterns $\mathbf{x}$. We call "predictor" the device making predictions, which may include a learning machine, whose parameters can be adjusted using training data. Prediction performances are evaluated with an objective function (or a "risk functional") on some test data, distinct from the training data. **The feature selection problem consists in making good predictions with as few variables/features as possible**.[2] In most of this chapter, we bring back the problem of feature selection to that of selecting a set of feature indices $\mathbf{S}$ of dimension N_s, corresponding to given columns of the data matrix, thus implicitly assuming that the same features are useful for making predictions *for any pattern*. In Section 2.5 we will generalize this framework to *local feature selection*, the problem of selecting features, which are most predictive for single patterns.

Feature selection is a multi-objective problem, which has been formalized in different ways, including minimizing a risk functional subject to using a number of selected features N_s lower than a given threshold; or minimizing N_s subject to keeping the risk lower than a given threshold. Such approaches, which include wrappers and embedded methods discussed in Section 3, require adopting efficient search strategies to explore the space of all possible feature subsets. Other approaches rank first the features with an ad hoc criterion and then optimize prediction performances by considering nested subsets of features built from that ordering. Two central questions must be answered before tackling a new feature selection problem:

1. What do we want to achieve with feature selection?
2. Do we really need it?

Reasons for performing feature selection include:

- Improving performance prediction.
- Reducing computational requirements.
- Reducing data storage requirements.
- Reducing the cost of future measurements.
- Improving data or model understanding.

[2] I this chapter, we will not make a distinction between "variable" and "feature". Such distinction may be relevant though for algorithms building features from the original variables.

The analysis of the results of recently organized benchmarks [10,13] reveals that feature selection is seldom needed to improve prediction performance. A standard argument in favor of feature selection is that it is needed to "overcome the curse of dimensionality". For cases in which the data matrix is skinny ($N \gg M$), many more features than training examples), it may seem that feature selection is a requirement to avoid the unfavorable case in which the number of free parameters of the model greatly exceeds the number of training examples, a case known to yield poor "generalization" (poor performance on test data) [25]. But, today's state-of-the-art machine learning algorithms, which have harnessed the problem of overfitting using powerful "regularization" techniques **do not require feature selection as a preprocessing** step to perform well. These include regularized kernel methods (such as Support Vector Machines (SVM), kernel ridge or logistic regression, and Gaussian processes) and ensemble methods (such as boosting and bagging [4]). These algorithms are described in standard textbooks [6,16] and software is freely available on-line. In fact, in many cases, feature selection is more harmful than useful because the process of feature selection is data hungry and the more training data are used for feature selection, the less are available to perform model parameter estimation. So, improving performance prediction may, after all, not be the main charter of feature selection. Rather, we may seek, for various other reasons, to limit the number of features used, without *significantly* degrading performance.

2. Getting started: Feature ranking and nested subset methods

In this section, we tackle a first goal: reducing the number of features while least degrading performance. We do not necessarily seek to minimize the number of selected features. In particular, we may satisfy ourselves with eliminating "irrelevant" features, and leaving some redundancy among the selected features. Eliminating redundancy will be addressed, to some extent, at the end of this section.

2.1. Feature ranking

To achieve our initial goal, we advocate the use of simple *univariate feature ranking with correlation coefficients*, sometimes referred to as "filter" methods. Feature selection is then performed by constructing nested subsets of features of increasing size, starting from a subset of one feature (the most correlated to the target), and progressively adding features (less and less correlated with the target). Models are built with every feature subset and evaluated by cross-validation. A standard cross-validation method is 10-fold cross-validation. We recommend using 10 times 10-fold cross-validation to reduce the variance. We also recommend that a separate test set (not used for training or cross-validation) be reserved for a final model testing. The suggested procedure is summarized in Figure 1. At the expense of additional computational burden, the whole procedure can be repeated multiple times (in an outer cross-validation loop) to reduce the variance of performance prediction. Such simple filter methods have proved to work well in practice [7].

In step 4 of the algorithm, a number of state-of-the-art predictive models are suitable, including SVM, kernel ridge regression, boosting and Random Forests. All the models cited perform very well in challenges [10,13]. The choice may depend on computational considerations.

1. **Reserve test data**: Split the available data into training matrices $\{X, Y\}$ and a test matrices $\{Xt, Yt\}$. A column X_j of X represents the values taken by the j^{th} feature for all the patterns.
2. **Choose criterion**: Choose a ranking statistic $R(X_j, Y)$ (*e.g.* the absolute value of the Pearson correlation coefficient).
3. **Form subsets**: Rank the features: $R(X_{r_1}, Y) \leq R(X_{r_2}, Y) \leq ... \leq R(X_{r_N}, Y)$; and form nested subsets of features: $\mathbf{S}_1 = \{r_1\}, \mathbf{S}_2 = \{r_1, r_2\}, ..., \mathbf{S}_N = \{r_1, r_2, ..., r_N\}$.
4. **Cross-validate**:
   - Split the training data many times (*e.g.* $9/10^{th}$ for training and $1/10^{th}$ for testing), each split being called a "fold".
   - Train models for each feature subset $\mathbf{S}_h$ and each data fold k and test them to obtain the performance $CV(h, k)$.
   - Average $CV(h, k)$ over all folds to obtain $CV(h)$.
   - Select the smallest feature set h^* such that $CV(h)$ is optimal or near optimal (according to a predetermined threshold).
5. **Evaluate on test data**: Restricting the data to the feature subset $\mathbf{S}_{h^*}$, train on a model the entire training set $\{X, Y\}$ and test it on the test set $\{Xt, Yt\}$.

Figure 1. Feature ranking "filter" method.

2.2. *Correlation*

In step 2 of the algorithm, a ranking criterion must be chosen. The absolute value of the Pearson correlation coefficient is a widely used ranking criterion, which we call PCC:

$$R_{PCC}(X_j, Y) = abs(\frac{(X_j - \overline{X}_j) \cdot (Y - \overline{Y})}{\sqrt{(X_j - \overline{X}_j)^2}\sqrt{(Y - \overline{Y})^2}}) \,. \tag{1}$$

The PCC is comprised between 0 and 1. It is proportional to the dot product between X_j and Y, after variable standardization (subtracting the mean and dividing by the standard deviation). The PCC is applicable to binary and continuous variables or target values, which makes it very versatile. Categorical variables can be handled by using a complete disjunctive coding (*e.g.* for 4 values [1 0 0 0], [0 1 0 0], [0 0 1 0], [0 0 0 1]).

The PCC is a measure of **linear dependency** between variables. Irrelevant variables (independent of the target) should have a PCC value near zero, but a value of the PCC near zero does not necessarily indicate that the variable is irrelevant: a non-linear dependency may exist, which is not captured by the PCC. It is standard to perform simple data preprocessing to reduce the impact of data non-linearities, *e.g.* taking the logarithm or the square root of the features. Preprocessing is often guided by visual inspection of the data, but it can be subjected to cross-validation. Two simple modifications of the PCC can be made to add robustness against outliers and alleviate the need of preprocessing:

- **Jacknife correlation coefficient**: One large outlying value may result in a spurious high correlation. A standard technique to avoid this problem is to repeat the computation of the correlation coefficient M times, each time removing one value, then averaging the results. The result is called "jacknife correlation coefficient".

- **Spearman correlation coefficient**: Rather than finding an appropriate non-linear scaling of the variables, one may reduce the effect of non-linearities by replacing the variable values by their rank. After this preprocessing, the Pearson correlation coefficient is equivalent to the Spearman correlation coefficient.

Many criteria are closely related to the PCC. For instance the T-statistic, the so-called signal-to-noise-ratio [8], and the F-statistic all give similar results for classification problems, although differences in normalization differentiate when classes are unbalanced. In the following sections, we are branching in several directions to overcome the limitations of feature ranking with the PCC.

2.3. Mutual Information

We began this section by presenting the problem of feature selection as that of removing "irrelevant" variables. Irrelevance of individual variables to the target may be defined in terms of random variable independence:

$$P(X_j, Y) = P(X_j)P(Y) \ , \tag{2}$$

this time denoting by "X_j" and "Y" the random variables distributed like the j^{th} feature and like the target variable. We use in the following the notation $(X_j \perp Y)$ to denote independence and $(X_j \not\perp Y)$ to denote dependence.

The Kullback-Leibler divergence, which measures the discrepancy between $P(X_j, Y)$ and $P(X_j)P(Y)$ can be used as a measure of irrelevance. This is nothing but the well-know mutual information criterion (MIC). For discrete variables, the MIC is defined as:

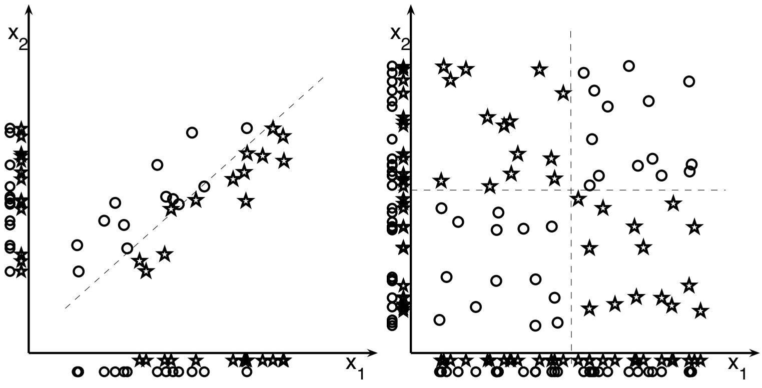
$$R_{MIC}(X_j, Y) = \sum_{x,y} P(X_j = x, Y = y) \log \frac{P(X_j = x, Y = y)}{P(X_j = x)P(Y = y)} \ . \tag{3}$$

The MIC is a positive quantity, which may be estimated by frequency counts for discrete variables. The advantage of using the MIC rather then the PCC is that dependency is no longer reduced to linear dependency. The drawback is the MIC is harder to estimate than the PCC, particularly for continuous variables. Other non-linear ranking criteria are reviewed in reference [11], Chapter 3.

2.4. Conditional relevance

Our first take at feature selection was to rank features according to their individual relevance to the target. Yet, some **individually irrelevant features may become relevant in the context of others**. This is illustrated in Figure 2 by scatter plots of two-dimensional binary classification problems.

It is commonly thought that feature ranking techniques cannot solve problems of relevance "in context". But actually, ranking criteria taking into account the context of other features can be defined, using the notion of **conditional dependency**. For instance, in Figure 2-a, X_2 is independent of Y but it is conditionally dependent, given X_1. Using conditional mutual information has been advocated by several authors (see reference [11], Chapter 6). However, when $N \gg M$ conditioning on more than a few variables is impractical for computational and statistical reasons.



(a) $X_2 \perp Y, X_2 \not\perp Y|X_1$ (b) $X_1 \perp Y, X_2 \perp Y, \{X_1, X_2\} \not\perp Y$

Figure 2. Multivariate dependencies. (a) Feature X_2 is individually irrelevant to Y ($X_2 \perp Y$), but it becomes relevant in the context of feature X_1 ($X_2 \not\perp Y|X_1$). (b) Two individually irrelevant features ($X_1 \perp Y$ and $X_2 \perp Y$) become relevant when taken jointly ($\{X_1, X_2\} \not\perp Y$).

Instead, one may use the Relief criterion [17] a heuristic criterion based on nearest neighbors, which works well in practice for classification problems:

$$R_{Relief}(j) = \frac{M(j)}{H(j)} = \frac{\sum_{i=1}^{M} \sum_{k=1}^{K} |x_{i,j} - x_{M_k(i),j}|}{\sum_{i=1}^{M} \sum_{k=1}^{K} |x_{i,j} - x_{H_k(i),j}|} , \tag{4}$$

To evaluate the criterion, first identify in the original feature space, for each example $\mathbf{x}_i$, the K closest examples of the same class $\{\mathbf{x}_{H_k(i)}\}, k = 1...K$ (nearest hits) and the K closest examples of a different class $\{\mathbf{x}_{M_k(i)}\}$ (nearest misses). All features are used to compute the closest examples. Then, in projection on feature j, the sum $M(j)$ of the distances between the examples and their nearest misses is compared to the sum $H(j)$ of distances to their nearest hits. In Equation 4, we use the ratio of $M(j)$ and $H(j)$ to create a criterion independent of feature scale variations. The difference $M(j) - H(j)$ was proposed in the original paper.

The Relief method works for multi-class problems, and it can be extended to continuous targets. One possible extension of Relief to continuous targets is to replace distances by similarities or "kernels" and define a "kernel alignment criterion", inspired by the "kernel alignment" idea [5]:

$$R_{Align}(j) = \sum_i \sum_k \kappa(y_i, y_k) k_j(\mathbf{x}_i, \mathbf{x}_k) K(\mathbf{x}_i, \mathbf{x}_k) , \tag{5}$$

where $[\kappa(y_i, y_k)]$ is a (M, M) similarity matrix between target values (*e.g.* the simple outer product YY^T or the correlation matrix), $[K(\mathbf{x}_i, \mathbf{x}_k)]$ is a (M, M) similarity matrix between patterns, and $[k_j(\mathbf{x}_i, \mathbf{x}_k)]$ is a (M, M) similarity matrix between patterns, projected on features j. The kernel alignment criterion is proportional to the square of the Pearson correlation coefficient if $K(\mathbf{x}_i, \mathbf{x}_k) \equiv 1$ and $[\kappa(y_i, y_k)]$ and $[k_j(\mathbf{x}_i, \mathbf{x}_k)]$ are the correlation matrices. For classification, it can be thought of a Parzen windows version of Relief. It is closely related to the Hilbert-Schmidt criterion [9]. Many other extensions of Relief have been proposed, see reference [19], Chapter 9, for a review.

2.5. Local relevance

Another extension of the Relief methodology is to select variables relevant to given patterns, rather than variables relevant to all patterns. This is the concept of "local feature selection", which is illustrated by the following criterion, in the case of classification problems:

$$R_{Local}(i,j) = \frac{\sum_{k=1}^{K} |x_{i,j} - x_{M_k(i),j}|}{\sum_{k=1}^{K} |x_{i,j} - x_{H_k(i),j}|} . \tag{6}$$

Compared to formula 4, we have suppressed the summation over all patterns. A similar extension of formula 5 can be made. Local feature selection can be married with many kernel methods for building predictive models, since kernel methods are based on comparisons of a new example to stored patterns, and are amenable to using a different feature set for each pattern. Other examples of local feature selection methods are found in reference [19], Chapter 11. The concept of local feature selection lends itself to generalizing feature selection to patterns not composed of a fixed number of features, like variable length strings or graphs, since it relies only on pairwise comparisons between patterns. However, despite its attractiveness it remains under-explored, one primary reason being that it is very difficult to overcome the overfitting problem.

2.6. Forward selection and backward elimination

Up to now, we have not been preoccupied by the problem of feature redundancy, we have only tried to eliminate irrelevant features. We now examine methods, which eliminate feature redundancy to some extent. The basic principle is to use conditioning on subsets of other variables. Two approaches can be taken:

- **Forward selection**: Starting from an empty subset of variables, add progressively variables, which are relevant, given previously selected variables.
- **Backward elimination**: Starting from all variables, remove progressively variables, which are irrelevant, given the remaining selected variables.

Conditional mutual information has been advocated by several authors (for a review of information-theoretic methods, see reference [11], Chapter 6). However, it is plagued by the difficulty of estimating mutual information. An alternative method, particularly useful in practice, is the **Gram-Schmidt orthogonalization** procedure. It extends the PCC ranking to **forward selection** by adding progressively variables, which correlate to the target, in the space orthogonal to the variables already selected. See Appendix A for a Matlab code implementation of [22]. A typical example of a **backward elimination** procedure is the RFE SVM (Recursive Feature Elimination Support Vector machine) [14]. For the linear SVM, it boils down to training first on all features, eliminating the feature with smallest weight in absolute value, and iterating until no feature remains. A non-linear version extends it to the non-linear SVM. The procedure bears a lot of similarity with the *unsupervised* "gene shaving" method [15], which uses eigenvalues in lieu of weights to perform the elimination procedure. One also easily sees that the Relief criterion and the kernel alignment criterion (Equations 4 and 5) can be extended to do forward selection and backward elimination. It suffices to replace the projection onto a single feature by a projection on a subset of features.

Both forward selection (FS) and backward elimination (BE) algorithms yield nested subsets of features. Thus, they are similar in spirit to feature ranking algorithms and selecting the optimum number of features can be performed by the procedure outlined in Figure 1. The two types of methods have different advantages and disadvantages. BS selects features using the context of all features not eliminated so far and therefore is more capable of finding complementary features than FS. However, for BS a catastrophic performance degradation is often observed for the smallest nested subsets. In contrast, for FS, even the smallest nested subsets are predictive, which is an advantage if one wants to trade *performance* for *number of features*.

3. Moving on: Optimal subset search

Nested subset methods described in the previous section are sub-optimal with respect to finding the most predictive feature subset because they are greedy methods (once a feature has been removed or added, its inclusion in the selected feature set is not questioned again). We examine now strategies in which the entire space of feature subsets is searched for an optimal feature subset. Before undertaking such an endeavor, we must define a criterion of optimality for feature subsets. The notion of conditional variable independence and "Markov blankets" provides us with such a criterion.

3.1. Markov blankets

Following [20], we define a **Markov blanket** of Y as a subset $\mathbf{M}$ of random variables included in the set $\mathbf{X}$ of all variables such that Y is conditionally independent of all other variables given $\mathbf{M}$. In other words, $\mathbf{M}$ in an Markov blanket *iff*:

$$Y \perp (\mathbf{X} - \mathbf{M}) \mid \mathbf{M} .$$

A minimal Markov blanket is called a **Markov boundary** (MB) [20]. There are various other Markov blanket/boundary definitions and Markov boundaries are not necessarily unique, but to simplify the discussion, we will talk here about "the MB". The conditional independence between the target variable and the other variables given the MB implies that all the other variables are truly non informative once the MB is given and can therefore be discarded. This makes the MB a good candidate for an optimal feature subset.

Kohavi and John [18] make a distinction between strong and weak relevance. Tsamardinos and collaborators [24] proved that, under certain conditions, the MB is the set of strongly relevant features. Other "relevant" features, in the sense that adding them to a subset of previously selected features changes the posterior probability of Y, are called "weakly relevant".

The MB captures a certain notion of relevance, but it is not universally optimum. Kohavi and John [18] make a distinction between "relevance" and "usefulness". **Not all strongly relevant features (belonging to the MB) are necessarily "useful" with respect to a given objective**.

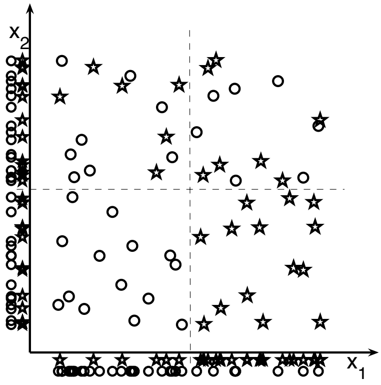


Figure 3. Objective-dependent relevance. (a) Feature X_2 is individually irrelevant to Y ($X_2 \perp Y$). But, it is "relevant" in the context of feature X_1 ($X_2 \not\perp Y|X_1$), with respect to the estimation of posterior probabilities: $P(Y|X_1, X_2 = x_2) \neq P(Y|X_1)$ for half of the values of x_2. However, feature X_2 does not contribute to improving classification (the vertical dashed line is the optimum Bayes classification boundary). Hence different objectives yield different feature relevance criteria.

An illustrative example for a two-class classification problem is given in Figure 3. By construction, the examples at the *top* of the figure are labeled randomly, while the examples at the *bottom left* of the figure all belong to one class, and those at the *bottom right* to the other class. Hence, feature X_2 is instrumental in determining the posterior probability distribution of Y. Yet, the optimum Bayes decision boundary is a vertical line, *i.e.* feature X_2 is not helpful to make optimal classification decisions. In this example, if the objective is to minimize classification error, X_2 is not useful, but if the objective is to predict the distribution of Y, it is useful. Even more interestingly, some "useful" features cannot be called "relevant" in any reasonable sense. For instance, for a linear predictor, a feature clamped to a constant value can be "useful" as a means of introducing a bias value, but can hardly be called "relevant".

Thus, even though efficient algorithms to compute the MB have been devised (*e.g.* [24]), it may be preferable to resort to algorithms, which directly optimize the feature subset for given predictors and objective functions. Those will now be reviewed.

3.2. Wrappers

Wrappers are methods, which use any machine learning algorithm as a black box, and search the space of all possible feature subsets to build a predictor with optimum performances. The methodology differs from the step-by-step procedure outlined in Figure 1 only in steps 2 and 3:

2. **Choose criterion**: Choose a search strategy to navigate in the space of 2^N feature subsets.
3. **Form subsets**: Select an ensemble of feature subsets to be investigated or generate new feature subsets iteratively until a stopping criterion is met.

Many search strategies exist, including exhaustive search, beam search, branch and bound, genetic algorithms, simulated annealing, greedy search, to only name a few. For a review, see reference [11], Chapter 4. Extensive search strategies are prone to overfitting for reasons, which will be explained in Section 4.1. Thus, unless a lot of training data are available, the best performing methods are closely related to "greedy search". Greedy search methods include forward selection and backward elimination procedures, which we have already mentioned in Section 2.6. These two approaches have been combined to increase the effectiveness of search, by alternating backward elimination and forward selection, until a stopping criterion is met.

3.3. Embedded methods

Embedded methods refer to a wide variety of techniques in which training data are given to a learning machine, which returns a predictor **and** a subset of features on which it performs predictions. Feature selection is performed in the process of learning, which saves the trouble of a two-step induction process.

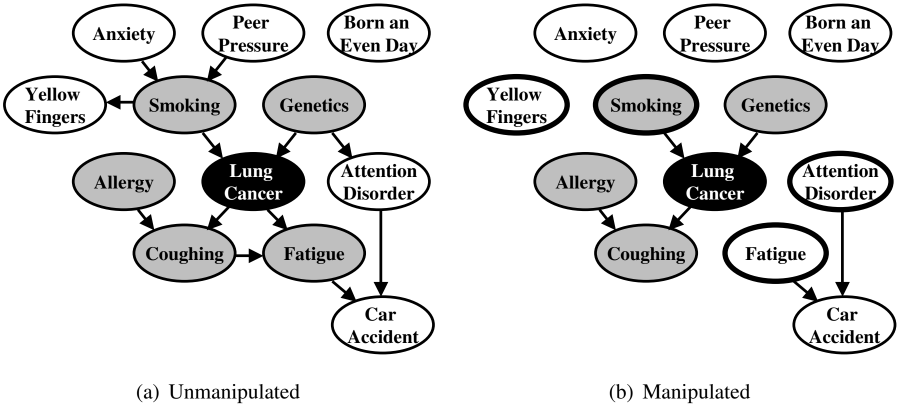
Embedded methods include algorithms, which optimize a regularized risk functional J with respect to two sets of parameters: the parameters of the learning machine α, and parameters $\sigma \in \{0,1\}^N$ indicating which features are selected. Typically, such algorithms alternate two steps until convergence:

$$min_\alpha \; J(\alpha, \sigma) \tag{7}$$

$$min_\sigma \; J(\alpha, \sigma) \; . \tag{8}$$

To facilitate the task of the learning algorithms and allow using parameter search methods like gradient descent, the indicator vector $\sigma \in \{0,1\}^N$ is sometimes replaced by continuous scaling factors $\sigma \in [0,1]^N$ (see reference [11], Chapter 5). A threshold on the scaling factors must be applied when the algorithm terminates selecting features. RFE SVM already mentioned in Section 2.6, may also be considered an embedded approach since it alternates training an SVM and eliminating the feature that least degrades the risk functional (corresponding simply to the feature with smallest weight in absolute value, for the linear SVM). There is a similar method called "multiplicative updates" or zero-norm method, which consists in training a regular SVM, re-scaling its inputs with the absolute values of the weights, and iterating until convergence [27]. This method can be shown to approximately minimize the zero-norm (defined as the number of selected features), under the constraint that training examples are well classified. The regular SVM, which minimizes the two-norm (Euclidean norm) does not have an embedded mechanism of feature selection. But another variant, which minimizes the one-norm (sum of the absolute value of the weights) also has the property that some of the weights are automatically driven to zero, and therefore qualifies as an embedded method [1,23]. One advantage of these methods is that they converge to an optimum feature subset and do not require performing cross-validation. See Section 4.3 for details.

The above mentioned methods have the flavor of *backward elimination* procedures. Other methods proceed in a *forward selection* manner. For example, tree classifiers (*e.g.* [21]) iteratively select variables to partition the data such that the data sub-divisions progressively contain a lesser variety of class labels. The RF method, which has been successful in challenges [4,10,13], follows this approach. Thus, a variable is selected only if it is "relevant", conditioned on previously selected variables. Data grid models [2] also partition data like trees, but not hierarchically. They provide a piecewise constant approximation to the data distribution. In the course of learning, variables whose optimal discretization consist in a single interval, are deemed uninformative and naturally eliminated.



(a) Unmanipulated

(b) Manipulated

Figure 4. Manipulations. The gray shaded nodes represents the Markov boundary of the target variable: "Lung Cancer". In the manipulated graph (b), the four manipulated nodes ("Yellow Fingers", "Smoking", "Attention Disorder", and "Fatigue") are emphasized. As a result of being manipulated, they are disconnected from their original causes. The Markov boundary in graph (b) no longer includes the "Fatigue" node.

3.4. Causal feature selection

Thus far, we have always assumed that data are i.i.d. and, in particular, that the training and test sets are distributed similarly. This is often NOT the case in practice. We consider in this section a particular case of distribution shift, which results from "manipulations" of the data by an "external agent". This gives rise to the distinction between features, which are "causes" of the target, and features, which are "consequences" of the target.

Motivation. In the case of i.i.d. data, feature selection is not concerned with causality: causes and consequences of the target are both predictive. For example, to predict lung cancer, smoking (a possible cause) may be as predictive as coughing (a possible consequence). However, **acting** on a cause may result in a change in the target while acting on a consequence will not. Oftentimes experiments are required to determine causal relationships. But many experiments are impractical, costly or unethical. For example, preventing people from smoking may be costly, forcing them to smoke would be unethical. For this reason, it is important to develop algorithms capable of "causal feature selection" to select variables, which truly influence the target when actions are performed. For instance, we would like to know, before enforcing a new policy preventing to smoke in public place, whether this will be beneficial to public health.

Markov blankets revisited. Bayesian networks are often used to uncover causal relationships. Informally, a Bayesian network (BN) is a graph whose nodes are random variables and vertices represent conditional probabilities. The Markov condition states that a node is independent of non-descendants given its parents. Hence, the data of $P(X|parents(X))$ is sufficient to calculate all conditional variable dependencies. Causal Bayesian networks are Bayesian networks in which arcs are interpreted as causal relationships. In a wide range of practical cases, the Markov boundary (MB) (see Section 3.1) coincides with the set of parents (direct causes), children (direct effects), and spouses (direct causes of direct effects). We show an example in Figure 4-a.

If test data are drawn from the same distribution as training data, the MB does not change and its optimality with respect to making predictions remains the same. However, if test data are manipulated (Figure 4-a) *i.e.* some nodes are set to given values by an external agent and therefore disconnected from their original causes, the MB shrinks to a subset of the original MB. With only the knowledge of which features are going to be manipulated and without actually seeing any test data, if the edges of the network are properly oriented to indicate causal relationships, we can infer which nodes of the MB will no longer be predictive features. Note that, in this framework, the target is NOT manipulated. Hence, no matter which manipulations are performed to the variables, the direct causes of Y will always be predictive. In particular, if **all** variables (other than Y) are manipulated, **only** the direct causes of Y are predictive.

There exist efficient algorithms to determine the local causal neighborhood of the target, see reference [19], Chapter 4, for a review. It should be noted though that several causal graphs may be consistent with the data distribution and therefore not all causal relationships can be determined from "observational data" alone. One must resort to performing some experiments, if such ambiguous causal relationships are to be resolved.

4. Assessment methods

In this section, we discuss methods of model selection and performance prediction.

4.1. Cross-validation

In previous sections, we have advocated cross-validation (see Figure 1). But we have not made recommendations for the value of K in K-fold cross-validation beyond mentioning that most practitioners choose the one-size-fit-all value $K = 10$. This section sheds light on this problem.

For simplicity of the discussion, we consider a single split of the training data into m training examples and p validation examples, $m + p = M$ (for a total of M training examples). This could eventually be repeated K times in a cross-validation loop. As before, we pursue the goal of obtaining best "generalization" performance, *i.e.* performance on "future" data, represented by the separate test set, which we reserved and do not touch during training. Feature selection is cast into a two-level inference problem: the training set is used to adjust the parameters of the predictive model, whereas the validation set is used to select the feature set. We are interested in studying the statistical complexity of the **second level of inference** (feature selection). That level consists in selecting among a **discrete number of predictive models**, corresponding to all the feature subsets under investigation. Because we select among a discrete number of models, the following type of bound applies, for classification problem [25]:

$$E_{gene} \leq E_{valid} + f(\frac{logN_s}{p}) \ , \tag{9}$$

where E_{gene} is the generalization error, N_s is the number of selected feature, p is the number of validation examples, E_{valid} the validation set error, and $f(.)$ is a non-decreasing function of $logN_s/p$, which also depends on the confidence with which the bound applies. Importantly, $logN_s$ is a measure of complexity of the feature selection problem since the ratio $logN_s/p$ governs generalization performances. In that respect, N_s should be made as small as possible and p as large as possible.

Table 1. Complexity of feature selection methods.

Method	**Number of subsets tried:** N_s	**Complexity:** $O(\log N_s)$
Feature ranking and nested subset methods	N	$\log N$
Greedy wrappers	$N(N+1)/2$	$\log N$
Exhaustive search wrappers	2^N	N

Unfortunately increasing p means reducing the number of training examples m and reducing N_s means reducing our chances of finding the optimal feature subset, both of which might result in increasing E_{valid}. So this might not have the expected beneficial effect on E_{gene}.

Table 1 gives an order of magnitude of the complexity of a few feature selection methods. **A rule-of-thumb to avoid overfitting at the second level of inference is that** $logN_s$ **must be commensurate to** p. It easily understood why nested subset methods and greedy wrappers performing forward selection or backward elimination run much less at risk of overfitting than exhaustive search wrappers: they require only $p = O(logN)$ examples instead of $p = O(N)$ examples. This is why they have been widely deployed in applications such as genomics and proteomics, text processing, and imaging where the number of features is in the tens of thousands, while the number of training examples remains orders of magnitude smaller. Wrapper methods, which explore an number of subsets exponential in the number of features, are only suitable if the number of training examples exceeds the number of features.

We examine in the next sections alternatives to two-level of inference methods, which save the burden of splitting the training data, and are computationally less expensive than cross-validation.

4.2. Statistical tests

Statistical tests are often used to assess the significance of individual features for feature ranking methods, in the following way. The ranking criterion R is thought of as a statistic, which is distributed in a certain way for irrelevant features. **If for a given feature** j**,** $R(j)$ **significantly departs from the expected value of** R **for irrelevant features, that feature is called "relevant"**. The "null hypothesis" to be tested is that the feature is irrelevant. This method is easily applied to ranking criteria, which are already tabulated statistics, like the Pearson correlation coefficient, the T-statistic, the F-statistic, the G-statistic, etc. Other ranking criteria like the Relief criterion or the "kernel alignment" criterion (Section 2.4), are not tabulated statistics. For these, we can emulate the distribution of irrelevant features with artificially generated "probe" variables, which are added to the original data matrix. Typically, one uses permutations of the values of the columns of the data matrix as probes and the pvalue of the test for a given (real) feature j is approximated by:

$$pval(j) = \frac{N_{sp}}{N_p} \, ,$$

where N_{sp} is the number of selected probes, *i.e.* the number of probes having a value of R greater than $R(j)$, and N_p is the total number of probes. Obviously, the larger the number of probes, the better the approximation. With this method, one is left with the choice of setting a threshold on the pvalue to select the most significant features: small pvalues shed doubt on the validity of the null hypothesis that the feature is irrelevant, *i.e.* relevant features should have small pvalues, which translates in few "probes" having a higher rank. The choice of the threshold of significance must take into account the problem of multiple testing, because we generally select simultaneously multiple features. This can be handled with the simple Bonferroni correction, which consist in multiplying the pvalue by N. However, this correction is generally too conservative and one prefers setting a threshold on the "false discovery rate", the *fraction of features falsely called significant among the selected features*, which is estimated by:

$$FDR(j) = pvalue(j)\frac{N}{N_s} ,$$

where N is the total number of features and N_s the number of selected features. See reference [11] Chapter 2 for details.

Statistical tests may also be used to assess the significance of features for multivariate methods, which select an optimal feature subset rather than ranking individual features. The model built from the set of selected features is taken as reference or "null model". A test is performed on the significance of the difference between the null model and a model built with one feature removed. Closed form tabulated statistics can sometimes be derived. The example of the Gram-Schmidt method is treated in reference [11] Chapter 2.

We do not generally recommend using statistical tests for selecting an optimum number of features in lieu of using cross-validation because there is usually little correlation between the prediction performance of the predictor and the fraction of false positive. Rather, the FDR is a useful complementary feature subset quality control indicator.

4.3. Penalty-based methods

Statistical tests are a disguised way of using the "training error" for model selection. In that respect, it is not surprising that they should not perform as well as cross-validation in most cases. There is another alternative to cross-validation, which lumps the two levels of inference into one: penalty-based methods. The general idea is the following: The "training error" is overly optimistic to evaluate model performance (since it has been minimized over the parameters of the model); hence, to become useful, it must be augmented by a penalty term (see *e.g.* [25]). Two types of penalty-based methods are presently most popular: Structural Risk Minimization (SRM) and Bayesian methods. Even though they are based on different theoretical arguments, they often end up with similar penalty terms, which penalize more complex models.

A classical example is that of weight "shrinkage". For a linear predictor $f(\mathbf{x}) = \mathbf{w}\mathbf{x} + b$, the SRM method advocates building a structure to rank models in order of increasing complexity, based on the Euclidean norm (or 2-norm) of the weight vector $\|\mathbf{w}\|_2$. In the Bayesian framework, this is equivalent to choosing a prior on the weights centered on $\|\mathbf{w}\|_2 = \mathbf{0}$. The resulting penalty term is proportional to $\|\mathbf{w}\|_2^2$. Depending on the choice of the norm and of the loss function, various methods are obtained, including ridge regression and Support Vector Machines (SVM) for the 2-norm, lasso and 1-norm SVM for the 1-norm, and 0-norm SVM for the 0-norm. See reference [11] Chapter 1 for other examples and more details. As previously mentioned in Section 3.3, the 2-norm methods do not perform feature selection while the 1-norm and 0-norm methods do. These last 2 types of methods converge to an optimal feature subset in the process of training. Some authors have reported that penalty-based methods may overfit in some sense by selecting too few features. Cross-validation may be used as a halting criterion to overcome this problem [27]. But then of course the benefit of penalty-based method is partially lost.

4.4. Ensemble methods and stability

A discussion of model selection would not be complete without mentioning that ensemble and Bayesian methods, which pool a large number of solutions and "vote" among them, circumvent the problem of choosing a single one and often yield better results. A typical "frequentist" approach is to use "bagging" to obtain feature subset diversity by resampling the training data set and/or the feature set before feature selection. Such is the underlying methodology of Random Forests (RF) [4]. A more popular approach among Bayesians is to use Markov Chain Monte Carlo methods, both searching for useful feature subsets and attributing them a "weight" (the posterior probability of the feature subset) [26]. Generally, two approaches can be taken:

1. Using a committee of predictors, each based on a different feature subset (the voting weights are based on the expected individual prediction performance of the committee members, as determined *e.g.* by cross-validation).
2. Pooling the feature sets, retaining only the features, which are most often selected among the best feature sets encountered. A single predictor is then built from the pooled set.

The performance improvement obtained by ensemble methods is generally attributed to a reduction in "variance" or "instability". Formal relations between generalization performance and stability have been established [3]. Since multivariate methods often yield instable results, a lot of effort has been concentrating on stabilizing multivariate methods. While both approached (1) and (2) mentioned above are applicable to univariate methods, approach (2) does not guarantee that the feature set obtained will be optimal for multivariate methods since there is not necessarily a complementarity of features in a pooled set. If one seeks a stable complementary subset of features, one can resort to computing a feature subset "centroid", *e.g.* the feature subset closest on average to all selected feature subsets, according to some edit distance.

5. Conclusion

Feature selection is a multi-faceted problem, which has evolved over the past few years from a collection of mostly heuristic methods to a theoretically grounded methodology. In this chapter, we took a few snapshots of the kaleidoscope of existing methods to guide practitioners towards the best solution to their particular problem. Our final recommendation is to always start with the simplest methods (*e.g.* univariate feature ranking with correlation coefficients) and add complexity only as needed. Many promising avenues remain under-explored, such as "local" or "causal" feature selection, or even "dynamic" feature selection, in which features are selected on-the-fly, like in the game of 20 questions [3].

[3] In the game of 20 questions, you can ask 20 questions, which can be answered by "yes" or "no" to guess the identity of an object. Each question can be thought of as a feature. If the feature set is restricted and the

References

[1] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *JMLR*, 3:1229–1243, 2003.

[2] M. Boullé. Report on preliminary experiments with data grid models in the agnostic learning vs. prior knowledge challenge. In *IEEE/INNS conference IJCNN 2007*, Orlando, Florida, August 12-17 2007.

[3] O. Bousquet and A. Elisseeff. Stability and generalization. *JMLR*, 2, 2002.

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT press, 2002.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, USA, 2001.

[7] I. Guyon et al. Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark. *Pattern Recogn. Lett.*, 28(12):1438–1444, 2007.

[8] T. R. Golub et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[9] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory*, pages 63–77. Springer, 2005.

[10] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the NIPS 2003 feature selection challenge. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT Press, Cambridge, MA, 2005.

[11] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, Editors. *Feature Extraction, Foundations and Applications*. Studies in Fuzziness and Soft Computing. Physica-Verlag, Springer, 2006.

[12] I. Guyon and A. Saffari. The challenge learning object package (CLOP): Matlab(r) machine learning and feature selection methods, 2007, `http://clopinet.com/CLOP/`.

[13] I. Guyon, A. Saffari, G. Dror, and J. Buhmann. Performance prediction challenge. In *IEEE/INNS conference IJCNN 2006*, Vancouver, Canada, July 16-21 2006.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[15] T. Hastie, R. Tibshirani, A. Eisen, R. Levy, L. Staudt, D. Chan, and P. Brown. Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1:1–21, 2000.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer series in statistics. Springer, New York, 2001.

[17] K. Kira and L. Rendell. A practical approach to feature selection. In D. Sleeman and P. Edwards, editors, *International Conference on Machine Learning*, pages 249–256. Morgan Kaufmann, July 1992.

[18] R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2):273–324, December 1997.

[19] H. Liu and H. Motoda, Editors. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007.

[20] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Fauffman, 1988.

[21] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[22] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar. Ranking a random feature for variable and feature selection. *JMLR*, 3:1399–1414, 2003.

[23] R. Tibshirani. Regression selection and shrinkage via the lasso. Technical report, Stanford University, Palo Alto, CA, June 1994.

[24] I. Tsamardinos, C.F. Aliferis, and A. Statnikov. Algorithms for large scale Markov blanket discovery. In *16th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, pages 376–380, St. Augustine, Florida, USA, May 12-14 2003. AAAI Press.

[25] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, N.Y., 1998.

[26] A. Vehtari and J. Lampinen. Bayesian input variable selection using posterior probabilities and expected utilities. Report B31, 2002.

[27] J. Weston, A. Elisseff, B. Schoelkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *JMLR*, 3:1439–1461, 2003.

goal of the game is replaced by guessing an attribute, which cannot be queried, the game of 20 questions is a dynamic feature selection problem

Acknowledgements

We are grateful to Gavin Brown for making corrections and suggestions.

A. Forward selection with Gram-Schmidt orthogonalization

```
function idx = gram_schmidt(X, Y, featnum)
%idx = gram_schmidt(X, Y, featnum)
% Feature selection by Gram Schmidt orthogonalization.
% X         -- Data matrix (m, n), m patterns, n features.
% Y         -- Target vector (m,1).
% featnum   -- Number of features selected.
% idx       -- Ordered indices of the features (best first.)

[m, N]=size(X);
if nargin<3 | isempty(featnum), featnum=min(m,N); end
idx=zeros(1,featnum); w=zeros(1,featnum);
rss=zeros(1,featnum);       % Residual sum of squares
colid=1:N;                  % Original feature numbering
n=N;
for k=1:featnum % Main loop over features
    fprintf(’\nTraining on feature set size: %d\n’, N-n+1);
    % Normalize
    %(subtract the mean to get a correlation rather that a cos)
    XN=sqrt(sum(X.ˆ2));     % Norms of the feature vectors
    XN(XN==0)=eps;
    X_norma = X./repmat(XN, m,1); % Normalized feature matrix
    % Project onto Y
    y_proj = sum(repmat(Y, 1, n).*X_norma);
    ay_proj=abs(y_proj);
    % Find the direction of maximum projection
    [maxval, maxidx] = max(ay_proj); % Dir. of max. proj.
    idx(k)=colid(maxidx);            % Index of that feature
    % Update the model
    w(k)=y_proj(maxidx)/XN(maxidx);  % Weight of that feature
    Y_proj = w(k)*X(:,maxidx); % Proj. Y on dir. X(:,maxidx)
    Y_residual = Y - Y_proj;         % New residual
    rss(k) = sum(Y_residual.ˆ2);     % Residual error model
    % Compute the residual X vectors
    X_proj = sum(repmat(X_norma(:,maxidx),1,n).*X);
    X_residual = X-repmat(X_proj, m, 1).* ...
                            repmat(X_norma(:,maxidx), 1, n);
    % Change the matrix to iterate
    Y=Y_residual;
    X=X_residual(:, [1:maxidx-1,maxidx+1:n]);
    colid=colid([1:maxidx-1,maxidx+1:n]);
    n=n-1;
    fprintf(’Training mse: %5.2f\n’, rss(k)/m);
    fprintf(’Features selected:\nidx=[’);
    fprintf(’%d ’,idx(1:k));
    fprintf(’]\n’);
end
```

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-44

Industrial Mining of Massive Data Sets

Françoise FOGELMAN-SOULIÉ [a,1] and Erik MARCADÉ [a]

[a] *KXEN, 25 quai Galliéni, 92 158 SURESNES Cedex, France*

Abstract. Today data mining is more and more extensively used by very competitive enterprises. This development, brought by the increasing availability of massive datasets, is only possible if solutions to challenges, both theoretic and operational, are found: identify algorithms which can be used to produce models when datasets have thousands of variables and millions of observations; learn how to run and control the correct execution of hundreds of models; automate the data mining process. We will present these constraints in industrial contexts; we will show how to exploit theoretical results (coming from Vladimir Vapnik's work) to produce robust models; we will give a few examples of real-life applications. We will thus demonstrate that it is indeed possible to industrialize data mining so as to turn it into an easy-to-use component whenever data is available.

Keywords. Data Mining, Robustness, Scaling, Statistical Learning Theory, Structural Risk Minimization.

Introduction

Data mining is a scientific domain which has progressively emerged from the convergence of various fields: statistics (1900-1990); cybernetics (1940-1970); machine learning, artificial intelligence, pattern recognition, decision trees and neural networks (1970-1990), statistical learning theory (1980-2000). In the past 10 years, data mining has finally developed into a strong domain thanks also to computer science which has provided the technical means to handle massive datasets (fast computers, large cheap memory and hard disks, fast data bases, networks, . . .). However, such volumes impose specific constraints for data mining techniques to be efficient. We will show in this chapter how we have used Vladimir Vapnik's theoretical work (Statistical Learning Theory, Structural Risk Minimization) to handle these constraints and implement an industrial data mining tool: KXEN Analytic Framework.

Today, there exist many techniques in data mining [1] allowing to process large size problems: however, in practice, companies very often are still unable to exploit all the data they have and produce all the models they need [2]. This is in large part due to the poor productivity of existing tools, which, being usable by experts only, so far impose a "craftsman" development mode for data mining applications (Figure 1).

[1] Corresponding Author: Francoise@kxen.com

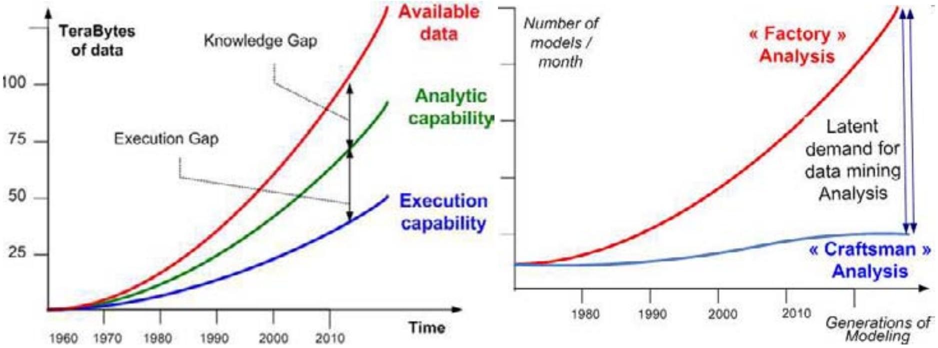


Figure 1. Gartner Analysis (from [2] left and [6] right)

1. The industrial context

Data sources nowadays are limitless and volumes grow exponentially: for example, at Yahoo! around 16 B transactions are produced every day, which translate into 10 Tbytes [3]; a large retailer (3000 stores) generate 300 M events from RFID tags per day [4]; the social network of Microsoft Instant Messenger has 240 M nodes [5]; mobile phone networks generate hundreds of million customer Call Detailed Records (CDR) per day and, for a large American metropolis, over 40M events per day over the technical mobile network. So data are plentiful. But, collecting and integrating data into a database still remains a very expensive task. When that task has been completed though, companies want to take advantage of it: needs for analysis thus grow, not as fast as the available data yet (Figure 1). Missing to fully exploit all available data produces gaps in knowledge and capacity to take actions, thus preventing companies to get the full Return On their Investment in producing the database. This failure is largely linked to the existing data mining tools: when being used by experts, they only allow to produce models in a craftsman fashion, when industrial production is required (Figure 1).

1.1. Which benefits do analyses bring ?

Data mining analysis allows companies to answer many questions and thus to become more efficient. For example, when launching a marketing campaign, a model is built to define a target optimized to yield best returns for smallest size (see an example in Section 4). Building more focused models can indeed increase the overall precision but the number of models is much larger. For example, a telecom company might want to build different scores for 20 geographic regions, 10 customer segments, 20 products and 10 score types (acquisition, cross-selling and up-selling, retention, fraud, ...): this results in 20 x 10 x 20 x 10 = 40,000 detailed models, which, by working on a more homogeneous population, produce smaller targets with better returns.

Today, most companies have a "gap" in their capacity to produce models and take action based upon them (Figure 1). Yet, the ability to produce data mining analyses in an industrial fashion is a key competitive factor: T. Davenport [7] shows how market leaders are "analytical competitors". He mentions for example Netflix, a leader in DVD rental, which produces 1 Billion predictions per day! Of course, such ability heavily relies on the data process the company has managed to put in place.

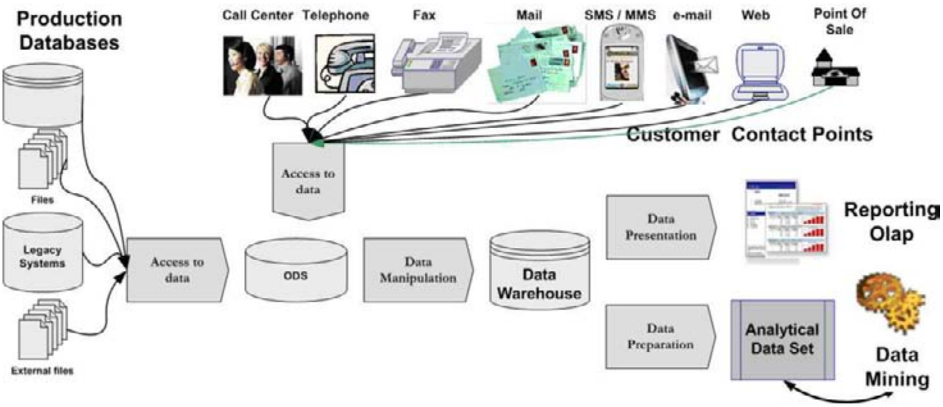


Figure 2. Data collection

1.2. Data

1.2.1. Data collection

The data collection process main complexity comes from the sheer number and heterogeneity of data sources: one needs to identify all the existing sources, to put in place collection mechanisms, to define a metadata dictionary and rules to align, manipulate and transform the data so as to finally produce the *Analytical Dataset* (ADS) from which models will be built. Figure 2 illustrates this process in the case of customer knowledge analysis. Of course, one can assemble an ADS without building a datawarehouse, but ultimately, if the number of models grows, one will need to industrialize the analysis process and it will be necessary to have a database in place.

1.2.2. Data Preparation

Most of the time, the datawarehouse is a multi-domain database, containing historical data from across the enterprise. In order to implement data mining analyses, one must choose in the datawarehouse the variables to be included into the ADS. Very often, transformations meaningful for the particular business or intended analysis will be made, producing new variables, such as aggregates, differences, ratios, transitions, … Then one will assess the data quality and finally, depending upon the algorithms and tools used, variables will need to be recoded (continuous or nominal variables…). This stage usually takes up more than half the time needed to implement a data mining analysis [8]

1.2.3. Data Quality

Data quality (distribution, missing data, outliers, correlations …) should be checked and maintained as high as possible: data must be exact, non redundant, complete … However, in practice, data quality is never perfect: data may be poorly entered, duplicates created, … a good data mining technique must thus be *robust* with respect to these quality problems.

In particular, processing missing values can be done in various ways: either by eliminating all observations with missing values (of course, one might need to eliminate many observations, especially when there are many variables !); or by replacing missing values through *imputation* of estimated values (most frequent category, average), which poses problems when values are not MAR (*missing at random*). Another solution used by KXEN is to create a special value *KxMissing* and handle it as any other value of the variable (cf Section 3.2 below).

1.2.4. Data Type

Today, most analyses use *structured* data coming from databases, flat files, spread-sheets ... However, more and more, *non structured* sources are becoming available: text (Web pages, SMS, emails, RSS flux, ...), multimedia (video, MMS, music). Non structured data volumes already account in average for more than 50% of all data in a company, yet most of this data are not kept into the datawarehouse [9] and thus not used in analyses. Data mining techniques need to be able to handle both structured and non structured data.

Because the ADS is usually built by extracting data from the database and storing it into a " flat " format, the *structure* which possibly existed between variables or observations is lost: variables are assumed non-correlated and observations i.i.d. However, such information could be very useful, as, for example, previous work on *social networks* has shown [10]: model performances can be enhanced by adding social networks-derived variables, taking into account the relationship between observations.

1.2.5. The analysis life cycle

Producing a data mining analysis to answer a business need today requires six weeks in average [8]. For many companies, this delay is too large especially when hundreds of models need to be produced. One must thus shorten that delay. This can be done in various ways: by reorganizing work processes (business users and IT in the same marketing department team for example). But the most effective way is to industrialize data access (giving users access to a business view on the datawarehouse), and simplify analysis processes (providing business users with user-friendly tools).

Speed is thus a critical performance factor. Reducing the time needed to produce a model (from design to production) allows to increase teams productivity (if a model can be built in 2 days instead of 6 weeks, teams can produce more models); increase performances (data used for modelling are more recent, market has not had time to change and deployed model performance is identical to expected performance); decrease time-to-market (reactivity to competitor's offer is faster). Very often, it is precisely this time-to-market reduction which companies value most.

1.2.6. Model factory

"Factory" analysis as defined by Gartner [6] and illustrated in 1 is the capacity to handle *massive datasets* (10-100 million customers, 5,000 variables) ... which requires an – almost – linear algorithm, minimal data manipulation (allowing only a few passes through the data with no duplication); the capacity to produce *very large number of models* (100-1000 projects per year, week, day ...) which requires the ability to automate model production, easy export and integration of model into the Information System; the capacity to *produce models very fast* (a few days, hours ...) which requires a user-friendly tool (for business users to use), allowing to automate heavy-duty tasks (data encoding, algorithms selection, model deployment, model execution and control …). The model factory can thus increase productivity (more models, produced faster by less people); increase benefits (it is possible to make models for any problem … even those for those it had never be done); increase speed (reduced time-to-market, data more recent). We now show how we have developed such a model factory engine, by using Vapnik's theory [11], [12].

2. Learning theory

2.1. Notations

Let us be given a data sample $D = \left(x^1, y^1\right), \left(x^2, y^2\right), ..., \left(x^n, y^n\right)$ where $x^i = \left(x_1^i, x_2^i, \ldots, x_p^i\right)$ is an *observation* in p variables and y^i its associated *label* – or *target*. y can be a discrete (classification) or continuous (regression) variable; we assume that in sample D, all labels y^i are known (no missing value). $\left(x^i, y^i\right)$ are supposed to be i.i.d. draws from a fixed but unknown distribution $P\left(X, Y\right)$.

Let $\Phi_\theta = \{f\left(., W, \theta\right), W \in \aleph\}$ be a class of functions from which we want to choose a model: for example, Φ_θ could be the class of polynomials of degree θ, or the class of MLP (multi-layer perceptrons) with θ hidden neurons ... A model from that class labels an observation x by the estimated label $y = f\left(x, W, \theta\right)$.

Statistical Learning Theory deals with the problem of finding the "best" model $\hat{y} = f\left(x, \hat{W}, \hat{\theta}\right)$ from the given dataset $D = \left(x^1, y^1\right), \left(x^2, y^2\right), ..., \left(x^n, y^n\right)$. For simplicity, we present the framework for regression (see [11], [13] for more details).

To measure the cost of replacing the true value y by its estimate $f\left(x, W, \theta\right)$ we define a *loss function* $L\left[y, f\left(x, W, \theta\right)\right]$. The *learning error* or *empirical risk* is then defined as the average loss on the learning dataset:

$$R_{emp}\left(W, \theta\right) = \frac{1}{n}\sum_{i=1}^{n} L\left[y^i, f\left(x^i, W, \theta\right)\right] \tag{1}$$

A common choice is the square error loss, for which empirical risk is just MSE (Mean Square Error):

$$L\left[y, f\left(x, W, \theta\right)\right] = \left[y - f\left(x, W, \theta\right)\right]^2$$
$$R_{emp}\left(W, \theta\right) = \frac{1}{n}\sum_{i=1}^{n} \left[y^i - f\left(x^i, W, \theta\right)\right]^2 \tag{2}$$

Generalization error is the expected loss, i.e. the error to be expected on a new observation:

$$R_{Gen}\left(W, \theta\right) = \int L\left[y, f\left(x, W, \theta\right)\right] \cdot dP\left(x, y\right) \tag{3}$$

The *Empirical Risk Minimization inference principle* or ERM requires to choose Wwhich minimizes empirical risk (4), which in case of MSE, is just Least Mean Square Error (LMSE) (5):

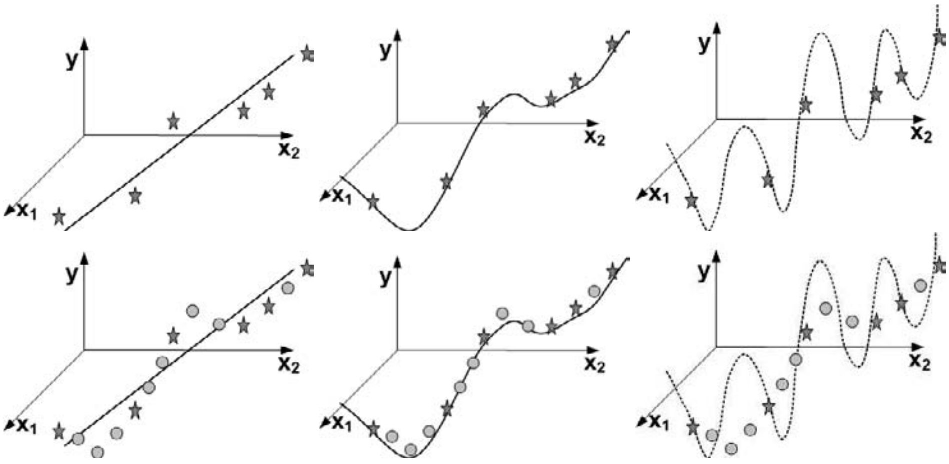


Figure 3. Accuracy of some models (top: the more complex model is more accurate) and Model robustness (bottom: the model with intermediate complexity is more accurate on new data shown as circles)

$$\hat{W}_\theta = \arg \underset{W}{min} R_{emp}(W, \theta) \tag{4}$$

$$\hat{W}_\theta = arg \ \underset{W}{min} \ \frac{1}{n} \sum_{i=1}^{n} \left[y^i - f\left(x^i, W, \theta\right) \right]^{\ 2} \tag{5}$$

The ERM principle allows to define the "best" $\hat{W}$ as that which gives the best accuracy on the learning sample, i.e. the best *data fit*, it does not say anything about how one effectively finds it (the learning algorithm) or how one chooses θ.

2.2. What do we expect from a model

From a given learning sample $D = \left(x^1, y^1\right), \left(x^2, y^2\right), ..., \left(x^n, y^n\right)$, the ERM inference principle consists in minimizing the empirical risk, i.e. in maximizing accuracy on the learning sample. There indeed exist many models in $\Phi_\theta$which minimize R_{emp}; the problem of finding model $f(x, W, \theta)$ from D is thus an ill-posed problem: Figure 3 shows various models built from D. In this case, ERM would lead us to pick the most complex model, which here has the best accuracy.

One can now look how model $f(x, W, \theta)$ behaves on new data (test set). Figure 3 (bottom) shows that if one wants a robust model, i.e. one with good accuracy on new data, then one should rather select the intermediate model, the most accurate one being very wrong on the new test set. Hence, ERM principle alone cannot ensure both accuracy and robustness.

2.2.1. Vapnik - Chervonenkis dimension

The *Vapnik - Chervonenkis dimension* – or VC dimension – measures the modelling capacity of the class of functions Φ_θ. We present this concept in the case of binary classification. Generalization to the general case of classification or regression is similar [11].

Let be given a sample of n observations $\left(x^1, x^2, ..., x^n\right)$ in p variables. There are 2^n ways to separate these observations into 2 classes. We will say that the family of

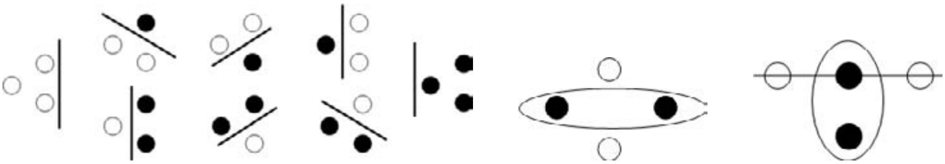


Figure 4. Samples of 3 and 4 points in $\Re^2$.

functions $\Phi_\theta = \{f(.,W,\theta), W \in \aleph\}$ *shatters* the sample if all 2^n separations can be achieved (through some proper choice of $\hat{W}_\theta$). We define the *VC dimension* h_θ of family Φ_θ as the maximum number of points which can be shattered by Φ_θ:

- There is at least one sample of h_θ observations which can be shattered by Φ_θ
- No sample of $h_\theta + 1$ observations can be shattered by Φ_θ

For example, the class of lines in $\Re^2$ has VC dimension 3 (Figure 4):

- There is at least one sample of 3 points which can be shattered by lines
- No sample of 4 points can be shattered by lines

Vapnik's Statistical Learning Theory [11], [12], [13] is based upon 4 principles:

1. Consistency (robustness): capacity to generalize well on new data;
2. Convergence speed: capacity to generalize increasingly well when the size of the learning sample increases;
3. Control of generalization capacity: strategy which allows to control the generalization capacity through control of the class VC-dimension;
4. Strategy to obtain good algorithms: strategy which allows to guarantee and measure the generalization capacity of the model produced.

2.2.2. Consistency

The ERM inference principle is said to be *consistent* for the class of functions $\Phi_\theta = \{f(.,W,\theta), W \in \aleph\}$ if $R_{emp}\left(\hat{W}_\theta,\theta\right)$ and $R_{Gen}\left(\hat{W}_\theta,\theta\right)$ converge to the same limit $\inf_W R_{Gen}(W,\theta)$ when the size n of the learning sample D increases, where $\hat{W}_\theta$ is the optimum value found by ERM (equation (4)). This happens if:

$$\forall \varepsilon > 0, \quad \lim_{n\to+\infty} P\left[\sup_W |R_{Gen}(W,\theta) - R_{emp}(W,\theta)| > \varepsilon\right] = 0 \tag{6}$$

Consistency means that, if n is large enough, then the error on new data will be the same as the learning error: the model is *robust* and can be trusted when applied to new data. Vapnik [11] has shown the following theorem:

***Theorem* 1** *Class Φ_θ is consistent if and only if Φ_θ has finite VC dimension h_θ.*

2.2.3. Convergence speed

V. Vapnik has proved [11] the following theorem:

***Theorem* 2** *for every $\eta \in [0,1]$, then, with probability* $1 - 4\eta$,

$$R_{Gen}\left(W,\theta\right) \leq R_{emp}\left(W,\theta\right) + \varepsilon\left(n,h\right) \qquad \text{with} \tag{7}$$
$$\varepsilon\left(n,h\right) = 2\sqrt{\frac{h\left[1+\ln\left(\frac{2n}{h}\right)\right]-\ln\eta}{n}}$$

This result is independent of distribution P: for n large enough, $\varepsilon \approx 0$, generalization and learning errors are the same, i.e. model is robust (Figure 5 top).

2.2.4. Generalization capacity control

In practice, we cannot make n as large as needed: n is just the size of the available dataset. From equation (7) above, there are two possible cases:

- If $\frac{n}{h}$ is large and if one minimizes empirical risk R_{emp} then R_{Gen} is of the same order (the difference ε is small);
- If $\frac{n}{h}$ is small, then one must minimize two terms: R_{emp} and $\varepsilon\left(n,h\right)$. Figure 5 (bottom) shows that, for n given, when h increases, then $\varepsilon \rightarrow +\infty$ and thus, from

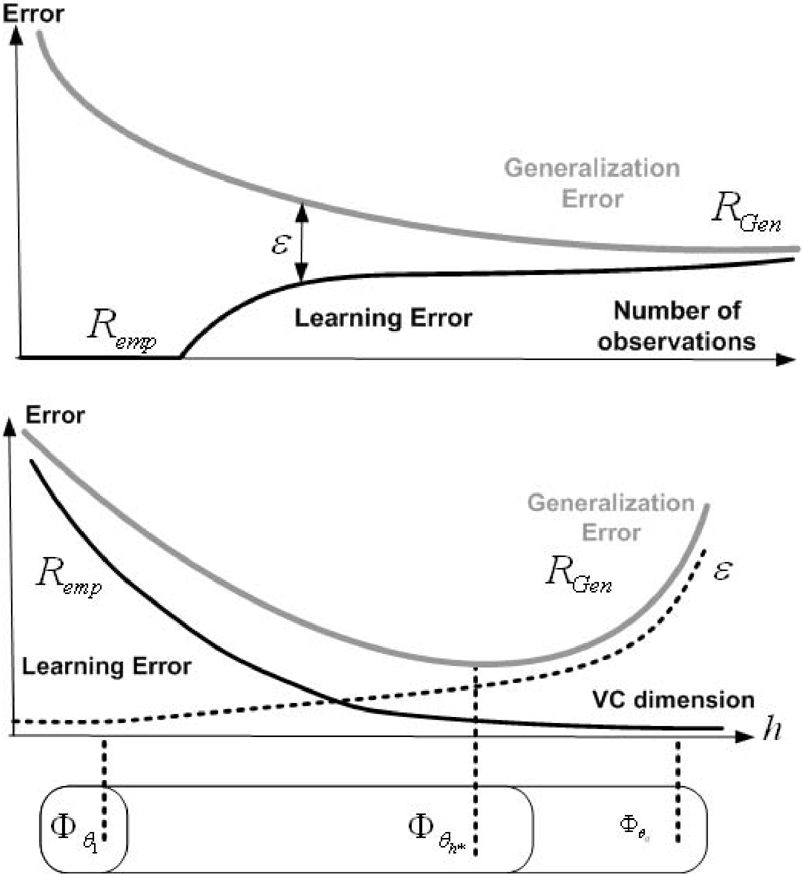


Figure 5. Consistency of ERM inference principle (top) and Control of generalization ability (bottom)

some VC dimension level h^* on, generalization error R_{Gen} starts increasing and becomes much larger than R_{emp} (this is the well known over-fitting effect).

Point h^* where R_{Gen} is minimum corresponds to the best compromise between accuracy (R_{emp} small) and robustness (R_{Gen} small).

2.2.5. Strategy to obtain good algorithms

We just saw that there exists an optimal value h^* which realizes the best accuracy–robustness compromise: we need a strategy to find that optimum. Vapnik [11] has introduced such a mechanism. In *Structural Risk Minimization* – SRM – he proposes to use a family of embedded classes of functions $\Phi_{\theta_1} \subset \Phi_{\theta_2} \subset ... \subset \Phi_{\theta_q} \subset ...$ with increasing VC dimension: $h_1 < h_2 < ... < h_q < ...$ (Figure 5).

The algorithm to obtain the best model then goes as follows: decompose the dataset Dinto two parts, the *estimation set* – or *learning set* – and the *validation set* (sometimes, a third part, or *test set*, is used to finally measure the performances of the model produced using the estimation and validation sets, but not for producing the model). We will use the validation error as an estimator of the generalization error:

1. Start with $k = 1$
2. Data fit: for Φ_{θ_k}, do:
   - On estimation set, produce "best" model in Φ_{θ_k} by ERM, i.e. choose

     $$\hat{W}_{\theta_k} = \arg\min_{W} R_{emp}\left(W, \theta_k\right)$$

     as in equation (4)
   - Measure error on validation set:

     $$R_{Val}\left(\hat{W}_{\theta_k}\right) = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} L\left[y^i, f\left(x^i, \hat{W}_{\theta_k}, \theta_k\right)\right] \tag{8}$$

   - If $R_{Val}\left(\hat{W}_{\theta_{k-1}}\right) \succ R_{Val}\left(\hat{W}_{\theta_k}\right)$ then $k = k + 1$ and GOTO 2;
   - Else stop and $k^* = k - 1$.
3. Model choice
   - Best model is obtained for θ_{k^*} and $\hat{W}_{\theta_{k^*}}$

SRM is a way to implement Occam's razor principle. More precisely, whereas Occam's razor uses model complexity as a measure of complexity, SRM uses the family of functions' complexity.

2.2.6. Conclusion

Vapnik's Statistical Learning Theory brings a set of results to control the VC dimension h of the class of models where we look for our solution. SRM ensures the best compromise between the model accuracy and its robustness. Since results are independent from data distribution, one does not need to estimate that distribution (which is a harder problem that finding the best model).

Let us remark that SRM does not say anything about the "best" class of models, except that its VC dimension needs to be finite.

SRM is used in many data mining techniques. Very early on, embedded families were built for MLPs (Multi-Layer Perceptrons) [14]:

- Through architecture: progressively increasing the number of hidden neurons;
- Through learning algorithm: if $\Phi_{\lambda_i} = \{f(x; W, \lambda_i),\ \|W\| \leq \lambda_i\}$ is the class of MLPs with bounded weights $W$and $\lambda_1 < \lambda_2 < ... < \lambda_h < \cdots$ The optimal solution in Φ_{λ_i}minimizes $\Re(W, \lambda_i)$, where C_i depends upon λ_i, the control parameter (this is called *weight decay*):

$$\Re(W, \lambda_i) = \frac{1}{n}\sum_{k=1}^{n}\left[y^k - f\left(x^k; W, \lambda_i\right)\right]^2 + C_i \sum_j W_j^2 \tag{9}$$

3. KXEN implementation

3.1. SRM in practice

KXEN uses Vapnik's Structural Risk Minimization. First the available dataset (the ADS) is decomposed into three sub-sets (Figure 6) for estimation (data fit), validation (model choice) and test (if one wants to measure the final model performances; this set is optional and is, of course, never used for finding the final model).

Let us define an embedded family of polynomials of some fixed degree d: $\Phi_{\theta_i}^d = \{f(x; W, \theta_i),\ \|W\| \leq \theta_i\}$, where f is a polynomial of degree d and $\theta_1 < \theta_2 < ... < \theta_h < \cdots$. For $\Phi_{\theta_i}^d$ the optimal vector is chosen using the observations in the estimation dataset through ERM (equation (4))

$$W_i^* = \arg\min_W \frac{1}{n}\sum_{k=1}^{n} L\left[y^k, f\left(x^k; W, \theta_i\right)\right] \tag{10}$$

under constraint$\|W\| \leq \theta_i$, which is equivalent, introducing the *Lagrange coefficient* C_i (or *ridge*) which depends upon θ_i, to minimizing the Lagrangian:

$$\Re(W, \theta_i) = \frac{1}{n}\sum_{k=1}^{n} L\left[y^k, f\left(x^k; W, \theta_i\right)\right] + C_i \sum_j W_j^2 \tag{11}$$

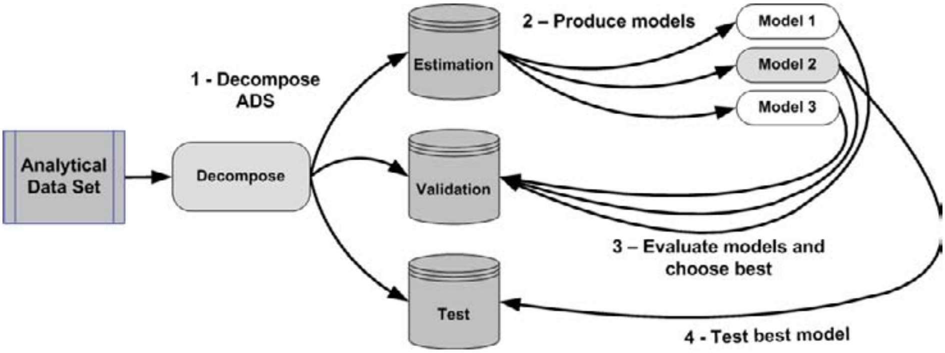


Figure 6. Implementing SRM in KXEN

This is obviously equivalent to a regularization method, where $\Re(W, \theta_i)$ is the regularized risk. KXEN can thus be considered as a polynomial ridge regression. Optimal parameter θ_{i*} is obtained through the method described in Section 2.2.5 using the validation dataset.

Finally, KXEN uses as loss function, neither the square error (2) nor the misclassification error, but KI– KXEN Information Indicator – which is the ratio between the area under the lift curve and the area under the perfect model. More precisely, KI is defined as follows for the classification case (the regression case can be easily generalized):

If $f(x; W, \theta)$ is a score model, then for threshold s, classifier C_s is defined as:

$$\begin{aligned} C_s(x) &= 1 \qquad \text{if } f(x, W, \theta) \geq s \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{12}$$

If we denote $G(s)$ the proportion of observations with score larger than s and $\alpha(s)$ and $\beta(s)$ the *sensitivity* and *specificity* of classifier C_s defined as:

$$G(s) = \int_s^{+\infty} f(x, W, \theta)\, dP(x) \quad \alpha(s) = \frac{TP}{nbTP} \quad \beta(s) = \frac{TN}{nbTN} \tag{13}$$

where TP, TN are the numbers of positive (true class 1) and negative (true class 0) observations correctly classified by C_s and $nbTP$, $nbTN$ the numbers of observations truly positive, negative; then the lift curve (Figure 7, top) represents $\alpha(s)$ as a function of $G(s)$ and AUC, the area under the ROC curve, is linked to KI by:

$$AUC = \int_{-\infty}^{+\infty} \alpha(s)\, d\left[1 - \beta(s)\right] \quad KI = 2AUC - 1 \tag{14}$$

KI is thus the *Gini index*. The difference between learning and validation errors then allows us to define KR– KXEN Robustness Indicator – by:

$$\varepsilon = KI_{Valid} - KI_{Estim} = 1 - KR \tag{15}$$

where KI_{Valid}, KI_{Estim} represent KI indices measured on the estimation and validation datasets respectively. If KR is close to 1, one can be confident that the model will be robust: it will generalize correctly to new data. On the other hand, if KR is low (KR varies between 0 and 1), the model can produce bad performances on new data. Figure 7 (bottom) shows, for example, the case of a sample where KR is small: lift curves for estimation, validation and test datasets are different and the model quality is poor (as degradation of performances on test set shows).

3.2. Data encoding

KXEN also uses SRM to automatically encode variables. To build a regression, classification or supervised segmentation, variables are encoded one by one as follows. For each variable $x_j, j = 1, ..., p$, we build an embedded family of encodings, and for each possible encoding, we produce a model explaining the label through that variable only, and we choose, with the usual SRM approach, the best encoding model. Each variable thus has its individual KI and KR which represent its ability to explain, alone, the target variable:

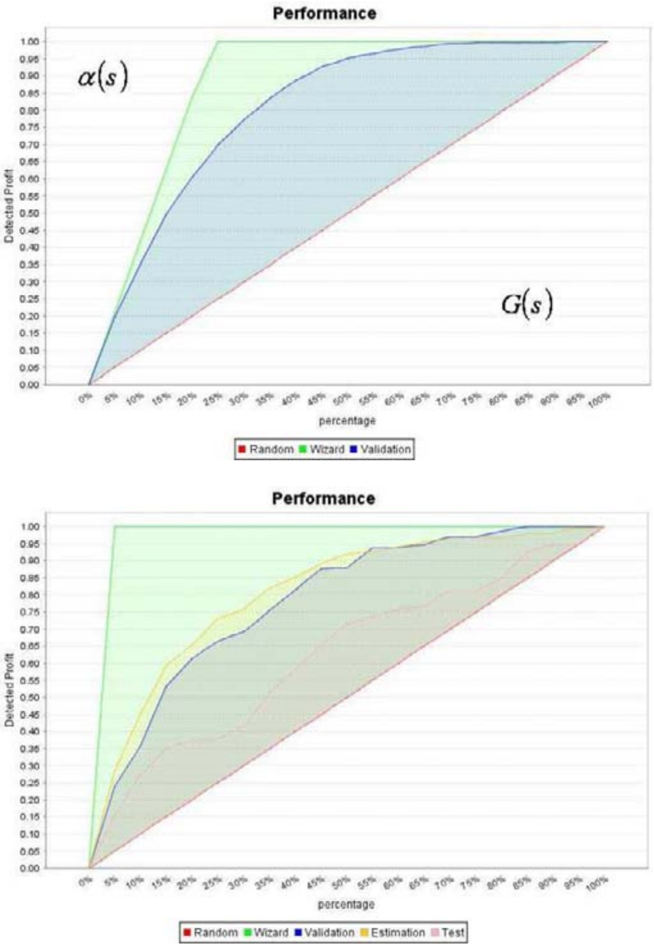


Figure 7. Lift curve

- If x_j is a continuous variable, we cut it into P bins (by default $P = 20$) and we progressively regroup intervals;
- If x_j is a nominal or ordinal variable, we progressively regroup its values.

The encoding produced is non-linear and depends upon the model (the target). For example, Figure 8 (top) shows the encoding of two continuous variables for a predictive model (“make more than \$ 50k per year”) for the well-known *Adult* dataset introduced in [15]: for variable *Age*, chances to make more than \$ 50k progressively increase until around 55 years and then decrease; for variable *Capital-Gain*, there is a threshold effect: capital gains need to be large enough to make more than \$ 50k. This encoding depends upon the target: if we now build a model to explain *Age*, with the same dataset, then variable *Capital-Gain* is coded as shown in Figure 8 (top right): capital gains increase monotonically with age. Categories of nominal variables are grouped depending upon the target: for example, Figure 8 (bottom) shows the encoding of two nominal variables for the same model as before (“make more than \$ 50k per year”). The 14 categories of

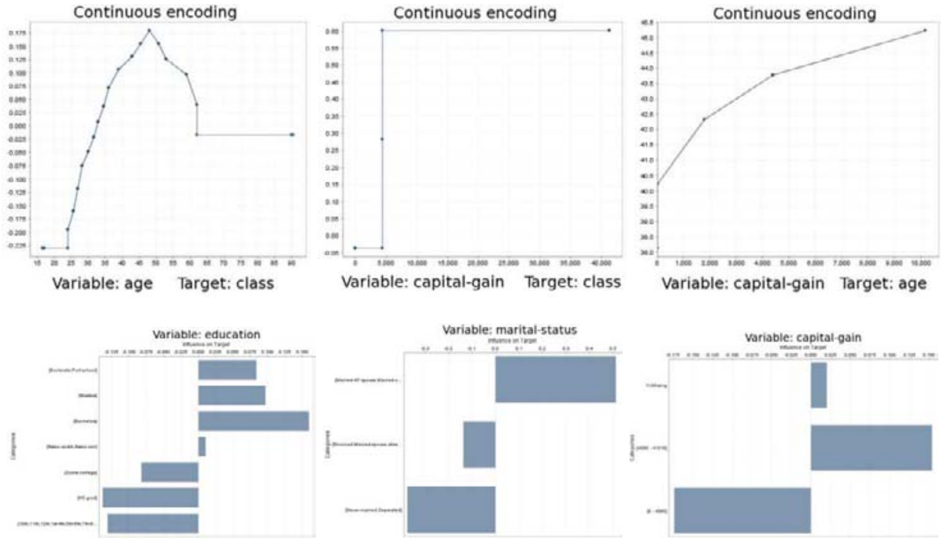


Figure 8. Coding of continuous variables *age* and *capital-gain* in *Adult* dataset (top) and Coding of nominal variables *education, marital-status* and *capital-gain* (bottom). Coding depends upon the target, i.e. the model built: codings of capital-gain in models to predict class (top-middle) and to predict age (top-right) are different.

variable *education* are grouped into 7 groups, and the 7 categories of variable*marital-status* in 3. Figure 8 (bottom right) also shows 2 categories of variable *capital-gain* with a threshold computed by the encoding model (5 178). One can also see that the model has created a category *KxMissing* to encode the missing values of that variable: this category has the same behaviour than large values of *capital-gain*, a sure sign that it is not Missing At Random – MAR – (one of the reasons why KXEN does not use imputation methods to handle missing data). Performance indicators are KI =0.808 and KR =0.992: the model is robust; its lift curve is shown in Figure 7 (top).

In general, non-linearities are well taken into account by the encoding, so that after this, a polynomial of degree 1 is sufficient; i.e. KXEN realizes a *linear ridge regression* in the non-linearly encoded variables space.

Exploitation of SRM thus allows to automatically encode variables (producing a robust encoding), to automate model production (the user does not need to choose among algorithms or tune hyper-parameters) and to obtain a robust model, with indicator KR measuring the confidence in model generalization performances. The class of proposed models is limited (for classification and regression) to polynomials: the user needs not compare various algorithms. This automatization allows to reduce the time needed to produce models by a factor of 10 on average, through massive reduction of the time spent in data exploration and encoding. KXEN techniques are almost linear, using data streams techniques to compute statistics and models (see paper by G. Hebrail on data streams in this book).

The software is based upon Vapnik's theory as we explained above in Section 3.1 and includes various modules: for data preparation (KEL, KSC et KTC), for automatic encoding (K2C), for building models (K2R, K2S, KTS et KAR) and to export models (KMX). As can be seen (Figure 9), KXEN does not provide a library of algorithms, but rather "analysis functions", as proposed by the JDM (Java Data Mining) standards [16].

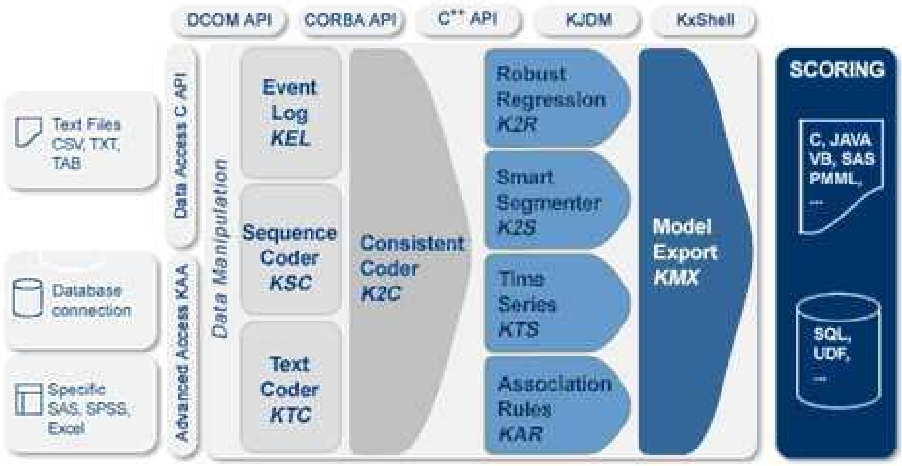


Figure 9. KXEN Analytic Framework

Number of Variables	
Sears	900
Large bank	1200
Vodafone D2	2500
Barclays	2500
Rogers Wireless	5800
HSBC	8000
Credit card	16000

Models per year	
Vodafone D2	760
Market research	9600
Cox Comm. D2	28800
Real estate	70000
Lower My Bills	460000

Figure 10. Many models and massive datasets

4. Some examples

Many customers use KXEN today on massive datasets, with many variables and build very large number of models (Figure 10). We briefly describe below a few applications using KXEN Analytic Framework.

LCL – Le Credit Lyonnais – is a very large French bank offering more than 400 bank, insurance and finance products and services. LCL launches each year around 130 direct marketing campaigns, through mailings, emailing, SMS, resulting in 10 M contacts. Targets for these campaigns are defined through about 10 general scores, established on groups of products. Marketing department wanted to be able to produce scores more accurate, refreshed more often and easily.

A project was launched on a live campaign to promote a home insurance. The customer database was decomposed in two: on the first part, the usual score was applied, on the other one, a specific score for that product was built with KXEN. Then the two scores were applied on 250 000 customers each to define the targets, the message was sent and returns were measured after some weeks: return from the KXEN score was 2.5 higher than with the other score (as should be expected from a specific score as opposed to a general-purpose score). In addition, the score was produced in a couple of days. Today,

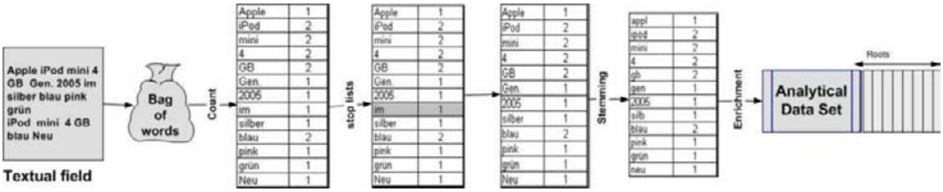


Figure 11. Text coding

more than 160 scores have been created – instead of the original 10 – and are routinely used in all campaigns.

Sears, as all large retailers, is facing a very strong competition and needs to continuously reduce costs and improve productivity. In particular, the catalog department has developed a strong modeling expertise to optimize its promotions. Because its IT system was becoming very hard to exploit, Sears decided to launch a project to modify its direct marketing system. First, they built a data mart integrating data from various brands (Sears, Orchard Supply Hardware, Lands' End) and various channels (stores, on-line, catalog), resulting in 900 attributes to describe each customer. KXEN was then deployed to automate the development of scores: the models are exported as SQL and UDF codes, running in the Teradata datawarehouse. Today, Sears create models in hours and score 75 million customers in 30 minutes [17].

Text mining: in 2006, the Data Mining Cup http://www.data-mining-cup.com attracted 580 participants. eBay Germany provided data from 8,000 auctions on its site. This data include two free-text fields *listing_title* and *listing_subtitle*, which the seller can use to describe his product. The task was to build a model to predict whether a product would sell higher than its category average selling price.

We used KXEN textual encoder module, KTC, which works as follows:

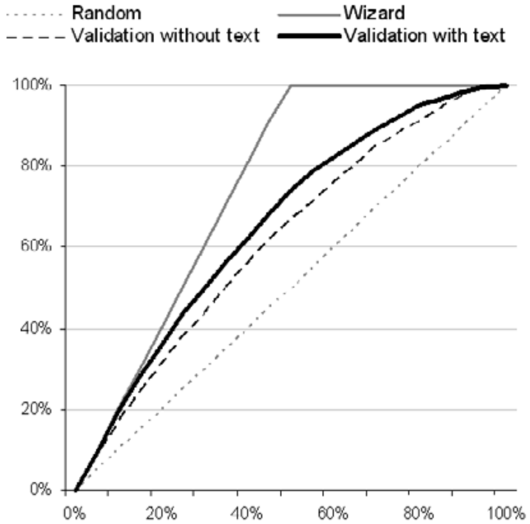
- Words are extracted and the usual indicators computed (counts, tf, tf-idf);
- Filters are applied (stop-lists, stemming) and word-roots extracted;
- These roots are added to the ADS (Figure 11).

We then used this ADS to produce the following models:

1. A simple regression K2R (after automatic encoding of all variables) without using text fields;
2. Same, but add variables, such as month of year, day of month, listing_end_monthofyear, listing_start_Monday...);
3. Use the KTC module for German to extract roots and then use K2R;
4. Build a specific language for this problem with specific words and synonyms (4 GB, 4GB, 4 gigas ...) and then use K2R;
5. Use a polynomial of order 2.

As can be seen in Figure 12, these different models perform better and better (the data mining cup winner got a score of 5020). The addition of roots (1000 in this case) extracted from the textual fields improves the results between models 2 and 3 (top right). At the same time, it increases the number of variables from 27 to 1027, which increases the time to produce the model (from 6s to 43 s), but the text variables are among the most significant (text variables significance is highlighted in dark in Figure 12, bottom). The algorithm we use thus has to be able to handle very large numbers of variables (retaining

Model	Score	Rank
K2R	2320	139
K2R + additional data	2852	123
KTC German	4232	68
KTC with specific language	4408	44
DMC (the winner)	5020	1
K2R order 2	5356	



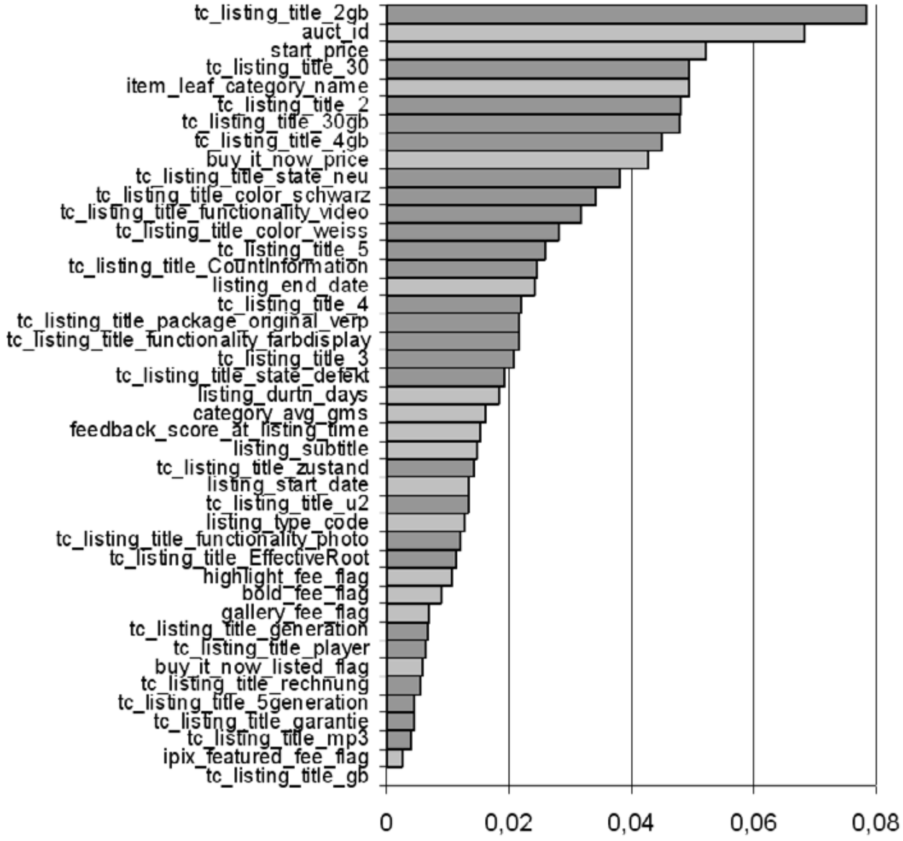


Figure 12. Results on Data Mining Cup 2006

2000 roots, instead of 1000, further increases slightly the model lift and computation time: 1mn 10 sec with 2027 variables).

5. Conclusion

We have shown how companies, who collect increasing volumes of data, are faced with the necessity to put in place *model factories*, if they want to industrialize their data mining analysis process. We then showed how Vladimir Vapnik's Statistical Learning Theory and, in particular, his Structural Risk Minimization, can be used to implement a tool, KXEN Analytical Framework, capable of producing robust models, on thousands of variables in very restricted time, thus answering the very challenging constraints of industrial companies. We do think that in the future, more and more companies will start exploiting their massive volumes of data – including text, social networks, ... – in a fully industrial, automated way.

References

[1] D. Hand, H. Mannila and P. Smyth, *Principles of Data Mining*. MIT Press. (2001)

[2] G. Herschel, CRM Analytics Scenario: The Emergence of Integrated Insight. *Gartner Customer Relationship Management Summit* (2006)

[3] U. Fayyad, A Data Miner's Story – Getting to Know the Grand Challenges, Invited Talk, *KDD'07*. (2007). http://videolectures.net/kdd07_fayyad_dms/

[4] H. Jiawei, Warehousing and Mining Massive RFID Data Sets, *adma'06* 2006. http://www.itee.uq.edu.au/~adma06/Jiawei_adma06_rfid.pdf

[5] J. Kleinberg, Challenges in Social Network Data: Processes, Privacy and Paradoxes, *KDD'07*, 2007. http://videolectures.net/kdd07_kleinberg_cisnd/

[6] G. Herschel, Customer Data Mining: Golden Nuggets, Not Silver Bullets. *Gartner Customer Relationship Management Summit* (2006)

[7] T. H. Davenport, J. G. Harris, *Competing on Analytics: The New Science of Winning*. Harvard Business School Press. (2007)

[8] W. W. Eckerson, Predictive Analytics. Extending the Value of Your Data Warehousing Investment. TDWI Best Practices Report. Q1 2007. https://www.tdwi.org/Publications/WhatWorks/display.aspx?id=8452

[9] P. Russom, BI Search and Text Analytics. *TDWI Best Practices Report.*, 2007. http://www.tdwi.org/Publications/WhatWorks/display.aspx?id=8449

[10] S. Hill, F. Provost, and C.Volinsky, Network-Based Marketing: Identifying Likely Adopters via Consumer Networks. *Statistical Science*, Vol. 21, No. 2, (2006) 256–276. http://pages.stern.nyu.edu/~fprovost/

[11] V.N. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[12] V.N. Vapnik, *Estimation of Dependencies Based on Empirical Data*. Reprint of 1982 Edition. Information Science and Statistics, Springer Verlag, 2006

[13] V.N. Vapnik, Problems of Induction, Foundation of Statistics and Empirical Inference. In Learning Theory and Practice. NATO Advanced Study Institute. July 8-19 2002 - Leuven Belgium. http://www.esat.kuleuven.ac.be/sista/natoasi/ltp2002.html

[14] I. Guyon, V. Vapnik, B. Boser, L. Bottou, S.A Solla, Structural Risk Minimization for Character Recognition, Advances in Neural Information processing systems 4. MIT Press, Cambridge, MA, (1992), 471-479.

[15] R. Kohavi, Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid, In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/

[16] M.F., Hornick, E. Marcade, S. Venkayala, *Java Data Mining. Strategy, Standard, and Practice. A practical guide for architecture, design, and implementation*. Morgan Kaufmann series in data management systems. Elsevier, 2007

[17] P. Bibler, and D. Bryan, Sears: A Lesson in Doing More With Less. *TM Tipline*. sept. 2005 http://ga1.org/tmgroup/notice-description.tcl?newsletter_id=1960075& r=#6

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-62

Large-Scale Semi-Supervised Learning

Jason WESTON [a]
[a] *NEC LABS America, Inc., 4 Independence Way, Princeton, NJ, USA 08540*

Abstract. Labeling data is expensive, whilst unlabeled data is often abundant and cheap to collect. Semi-supervised learning algorithms that use both types of data can perform significantly better than supervised algorithms that use labeled data alone. However, for such gains to be observed, the amount of unlabeled data trained on should be relatively large. Therefore, making semi-supervised algorithms scalable is paramount. In this work we review several recent techniques for semi-supervised learning, and methods for improving the scalability of these algorithms.

Keywords. semi-supervised, unlabeled data, transduction, large-scale, scalability

Introduction

One of the major ongoing challenges for the field of machine learning is dealing with the vast amount of data being generated in target applications. A lot of recent sophisticated machine learning algorithms appear to perform well on relatively small problems but don't scale very well (in terms of computational time) to a large amount of training data. This is ironic because with increasing dataset sizes, machine learning algorithms are guaranteed improved performance, but this performance is never attained simply because of the computational burden of calculating the result.

Another issue is that even though there is a large amount of data available, supervised machine learning algorithms need more than this: they need data that is labeled with a supervised target. Classification, regression and structured learning algorithms often give excellent performance compared to non-machine learning alternatives such as hand-built systems and rules, as long as the labeled data provided is of sufficient quality. However, although unlabeled data, such as data on the web, is abundant, labeled data is not – it is in fact often quite expensive to obtain, both in terms of monetary cost and labeling time.

Semi-supervised learning [2] is a framework in machine learning that provides a comparatively cheap alternative to labeling a huge amount of data. The aim is to utilize both the small amount of available labeled data *and* the abundant unlabeled data together in order to give the maximum generalization ability on a given supervised task. Using unlabeled data together with labeled data often gives better results than using the labeled data alone.

In this article we will discuss methods for performing semi-supervised learning which aim to scale up to large datasets to really achieve these goals. We will start with a summary of some of the classical approaches to this problem, such as transductive support vector machines (TSVMs) [16], label propagation type algorithms [20], and cluster-assumption encoding distance metrics [3,4]. We will then outline several recent techniques for improving the scalability of these algorithms, focusing in particular on fast to compute distance metrics [17] for kernel methods and a fast optimization algorithm for TSVMs [5].

1. Semi-Supervised Learning

1.1. Supervised, Unsupervised and Semi-Supervised Learning

We will study the standard setting for machine learning algorithms where one is given a (possibly labeled) training set of m examples. If one is given data that is unlabeled:

$$(x_1^*), \dots, (x_U^*), \tag{1}$$

then we are studying an *unsupervised* learning problem. Typical algorithms for making use of such data include estimating the density $p(x)$, clustering, outlier detection and dimensionality reduction – each paradigm having different aims with respect to what they want to discover from the data. The examples x_i could be any type of objects but are typically vectors in a d dimensional space, allowing such algorithms to make use of many mathematical techniques.

Sometimes, we have more information about the data. If a teaching signal is somehow provided that has *labeled* the data we now have data that are L pairs of examples:

$$(x_1, y_1), \dots, (x_L, y_L). \tag{2}$$

This is called the *supervised learning problem* and includes classification, regression and structured output learning, amongst others, depending on the kinds of labels provided.

We note that in supervised learning, there are two main families of methods: generative models and discriminative models. In a generative model one is interested in estimating the joint probability distribution:

$$p(x, y) = p(x|y)p(y) \tag{3}$$

so that one can make predictions

$$p(y|x) = \frac{p(x|y)p(y)}{\int_y p(x|y)p(y)dy}. \tag{4}$$

In a discriminative model one only estimates the probability of label assignment given an example:

$$p(y|x) \tag{5}$$

or (in two-class classification) whether

$$p(y|x) > 0.5, \tag{6}$$

i.e. one is only interested in which class an example belongs to. In this article we will mostly focus on classification tasks using discriminative models.

However, one problem is that often a teaching signal is expensive or difficult to obtain. This is because human labeling of data can cost both time and money. The labeling may require an expert depending on the problem and the difficulty of the labeling task. For example in bioinformatics labels can be difficult to obtain, e.g. the function of a protein can only be obtained through successful wet-lab experiments. In such cases, to supplement the (small amount of) labeled data (2) one may have access cheaply to a large unlabeled set of data (1). This setting is called the *semi-supervised learning* setting. The hypothesis is that using both sets of data together can help improve generalization on the learning task of interest.

1.2. Semi-Supervised Learning Paradigms

Naively, given the definition above, one could see semi-supervised learning as a way of mixing unsupervised learning and supervised learning together. So for example one could perform density estimation or clustering on the unlabeled data and classification on the labeled data, somehow combining these predictions into a shared model. Indeed, such combinations are possible, and several variants already exist. In the following paragraphs we describe three particular paradigms for semi-supervised learning that fall under this description.

Supervised setting The typical setting for measuring the success of semi-supervised learning is to treat it as a supervised learning problem where one has additional unlabeled data that could potentially be of benefit. One first trains a model to predict labels given the labeled training set and unlabeled data, and then measures the error rate on a separate labeled test set. However, other interpretations of semi-supervised learning are possible.

Unsupervised setting One can also be interested in the task of unsupervised learning where one has additional labeled data that could potentially be of benefit. So for example, one can learn a clustering of the data, where one is given some extra must-link or must-not-link constraints that are implicit in the labeled data. This paradigm has the advantage of being able to handle missing classes, which could be important in some problems, e.g. speaker identification or protein family detection, to name two.

Level-of-detail setting Finally, yet another way of looking at semi-supervised learning is to see training labels y_i for examples x_i as having various levels of detail (granularity). For example in text processing, at a very coarse level one could label whether example sentences are grammatically correct or not. However, a labeling with a finer level of detail would also label the parts-of-speech of the words in the sentence. An even finer level of detail could specify the syntactic parse tree of the sentence. Seen this way, semi-supervised learning should handle labels y_i that go between two extremes: from no label at all to very detailed label information. Each example can be labeled with a different detail-level and the learning algorithm has to be designed to deal with this fact.

1.3. The History of Semi-Supervised Learning

Semi-supervised learning probably started with the concept of "self-learning" in the 1960s and 1970s [11]. In this paradigm one makes predictions on unlabeled data, and then uses these predicted labels to train the model as if one had an increased set of labeled data. Vapnik also introduced the concept of transduction in the 1970s [16]. The transductive support vector machine (TSVM) [16] algorithm can be seen algorithmically as a kind of refinement of the self-learning approach: one also looks at the predictions of ones model on an unlabeled set of data, but this time one tries to choose the labeling which gives the most confident classification of those data, assuming the classifier outputs a measure of confidence, in this case using the notion of margin. Generative model approaches using the EM algorithm were also developed in the 1970s [8]. In [1] the co-training algorithm was introduced that is a clever use of unlabeled data that has multiple "views" or redundant sets of features to perform a kind of self-learning with a high degree of confidence. In the 1990s and 2000s several semi-supervised methods have been introduced that fit more into a regularization-based view – the use of unlabeled data is loosely seen as a regularization term in the objective function of a supervised learning algorithm. Amongst these are three notable algorithm types: TSVM-like algorithms, propagation of labels in a graph-based approach [20] and change of representation methods [3]. We will discuss these methods further in Section 2.

One issue with many of these algorithms is that they do not scale that well. In this article we will describe some of the recent advances in methods for making these algorithms more efficient. Those methods are discussed in Section 3.

1.4. Why Semi-Supervised Learning?

Supervised data is expensive both in monetary cost and labeling time. Labeling sentences with parse trees or finding protein function or 3D structure, to give two examples of labeled data, require human expertise.

Unlabeled data, on the other hand, is cheap to collect in many domains: audio-based problems, vision problems, and text processing problems, to name a few. In fact, most sensory-based problems that humans are good at have an abundant supply of unlabeled data. However, there are also other kinds of data, not natural to a human's perception, which are also relatively easy to collect compared to labeled examples of that same data. So, returning to our bioinformatics example, knowing the function (label) of a protein is costly, but obtaining its unlabeled primary structure (sequence of amino acids) is cheap.

In our view, true AI that mimics humans would be able to learn from a relatively weak training signal. For example, in the field of natural language processing, linguists label sentences with parse trees, but humans learn from data which usually has significantly less detailed labels. This argument perhaps strengthens the importance of *semi-supervised learning* as a topic of study in the field of machine learning.

1.5. Why Large-Scale Semi-Supervised Learning?

Many of the semi-supervised algorithms developed so far do not scale as well as one would like. For example, standard TSVM algorithms scale like $(U+L)^3$ where $U+L$ is the total number of examples [4], or worse. Because of this many experiments published are on relatively small, simple datasets comprising of around 50 training examples and a few thousand unlabeled examples [4,5,14].

Unfortunately, it appears that the impact of one extra unlabeled example in terms of improvement in generalization is smaller than the impact of one extra labeled example. Hence, if one has a realistic number of labeled examples, say a few thousand, then for unlabeled data to make a significant impact, you are likely to need hundreds of thousands of examples.

Hence, one clearly needs semi-supervised learning algorithms that scale well.

1.6. How Does Unlabeled Data Help in a Supervised Task?

Unlabeled data somehow gives knowledge about the density $p(x)$ but tells you nothing about the conditional density one is interested in $p(y|x)$ unless some assumptions are made in a training algorithm that hold true for a particular dataset.[1]

There are several possible assumptions that one can make about the data, each leading to different algorithms.

The cluster assumption One can assume that examples in the same cluster have the same class label. This implies that one should perform *low density separation* [4]; that is, the decision rule one constructs should lie in a region of low density. Many algorithms, such as TSVM [16], either explicitly or implicitly employ this assumption.

The manifold assumption One can also assume that examples in the same manifold have the same class. This is somewhat similar to the cluster assumption, but motivates different algorithms [20].

Zipf's law effect One obvious way unlabeled data can help in language problems is that one gets to see words that one has never seen before because a finite training set cannot cover the language. Zipf's law states that in a corpus of natural language utterances, the frequency of any word is roughly inversely proportional to its rank in the frequency table. In language problems, we are effectively always seeing new features (i.e. new words) every time we look at new examples. Our conjecture is that effective use of this knowledge in algorithms should surely improve performance over supervised learning alone.

Non-i.i.d. data Many machine algorithms and toy datasets assume that data are identically and independently distributed (i.i.d.) but in real life data this may not be true. We conjecture that if you see a test set that is drawn from a (slightly) different distribution to the training set then semi-supervised learning might help compared to purely supervised learning which is unaware of the distribution of the test set. For example, one might train a parser on the Wall Street Journal, but then apply it to the novel Moby Dick.

2. Semi-Supervised Learning Algorithms

In this section we will review some of the state-of-the-art methods for semi-supervised learning. While we cannot describe all available methods, we instead focus on three classes of method: (i) graph-based approaches, (ii) change of representation approaches and (iii) margin-based regularization based approaches.

All of these methods (either explicitly or implicitly) are special cases of a regularization approach to semi-supervised learning. In supervised learning a regularization approach leads one to optimize the empirical risk (training error) plus an additional term that limits the complexity of the decision rule:

$$\min_\alpha \sum_{i=1}^{L} \ell(f(x_i, \alpha), y_i) + \gamma\Omega(\alpha) \tag{7}$$

[1] Note that in supervised learning, a similar issue applies: most algorithms assume $p(y|x)$ and $p(y|x')$ are similar when x is close to x'.

where $\ell(\cdot)$ is a loss function encoding the penalty for incorrectly labeling a training example, and α encodes the parameters of the decision function.

In semi-supervised learning one approach is to do exactly the same as in supervised learning, but add one additional term that encodes the assumption one has made about the data w.r.t. using unlabeled examples, e.g. one may wish to encode the cluster assumption. This leads one to minimize:

$$\min_\alpha \sum_{i=1}^{L} \ell(f(x_i, \alpha), y_i) + \gamma\Omega(\alpha) + \lambda\Lambda(x^*, \alpha). \tag{8}$$

The regularizer $\Lambda(x^*, \alpha)$ is often point-wise, i.e.:

$$\Lambda(x^*, \alpha) = \sum_{i=1}^{U} \ell^*(f(x_i^*), \alpha) \tag{9}$$

where $\ell^*(\cdot)$ is a function that encodes the assumption of choice.

We now discuss some specific semi-supervised learning algorithms.

2.1. Graph-Based Approaches

Several semi-supervised algorithms work by constructing a graph on the unlabeled data, and then regularize using this graph in some way.

The authors of [20] (see also [12,19]) studied the so-called label-propagation type algorithm. In this algorithm one first defines F as a $(U + L)$ x 2 matrix for two-class classification problems, where

$$f(\bar{x}_i) = \text{argmax}_j F_{ij} \tag{10}$$

is the class prediction for a point $\bar{x}_i$, and $\bar{x}$ is the set of both labeled and unlabeled examples (ordered so that the labeled examples are first). A $(U+L)$ x 2 label matrix Y is also defined, where $Y_{ij} = 1$ if example $\bar{x}_i$ is classified into class j and $Y_{ij} = 0$ otherwise (so $Y_{ij} = 0$ for unlabeled examples $i > L$).

One then solves the following quadratic program:

$$\min_F \sum_{i=1}^{L+U} ||F_i - Y_i||^2 + \lambda \sum_{i,j=1}^{L+U} W_{ij}||F_i - F_j||^2 \tag{11}$$

where W_{ij} are weighted edges on the graph that one constructs a priori on the given data. For example, one can define $W_{ij} = 1$ if example i is one of example j's k-nearest neighbors, although many other schemes are possible. The matrix W is usually normalized so that the sum of outgoing links from any node in the graph sum to 1.

Intuitively, the first term minimizes the training error, whilst the second term regularizes using the unlabeled data. If two unlabeled examples are close in input space, then this algorithm tries to give them the same label. Because one is enforcing this constraint on all unlabeled and labeled points at the same time, there is a "label-propagation" effect where neighbors-of-neighbors will also try to have the same label as well. This can be clearly seen when writing down the algorithm as an iterative algorithm [19]:

1. Iterate $F(t + 1) = \alpha WF(t) + (1 - \alpha)Y$ until convergence.

2. Label each point $\bar{x}_i$ as $\arg\max_j F_{ij}(t)$.

That is, on each iteration, the label of an example is influenced by the weighted combination of all the labels of its neighbors. One can also write the solution in closed form:

$$F = (I - \alpha W)^{-1}Y. \tag{12}$$

This algorithm is essentially a generalized version of k-nearest neighbor with extra semi-supervised regularization. This can be seen because the first iteration predicts the label of an unlabeled example using the example's neighbors. Subsequent iterations provide regularization. One issue then is that although the regularization term seems reasonable, k-nn is not necessarily the best base classifier one could choose. A further issue is that, if one solves the problem as a quadratic program or as a matrix inversion problem, the complexity is $(L + U)^3$. However, one could *approximate* the solution by only running a few iterations of the iterative version of the algorithm.

LapSVM A more recent use of this same regularizer is the Laplacian SVM [14]. In that work they add the same regularizer to the SVM [16] objective function:

$$\min_{w,b} \sum_{i=1}^{L} \ell(g(x_i), y_i) + \gamma ||w||^2 + \lambda \sum_{i,j=1}^{U} W_{ij} ||g(x_i^*) - g(x_j^*)||^2 \tag{13}$$

where $g(x) = w \cdot x + b$. Here, as in a standard SVM, one is solving a two-class classification problem $y_i \in \pm 1$ and the final predictions are given by $f(x) = \text{sign}(g(x))$. This algorithm gives excellent results, but still suffers from having to compute a matrix inversion in the specific optimization technique the authors propose.

2.2. Change of Representation-Based Approaches

Another approach to semi-supervised learning is to simplify the regularization of the optimization: instead of optimizing both the supervised learning part of the optimization with an additional regularizer *at the same time* one could decompose into a two-part strategy:

- Build a new representation of the data using unsupervised learning on the unlabeled data.
- Perform standard supervised learning using the labeled data and the new representation.

The aim here is to map the data $x_i \mapsto \phi(x_i)$ such that the distances in the new representation encode manifold or cluster assumptions about the data. One can then use a standard classifier such as SVM on this data.

One study of this technique is given in [3]. In that work, several representations are defined for SVM that try to encode the cluster assumption. One desires the relative distance in the new representation is relatively smaller between two points that are in the same cluster. SVM training does not necessarily require the actual feature vectors of training examples in order to learn, but only the inner products:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j). \tag{14}$$

These inner products are referred to as kernels. The re-representations proposed by the authors of [3] are referred to as cluster kernels.

Two proposed cluster kernels are:

1. The random walk kernel. One defines $W_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$ so that the probability of "walking" along the graph between points i and j is defined as $P(x_i \rightarrow x_j) = \frac{W_{ij}}{\sum_p W_{ip}}$. One can then compute t steps of a random walk using $P \leftarrow P^t$ (see also [15]). The idea here is to define a kernel[2] based on P^t. This representation has a similar motivation to the label propagation algorithm we have seen before: if it is easy to "walk" between two examples, i.e. they are neighbors, or neighbors-of-neighbors, then they are likely to have a similar label.

2. The spectral clustering kernel. One performs spectral clustering [10] on the labeled and unlabeled data, and then builds a representation via the first k eigenvectors for training the SVM.

Both of these kernels have a complexity of $O((U + L)^3)$ to compute.

2.3. Margin-Based Approaches

Another way of encoding a cluster assumption type regularizer is via the margin (distance) of the examples from the decision rule of a classifier of the form

$$g(x) = w \cdot x + b. \tag{15}$$

The transductive SVM (TSVM) is an example of this approach. The idea is that the decision rule should lie in a region of low density (the cluster assumption) so an unlabeled example x tends not to have the value $g(x) = 0$. This is encoded with the following optimization problem:

$$\min_{w,b} \sum_{i=1}^{L} \ell(g(x_i), y_i) + \gamma||w||^2 + \lambda \sum_{i=1}^{U} \ell^*(g(x_i^*)). \tag{16}$$

The loss functions $\ell(\cdot)$ and $\ell^*(\cdot)$ are typically chosen to be the hinge loss function

$$\ell(g(x), y) = \max(0, 1 - yg(x)) \tag{17}$$

and the symmetric hinge loss function

$$\ell^*(|g(x)|) = \max(0, 1 - |g(x)|) \tag{18}$$

respectively. The idea is that labeled examples are "pushed" away from the hyperplane so that they are correctly classified and achieve a margin of 1. Unlabeled examples are also "pushed" away from the hyperplane in the same way. To make this work, typically a balancing constraint is added so that all the unlabeled examples do not get "pushed" to the same side:

[2] Technically, one has to symmetrize this matrix to make it a valid kernel as described in [3].

Table 1. A comparison of supervised and semi-supervised algorithms on two small datasets.

Algorithm	*USPS*	*Mac-Win*
k-NN	9.3%±0.32	29.1%±0.68
SVM	7.4%±0.31	28.1%±1.39
Label propagation	1.79%±0.07	30.2%±1.35
SVM + cluster kernel	5.9%±0.24	11.0%±0.45
TSVM	5.62%±0.2	11.1%±0.48

$$\frac{1}{U}\sum_{i=1}^{U} g(x_i^*) = \frac{1}{L}\sum_{i=1}^{L} y_i \, . \tag{19}$$

Unfortunately this optimization problem is non-convex because of the absolute value in the $\ell^*(\cdot)$ function. The implementation in the SVMLight software [9] works by solving successive SVM optimization problems – it improves the objective function with a heuristic method: it iteratively switches the labels of two unlabeled points x_i and x_j to improve the objective function. It is rather slow with anything more than a few thousand examples. Another implementation called ∇TSVM [4] proposed to optimize this function directly by gradient descent, with a complexity of $(L+U)^3$.

2.4. Experimental Comparison

An experimental comparison on some small datasets is given in Table 1. Two supervised algorithms (k-NN and SVM) are compared to one algorithm from each of the three semi-supervised approaches described above: label propagation, SVM + spectral clustering kernel and TSVM. USPS is an optical digit recognition dataset, we used 32 labeled examples and 4000 unlabeled examples in 256 dimensions. Mac-Win is a text classification dataset (with a bag-of-words representation) with 16 labeled examples and 2000 unlabeled examples in 7511 dimensions. The results are average over 30 splits of the data. For USPS, semi-supervised learning improves the error rate. For Mac-Win the problem may be too high dimensional for label propagation to work well, however it does perform well on USPS.

3. Large-Scale Semi-Supervised Learning Algorithms

Having reviewed several of the semi-supervised algorithms developed in the literature, we will now describe several recent advances in speeding up these algorithms.

3.1. Graph-Based Approaches

As already mentioned, if one performs *early stopping* of the iterative algorithm for label propagation, instead of a direct optimization of the given objective function, one can speed this algorithm up. Such an approximation might give a loss of accuracy in the final predictions.

However, several other speed-ups are possible. Firstly, the algorithm is much faster if the matrix W is very sparse. Secondly it was proposed in [6] to approximate the solution in a different way. The idea is that some unlabeled examples can be expressed as a linear combination of other examples, thus reducing the number of variables one has to optimize. This reduces the complexity of the algorithm from $O(k(U+L)^2)$, where each point has k neighbors, to $O(m^2(U+L))$ where $m << (U+L)$.

3.2. Change of Representation Approaches

For change of representation methods, the problem has been split into two tasks: an unsupervised learning algorithm and a supervised learning algorithm. Clearly then in order to perform large-scale semi-supervised learning, one should choose algorithms from both of these two areas that also scale well.

Therefore, one should first choose the fatest clustering algorithm one can. For example, k-means [7] is generally faster than spectral clustering, the algorithm we used before. It has a complexity of $O(rkUd)$, with a problem of dimensionality d running for r iterations. Empirically, running time grows sublinearly with k, n and d.

Thus, the following k-means cluster kernel was proposed in [17]:

1. Run k-means N times.
2. $K_{bag}(x_i, x_j) = \dfrac{\text{number of times } x_i \text{ \& } x_j \text{ in same cluster}}{\text{number of runs}}$
3. Take the product between the original and bagged kernel:
$$K(x,x') = K_{orig}(x,x') \cdot K_{bag}(x,x') \tag{20}$$

The idea is that the original kernel K_{orig} is rescaled by the “probability” that two points are in the same cluster as approximated by K_{bag}. This method was empirically evaluated in the context of protein superfamily classification, where one has to predict whether a protein belongs to a target superfamily or not. Experimental results are shown in Table 2, averaged over 54 target families. This method is compared on a small dataset with $L = 3000$ and $U = 3000$ so that other methods are feasible to compute, and then its error rate is computed with a much larger set of $U = 30000$ unlabeled examples. This method seems to have a good performance compared to other approaches, while having far better scaling properties.

3.3. Margin-Based Regularization Approaches

A well-founded technique for optimizing TSVM that also has improved scaling properties was recently developed in [5]. The idea is to make use of the concave-convex procedure (CCCP) [18], a technique of non-convex optimization. In CCCP, one rewrites the cost function of interest $J(\theta)$ as the sum of a convex part $J_{vex}(\theta)$ and a concave part $J_{cav}(\theta)$. One then iteratively solves

$$\theta^{t+1} = \text{argmin}_{\theta} \left\{ J_{vex}(\boldsymbol{\theta}) + J^{'}_{cav}(\boldsymbol{\theta}^t) \cdot \boldsymbol{\theta} \right\}. \tag{21}$$

Each iteration of the CCCP procedure approximates the concave part by its tangent and minimizes the resulting convex function. $J(\theta^t)$ is guaranteed to decrease at each iteration, and the whole procedure is guaranteed to converge to a local minima.

Table 2. Comparison of semi-supervised methods on a protein prediction problem taken from [17]. Error rates are measured using the Receiver Operator Characteristic (ROC) and the ROC up to the first 50 false positives (ROC_{50}).

	ROC_{50}	ROC
L=3000 U =3000		
SVM	0.416	0.875
SVM + spectral clustering kernel	0.581	0.861
SVM + random walk kernel	0.691	0.915
SVM + k-means kernel ($k = 100$)	0.719	0.943
SVM + k-means kernel ($k = 400$)	0.671	0.935
TSVM	0.637	0.874
U = 30000		
SVM + k-means kernel ($k = 100$)	0.803	0.953
SVM + k-means kernel ($k = 400$)	0.775	0.955

Table 3. A comparison of CCCP-TSVM with existing TSVM methods on a variety of small-scale datasets, taken from [5].

dataset	classes	dims	points	labeled
g50c	2	50	500	50
Coil20	20	1024	1440	40
Text	2	7511	1946	50
Uspst	10	256	2007	50

	Coil20	g50c	Text	Uspst
SVM	24.64	8.32	18.86	23.18
SVMLight-TSVM	26.26	6.87	7.44	26.46
∇TSVM	17.56	5.80	**5.71**	**17.61**
CCCP-TSVM	**17.28**	**4.88**	**5.71**	18.08

For the TSVM, such an approach is quite simple to implement and only requires solving a sequence of convex problems similar to the SVM algorithm. Hence it has quadratic empirical complexity in the number of examples, like SVMs have, and even on small problems of 2000 unlabeled examples is around 133 times faster than SVMLight, whilst having similar, or better, accuracy. An empirical comparison with existing approaches is shown in Tables 3, 4 and Figure 1. CCCP-TSVM sofware is available at `http://www.kyb.tuebingen.mpg.de/bs/people/fabee/universvm.html`.

Finally, another type of speed-up for TSVM was given in [13]. The authors proposed a "multi-switch" version of the SVMLight-based approach. The idea is to swap the labels of many examples at once, rather than the pair of examples approach suggested by [9]. This method was developed for a very fast linear SVM software called SVMLin, available at `http://people.cs.uchicago.edu/~vikass/svmlin.html`.

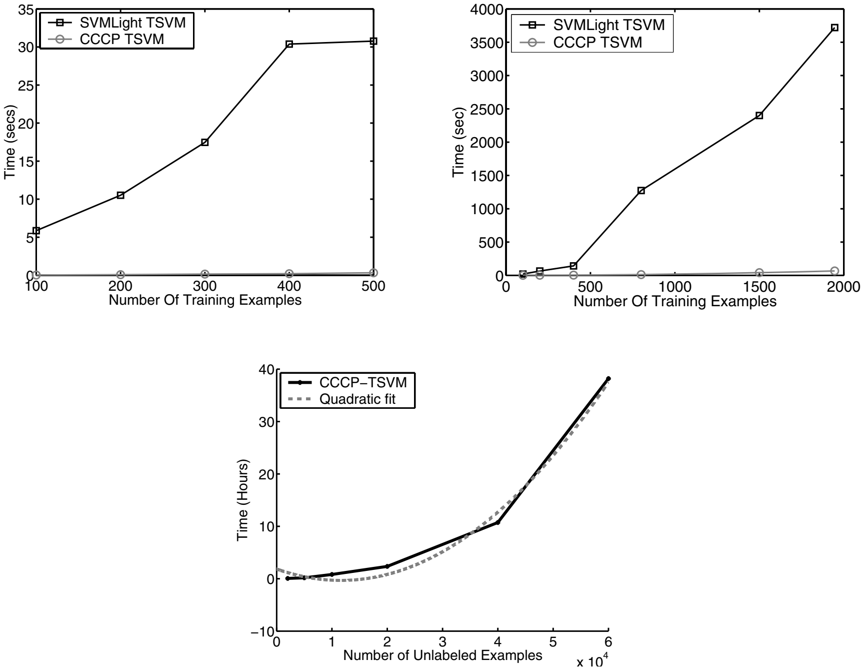


Figure 1. Training times for g50c (upper left), Text (upper right) and MNIST (lower) for different numbers of unlabeled examples, U. CCCP-TSVM are 133 times faster than SVMLight-TSVMs on the text dataset for $U = 2000$, and scale quadratically on larger datasets like MNIST.

Table 4. Large-scale results using CCCP-TSVMs [5] on the MNIST dataset, a 10-class digit recognition problem with $L = 1000$ labeled examples, and varying amount of unlabeled examples U.

Method	L	U	Test Error
SVM	1000	0	7.77%
TSVM	1000	2000	7.13%
TSVM	1000	5000	6.28%
TSVM	1000	10000	5.65%
TSVM	1000	20000	5.43%
TSVM	1000	40000	5.31%

4. Conclusion

In this article we have reviewed some of the main types of discriminative semi-supervised learning algorithms for classification that are available, and discussed some of the latest advances in making those algorithms scalable.

Semi-supervised learning is useful when labels are expensive, when unlabeled data is cheap and when $p(x)$ is useful for estimating $p(y|x)$, e.g. if either the manifold or cluster assumptions are true. We have reviewed several different algorithmic techniques for encoding such assumptions into learning, generally this is done by somehow "marrying" unsupervised learning into a supervised learning algorithm. Instances of this approach are graph-based approaches, change of representation based approaches and margin based approaches. All of these can somehow be seen as either explicitly or implicitly adding a regularizer that encourages that the chosen function reveals structure in the unlabeled data.

Large-scale learning is often realistic only in a semi-supervised setting because of the expense of labeling data. Moreover, the utility of an unlabeled example is less than a labeled one, thus requiring a relatively large collection of unlabeled data for its use to be effective. However, to make current algorithms truly large-scale, probably only linear complexity with respect to the number of examples will suffice. At least in the non-linear case, current approaches still fall short, leaving the field still open for further research.

References

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, pages 92–100, 1998.

[2] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., USA, 09 2006.

[3] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. volume 15, pages 585–592, Cambridge, MA, USA, 2003. MIT Press.

[4] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. pages 57–64, 01 2005.

[5] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, 08 2006.

[6] O. Delalleau, Y. Bengio, and N. Le Roux. Large-scale algorithms. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 333–341. MIT Press, 2006.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley & Sons, New York, 1973.

[8] D. Hosmer Jr. A Comparison of Iterative Maximum Likelihood Estimates of the Parameters of a Mixture of Two Normal Distributions Under Three Different Types of Sample. *Biometrics*, 29(4):761–770, 1973.

[9] T. Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning, ICML*, 1999.

[10] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, Cambridge, MA, 2002. MIT Press.

[11] H. Scudder III. Probability of error of some adaptive pattern-recognition machines. *Information Theory, IEEE Transactions on*, 11(3):363–371, 1965.

[12] J. Shrager, T. Hogg, and B. A. Huberman. Observation of phase transitions in spreading activation networks. *Science*, 236:1092–1094, 1987.

[13] V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear SVMs. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 477–484, New York, NY, USA, 2006. ACM Press.

[14] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *International Conference on Machine Learning, ICML*, 2005.

[15] M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. *Neural Information Processing Systems 14*, 2001.

[16] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

[17] J. Weston, C. Leslie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. volume 16, pages 595–602, Cambridge, MA, USA, 2004. MIT Press.

[18] A. L. Yuille and A. Rangarajan. The concave-convex procedure (CCCP). In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[19] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. volume 16, pages 321–328, Cambridge, MA, USA, 2004. MIT Press.

[20] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-76

User Modeling and Machine Learning: A Survey

Thierry ARTIÈRES

LIP6, Université Paris 6, France

Abstract. This paper presents a survey of UM tasks and of related machine learning and data mining problems. It first proposes a panorama of UM applications merged in a few categories according to the similarity of the tasks from the machine learning point of view. Then it details a little more details usage mining for hypermedia and web-based applications.

Keywords: User modeling, log processing, usage mining.

Introduction

The increasing complexity of user tasks and the availability of increasing amount of information to a wide variety of users have made user modeling (UM) an important component of many today applications and services. UM aims at modeling the user in order to help him efficiently use the systems he is offered and retrieve the information he is looking for through adaptation and personalization e.g. filtering the information according to his will and needs.

Information systems, hypermedia, websites, application software are becoming more and more complex, hence difficult to use efficiently. The increasing amount of on-line information available to a user through Internet makes recovering information harder and harder. While many software, hypermedia, websites and services are potentially used by a variety of users, these systems have traditionally been developed in a "one size fits all" manner. Consequently, they are often not adapted to most of the users, with various knowledge, preferences, and needs.

There are two main questions when building a user modeling component. The first question is how to represent a user, i.e. what is a user model? It should include all information (goal, plans, preferences, beliefs, etc.) useful for improving interaction between the user and the system. The second question is how to gather the information about a user. There are many situations where explicit collection of user data (e.g. through form filling) is not adapted or possible. Then one can use non intrusive machine learning techniques for automatic gathering of high level information, the user model, from low level data, i.e. interaction logs with the system.

This paper presents a survey of UM tasks and of related machine learning and data mining problems. This is not an easy task according to the wide scope of UM applications and to the variety of machine learning techniques commonly used for solving one particular UM task; Taxonomy of UM tasks and of ML techniques do not match well [1]. [2] proposed a synthetic survey but for a limited scope of UM applications. In this paper we first propose a brief panorama of UM applications that we classify in a few categories according to the similarity of the tasks from the machine learning point of view. Then we discuss of what are user models in this variety of applications. Finally we discuss in some more details usage log mining based applications.

1. Broad User Modeling Applications

We present briefly in this section the main UM tasks by grouping these in four families of applications that exhibit some similarity with respect to the ML tasks they rely on and to the ML techniques used, Figure 1 illustrates this taxonomy.

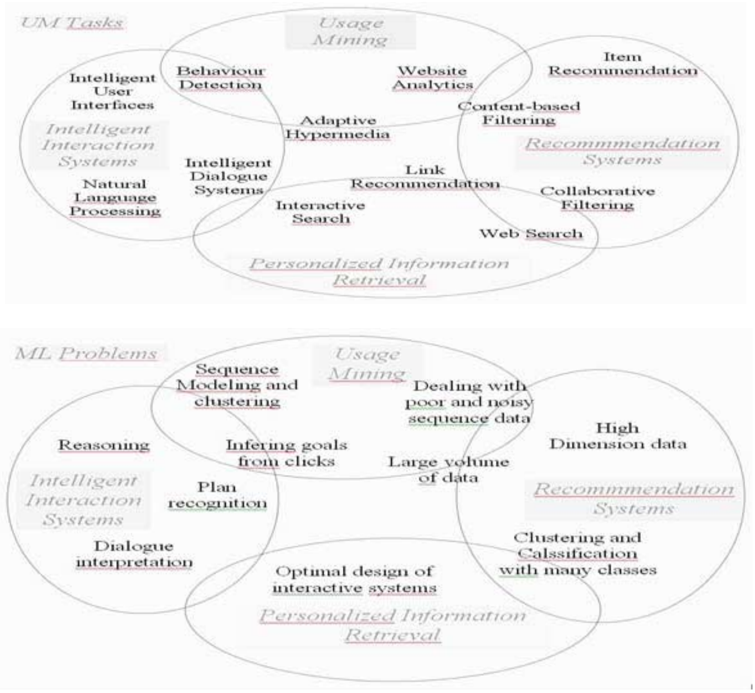

UM Tasks
Usage Mining
Intelligent User Interfaces
Behaviour Detection
Website Analytics
Item Recommendation
Content-based Filtering
Adaptive Hypermedia
Intelligent Interaction Systems
Recommmendation Systems
Intelligent Dialogue Systems
Link Recommendation
Natural Language Processing
Collaborative Filtering
Interactive Search
Web Search
Personalized Information Retrieval
ML Problems
Usage Mining
Sequence Modeling and clustering
Dealing with poor and noisy sequence data
High Dimension data
Reasoning
Infering goals from clicks
Large volume of data
Intelligent Interaction Systems
Plan recognition
Recommmendation Systems
Dialogue interpretation
Clustering and Calssification with many classes
Optimal design of interactive systems
Personalized Information Retrieval

Figure 1. Schematic map of broad categories of UM applications (top) and of ML related tasks and techniques (bottom).

1.1. Intelligent Interaction Systems

Today, in order to deal with an increasing number of complex applications a user should know all of its functionalities together with how to use it, while some of these functions often require a sequence of basic actions. Unfortunately, it is usually not possible for a user to even know all what his applications could do for him. Intelligent User Interface (IUI) aims at improving interaction between the user and the machine. IUI dates back the sixties with graphical user interface. It was not before the eighties that the user started becoming the focus of attention for interface design. One of the emblematic works in IUI was the *Lumiere* project which lead to the Office'97 Assistant [3]. The project aimed at providing tools for reasoning under uncertainty about the goals of a software user. It relies on Bayesian Networks, whose structure is chosen by hand, for encoding links between goals and needs. Another aspect of Intelligent Interaction Systems concerns the optimization of man-machine interaction, through dialogue interpretation. Dialogue interpretation is much related to natural language processing (NLP). It focuses on interpreting user arguments and may benefit from integrating a model of a user's beliefs and inferences [4]. Such systems often rely on Dynamic Bayesian networks and similar statistical tools.

Main ML problems in IIS tasks lie in the high cognitive level of modeling, i.e. the complexity of the information one wants to infer about the user. IIS aim to reason about the goals, preferences, actions of the user, hence usually require much prior information for designing the system (Bayesian networks structure etc). Hence IIS main difficulties lie more in the design of models for reasoning and in the integration of user modeling components in natural language processing tools (e.g. for dialogue).

1.2. Usage Modeling and Mining

A second series of UM tasks, such as Educational Hypermedia and Adaptive Hypermedia (AH), and Website Analytics solutions (WA), concern usage modeling and mining. Many of these applications are related to model, cluster or analyse user log sessions acquired through the interaction of a user with a system. Tasks differ in the environment where the user is observed (i.e. navigates). It may be much constrained, as in AH or approaching real life like in office activities help systems.

Traditional hypermedia systems have been described as "user-neutral" since they do not consider the characteristics of the individual user and provide the same content (pages) and the same links to all users [5]. The goal of AH is to improve the usability of such hypermedia through their automatic adaptation to individual users. This adaptation consists either in personalizing the content of the hypermedia, e.g. by summarizing documents based on the user's interests, or in personalizing the navigation through dynamic link adding, removing or recommending. A number of applications have emerged recently by considering wider user environments. [6] studied the segmentation of the multiple activities of a computer user, while the TaskTracer system [7] aims to detect task switch to configure the computer for the current task. [8] proposed a system for monitoring user activities in an office (Phone conversation, Face to Face conversation, Nobody present, etc), based on video, acoustic, and computer interactions. Finally, a slightly different task is faced by website analytics solutions such as WebTrends or E.Pihany (see [9] for a review). The goal of such tools, which are often integrated in Customer Relationship Management (CRM) systems, consists in providing business intelligence though high level knowledge about customers. It provides simple information such as the number of unique visitors per day to more complex measures about the efficiency of a marketing campaign. Some other tasks are related to usage mining such as interface design and interface evaluation based on various input data such as click streams or eye-movements.

From the ML point of view most of these tasks focus on the processing (clustering, correlation discovery etc) of observation sequences. They often rely on statistical models such as Markov models and variants (Hidden Markov Models, Hierarchical Hidden

Markov Models etc) to infer high level knowledge from low-level signals (e.g. click stream). As a consequence, while some of the systems are automatic [6], [8] others overcome the difficulty by relying on the cooperation of the user [7]. Usage mining usually comes with huge quantities of logs whose processing requires specific answers, for instance [10] suggests fast variants of HMMs.

1.3. Recommendation Systems

In the context of e-commerce, Recommender Systems (RS) aim to provide relevant information to the user (document, product, etc.) based on the experience learned from a (large) number of users. Many of the largest commerce Web sites (Amazon.com, CDNOW, eBay, Moviefinder.com …) are using recommender systems to help their customers find relevant products (e.g. to purchase) [11], [12]. Many recommendation systems are server-side and based on various filtering strategies. Content based filtering consists in recommending items similar to already bought items. Collaborative filtering consists in recommending items bought by similar users [13]. It offers the advantage of naturally handling taste which is otherwise difficult to model. Recommendation systems and adaptive hypermedia lead to few derived tasks such as Website Personalization (see My Yahoo!, My Netscape).

Recommender systems come with some major ML problems. First one usually has to deal (e.g. for clustering users) with incomplete data. Second, the industrial context of recommender systems (huge e-stores) induces a scale problem. There are many users and many products that require computationally efficient methods. Then one has to deal (e.g. to cluster users) with very high dimensional sparse data requiring efficient dimension reduction techniques. Also, a particularity of huge e-stores lies in the number of categories (i.e. items) to recommend. While some items are very well known, often bought or examined, the majority of products are rarely bought or known by the users. Being able to make recommendation for this large number of items without much training data is a today challenge. Finally, alike many other web-related domains, Recommender Systems have to face spam, e.g. crawling attacks aiming at altering the system's recommendation behaviour w.r.t. a particular item.

1.4. Information Retrieval

Information Retrieval (IR) has long been associated with UM[1]. Despite their indubitable capabilities today search engines cannot actually handle natural language queries since they are not able to deeply understand their meaning. Hence there is much room for improvement, one direction is personalization. A number of techniques have been proposed to personalize text search engines that are often rather simple ML extensions of IR techniques, such as tfidf weighting using history [14], query expansion based on previous queries, immediate result reranking, personalized page-rank (i.e. topic sensitive) [15]. Also note that some attempts have been made to implicitly acquire user information in an information retrieval task through eye-movements only. Some methods have also been proposed for search engines on more complex data (e.g. image, video) [16]. The difficulty of such tasks, hence the weakness of existing search engines,

[1] See the series of UM and IR workshops at UM conference.

makes the interaction acceptable for users. The solution is then to put the user in the loop and to design interactive search engines. This is a first step towards the actual integration of UM in search engines which would require designing specific tools and methods for learning user interaction.

A closely related problem consists in recommending links on the web. Since it is a very difficult task ML techniques are rather simple, it is then closer to recommender systems than to hypermedia navigation help systems. A number of systems have been proposed for web users that focus on helping the user to search for information, e.g. WebIC [17], and Web Watcher [18]. Both systems rely on the textual content of accessed pages. Web IC implements a kind of content-based filtering approach by looking for specific words based on the browsing patterns and seeks pages that contain these assumed relevant words. Web Watcher is more a collaborative filtering technique in that it relies on the annotation of any page by all the words in the queries of all users that reached the page. Lastly, a number of social media sites appeared in the last few years (Flickr, Delicious, CiteULike, YouTube) which play the role of recommender systems but usually without requiring ratings. These systems make use of contacts (through collaborative Filtering) and/or of metadata like tags on images (collaborative and content based Filtering).

In addition to inherited problems from Information Retrieval (including huge quantities of data), Personalized IR faces the real problem of optimally (from the ML point of view) integrating the user in the loop of interactive and adaptive IR systems. As pointed by [19], users are increasingly understood to be the driving force of the Internet, the problem is how to efficiently empower them.

2. User Models

The definition of a *user model* (UM) is usually an ad-hoc compromise between the available data from the user and what one wants to do with the user data. One can characterize a user model according to some features such as explicit vs. implicit (information asked to the user such as his age etc or automatically inferred from interaction), static (long-term) vs. dynamic (short-term), understandable vs. not understandable.

Simplest UMs are feature vectors. Stereotypes, introduced by [20], are typical sets of user characteristics (e.g. ratings of some items); it is a popular way for representing user knowledge in adaptive systems and recommender systems. Slightly more complex UMs are used in the information retrieval community (search engine personalization). One standard technique is to detect most significant words from the content of the documents the user accessed and to assume these words represent the user interests [21]. The EH community uses another kind of UMs, that rely on the hand-made definition of a domain model [22]. A *domain model* represents the knowledge about the concepts of the application and of their relations. It may be a tree or a graph of concepts with, eventually, typed edges (with semantic such as generalization-specialization). An overlay UM shares the same representation as the domain model and is used to represent a user in the concept space (knowledge, interest etc). The use of overlay user models has been extended to the AH field, however domain models are either unavailable or much more complex in AH than in EH. Techniques for automatic design of domain models have been proposed, ranging from using naïve domain models derived from the website structure, to the automatic discovery of concept hierarchies (i.e. ontology) from the hypermedia content which is a clear difficult ML problem (e.g. [23]). Among the least understandable user models are machine learning based user models that are usually implicit. For instance, a user model may consist in a neural network that predicts the next web page the user is expected to access. Of course the black-box feature of such UMs is a clear drawback since is prevent the user to know what the system knows about him and to interact with it [2].

3. Usage Log Processing

Usage log modeling and mining is a major component in a wide scope of user modeling applications described in Section 2. It is used in interface and hypermedia personalization, website analytics, personalized information retrieval, and in some cases for recommendation systems. However, although the goal is fundamentally similar (extracting information from sequences) techniques and tasks are much different when dealing with poor and noisy server-side logs for web applications or with richer and cleaner logs such as hypermedia usage logs.

3.1. Preprocessing and feature extraction

A first step before using data mining and machine learning tools is the preprocessing of the logs [24]. This is a crucial step especially for web usage mining which are usually gathered on the server side and are very noisy and inaccurate, due to the poorness of server logs and to proxy and cache technologies. Preprocessing includes logs cleaning, session identification, and user identification, which are actually difficult tasks. As an illustration of this difficulty the very first non-trivial information that produces a website analytics solution is the number of unique visitors per day. The preprocessing step often relies on a number of ad-hoc parameters and on heuristics that lead to inaccuracy and unreliable information, it would benefit from a more principled ML approach.

When dealing with usage logs the preprocessing step is completed with a feature extraction stage where one has to choose the representation of a session that will be processed by the system (e.g. HMMs). One may decide to represent a session by a sequence of feature vectors, where a feature vector is computed for every page accessed or for a fixed duration (e.g. every 30 seconds) [6], [25], [26]. Then one has to define features. In adaptive hypermedia applications for instance a session log is a rich sequence of events such as a click on a link, on an image, on the print button, the use of the vertical scroll. One may compute many features that characterize the path in the hypermedia (with cycles, backs etc), the user activity (number of scrolls, clicks etc). Furthermore, the log input sequence is often augmented with additional information such as the content of the accessed pages, which may be exploited to compute higher level features such as the average textual distance between two accessed pages etc.

3.2. Mining Sequential patterns

Mining user logs can be done with a few techniques and may serve multiple goals. Two main categories of methods are used. First one can put all the data in a data base and use appropriate methods (e.g. association rules). Second one can use machine learning techniques (e.g. HMMs) for dealing with raw observation sequences.

A first series of application consists in inferring simple reporting information about the user's browsing behaviour. Association rules are one of the simplest techniques for such a task. In fact, this is one technique that does scale with huge quantities of logs where most methods don't. It is used to extract information about co-occurrence of pages in a session, which help to understand general trends of user's behaviour [9, 24]. A second series of application is more related to prediction of the next page for cache optimization and prefetching. The idea is for instance to optimize navigators' cache by presending the pages with high probability to be accessed in the future [26]. Many such systems exploit Markovian models. A third series of tasks consists in clustering and classifying users or user's sessions, e.g. for clustering users or for designing help systems. One may use clustering algorithms on fixed dimension representations of sequences [27]. Alternatively one may directly operate on raw sequences and use standard clustering algorithms together with a sequence alignment-based distance between sequences (e.g. dynamic programming). Alternatively one can use probabilistic models. For instance, [28] and [29] investigate model-driven clustering, i.e. learning a mixture model, for identifying typical categories of sequences, where component models are variants of Markovian models (Markov chains, HMMs, Multi-stream HMM etc).

One major task when dealing with rich logs such as in adaptive hypermedia is the detection and tracking of user's navigation behaviour. Often one considers a typology of typical navigation behaviours that every user is assumed to adopt in his/her navigation. Actually a common and intuitive hypothesis consists in assuming that a user that navigates on the web or in a hypermedia will act differently according to his/her short term goals. A few characterization of user navigation patterns have been proposed. For instance [30] proposed a typology which distinguishes between a few broad behaviours such as searching for particular information, browsing without precise goal, scanning a specific part of the hypermedia/web, etc. A number of studies have focused on the use of various Markovian models for detecting and tracking user navigation behaviour within such taxonomy ([25], [29]) with the aim to recommend a relevant link to the user according to his browsing history. When processing such logs with Markovian models, one has to decide what a state means. It may be a page of the hypermedia [26] or something that depends implicitly on the history of the navigation [25]. A conclusion of these studies is that the search behaviour is by far easier to recognize than other navigation patterns, and that the corresponding help strategy is the most efficient. These works also show that relying on a general taxonomy of user behaviors (like in [30]) is often not informative enough to provide relevant link recommendation. Instead a more promising idea is to automatically discover behaviour taxonomy from a corpus of sessions, which may be viewed in the context of Markovian modeling as a problem of Markovian topology learning from data [29].

4. Conclusion

Machine Learning has long been associated to User Modeling. The scope of UM applications is very wide so that all the ML paradigms and all the ML techniques may be useful in a particular UM settings. We presented a panorama of the UM field and a few of today's ML-related problems, dealing with high dimensional data for recommender systems, integrating the user in the loop in information retrieval systems, detection and tracking of user behaviour from click-streams for usage mining.

5. Bibliography

[1] Frias-Martinez E., Chen, S.Y., Liu X., Survey of Data Mining Approaches to User Modeling for Adaptive Hypermedia, IEEE Transactions on Systems, Man and Cybernetics: Part C, 2006, 36 (6) : 734-749.

[2] Webb, G. I., Pazzani, M. J., and Billsus, D., Machine Learning for User Modeling, User Modeling and User-Adapted Interaction 11, 1-2, Mar. 2001, 19-29.

[3] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users, UAI, 1998.

[4] Zukerman I., and George S., A Probabilistic Approach for Argument Interpretation, User Modeling and User-Adapted Interaction, 15(1), 2005.

[5] Brusilovsky P., Adaptive Hypermedia, User Modeling and User-Adapted Interaction, 11(1/2), pages 87-110, 2001.

[6] Slaney M., Subrahmonia J., Maglio P., Modeling Multitasking Users, User Modeling, 2003.

[7] Dragunov A., Dietterich T., Johnsrude K., McLaughlin M., Li L., Herlocker J., TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers, International Conference on Intelligent User Interfaces, 2005.

[8] Oliver N., Horvitz E., and Garg A., Layered representations for human activity recognition, Int. Conf. on Multimodal Interfaces, 2002.

[9] Eirinaki M., Vazirgiannis M., Web Mining for Web Personalization, ACM Transactions on Internet Technologies, Vol. 3, n° 1, Februray 2003, pp 1-27.

[10] Felzenszwalb P., Huttenlocher D., Kleinberg J., Fast Algorithms for Large-State-Space HMMs with Applications to Web Usage Analysis, NIPS, 2003.

[11] Schafer J., Konstan J., Riedl J., Recommender systems in e-commerce, ACM Conference on Electronic Commerce, 1999.

[12] McCarthy J. F., Anagnost T. D., MusicFX: An Arbiter of Group Preferences for Computer Aupported Collaborative Workouts, Int. Conference on Computer Supported Cooperative Work, 1998.

[13] Bueno D., Conejo R. and David A., METIOREW: An Objective Oriented Content Based and Collaborative Recommending System, ACM Conference on Hypertext and Hypermedia, 2001.

[14] Shen X., Tan B., Zhai C., 2005 Implicit user modeling for personalized search, international conference on Information and knowledge management, 2005.

[15] Haveliwala T.H.. Topic-Sensitive PageRank, International World Wide Web Conference, 2002.

[16] Sebe N., Tian Q., Personalized Multimedia Retrieval: The New Trend?, International Workshop on Multimedia Information Retrieval (Session on Personalized Multimedia Information Retrieval), 2007.

[17] Zhu T., Greiner R., Haübl G., Learning a model of a web user's interests, User Modeling 2003.

[18] Joachims T., Freitag D., Mitchell T., WebWatcher: A Tour Guide for the World Wide Web, International Joint Conference on Artificial Intelligence (IJCAI), 1997.

[19] Dupret G., Piwowarski B., Hurtado C., Mendoza M., A Statistical Model of Query Log Generation, in String Processing and Information Retrieval, LNCS, Springer Berlin / Heidelberg, pp 217-228, 2006.

[20] Rich, E. (1979) User modeling via stereotypes. in Cognitive science, Vol. 3, pp. 329-354.

[21] Sugiyama K., Hatano K., Yoshikawa M., Adaptive web search based on user profile constructed without any effort from users, International conference on the WWW, 2004.

[22] De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Aha! The adaptive hypermedia architecture, HT'03.

[23] Njike H., Artières T., Gallinari P., Blanchard J., Letellier G., Automatic learning of domain model for personalized hypermedia applications, IJCAI, 2005.

[24] Cooley R., Mobasher B., Srivastava J., Data preparation for mining world wide web browsing patterns, Journal of Knowledge and Information Systems, 1, (1999), 5—32.

[25] Bidel S., Lemoine L., Piat F., Artières T., Gallinari P., Classification and tracking of hypermedia navigation patterns, International Conference on Neural Networks (ICANN), 2003.

[26] Zukerman I., Albrecht D.W., 2001, Predictive statistical models for user modelling, User Modeling and User-Adapted Interaction, 11(1/2):5-18, 2001.

[27] Draier T., Gallinari P., Characterizing Sequences of User Actions for Access Logs Analysis, User Modeling, 2001.

[28] Cadez I., Gaffney S., Smyth P., A general probabilistic framework for clustering individuals and objects, KDD, 2000.

[29] Binsztok H., Artières T., Gallinari P., A model-based approach to sequence clustering, European Conference on Artificial Intelligence (ECAI), 2004.

[30] Canter D., Rivers R., Storrs G., Characterizing user navigation through complex data structures, Behavior and information technology, vol 4, 1995.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-85

Smoothness and sparsity tuning for Semi-Supervised SVM

Gilles GASSO, Karina ZAPIEN and Stephane CANU
INSA-Rouen, LITIS EA 4108
BP 08 - 76801 Saint-Etienne du Rouvray, France

Abstract. Using unlabeled data to unravel the structure of the data to leverage the learning process is the goal of semi supervised learning. Kernel framework allows to model the data structure using graphs and to build kernel machines such as Laplacian SVM [1]. But a remark is the lack of sparsity in variables of the obtained model leading to a long running time for classification of new points. We provide a way of alleviating this problem by using a L_1 penalty and a algorithm to efficiently compute the solution. Empirical evidence shows the benefit of the algorithm.

Keywords. Semi-supervised learning, Laplacian SVM, sparsity, regularization path

Introduction

Semi-supervised learning addresses the problem of a database containing few labeled data and a relative large amount of unlabeled data. This situation can occur when the cost of giving a label to each point by an expert is high. The question arises if the knowledge of the labels of few points is sufficient to construct a decision function able to guess the correct class of the unlabeled data. Different approaches have been proposed [2] and many of them rely more or less on the *cluster assumption*: labeled and unlabeled data can be clustered and two points which are "close" share the same label. Thus, some algorithms determine a decision function which avoids the high density regions. Other algorithms use graphs to represent the data structure. Flexibility of the maximum margin kernel framework allows to model graph smoothness and to build algorithms such as Laplacian SVM [1]. This adopts a geometric point of view: assuming the data lie on submanifolds, the decision function must avoid passing through these manifolds. Hence, a penalty term preserving the manifolds is added to the classical L_2-SVM problem (classification of the ℓ labeled data (x_i, y_i)) via the Laplacian L of the adjacency graph [1]:

$$\begin{cases} \min_{\boldsymbol{\beta},b,\xi} \sum_{i=1}^{\ell} \xi_i + \frac{\lambda_2}{2}\boldsymbol{\beta}^\top K\boldsymbol{\beta} + \frac{\mu}{2}\boldsymbol{\beta}^\top KLK\boldsymbol{\beta} \\ \text{s.t.} \qquad y_i f(x_i) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \;\; i = 1, \ldots, \ell \end{cases} \tag{1}$$

K is the kernel matrix of labeled and unlabeled data, λ_2 is the L_2-norm regularization parameter whereas the parameter μ takes into account the manifold constraint. The decision function $f(x) = \sum_{j=1}^{\ell+u} \beta_j k(x, x_j) + b$ (with k the kernel) involves all the labeled as well as unlabeled data. Generally, the function f is not sparse in variables (few parameters β_j are null) leading to a long running time for classification of new points. To alleviate this drawback, we propose a sparse version of the algorithm.

1. Tuning sparsity and smoothness of Laplacian SVM

To obtain a sparse solution, we explicitly include in the learning problem (1) a L_1-norm constraint: $\sum_{j=1}^{\ell+u} |\beta_j| \leq s$ in order to yield a decision function depending only on relevant training points (notice that according to the kernel trick, there is a duality between the training points x_i and the variables $k(x_i, .)$ induced by the kernel). Making the regularization parameter s small tends to cause some of the coefficients β_j to be null. With this formulation, it is necessary to realize a trade off between the sparsity, the manifold constraints and the L_2 penalty (smoothness of f) by tuning appropriately s, μ and λ_2. To examine the evolution of the decision function w.r.t to these hyperparameters, we compute regularization paths. According to [3,4], it can be shown that the solution paths are piecewise linear and can be computed in a smart and efficient way. Indeed, for instance, if μ and λ_2 are fixed, we can easily compute the sparsity path i.e. the set of all solutions $\{f_s(x), 0 \leq s \leq \infty\}$. At step $t+1$, the parameters of the model are obtained from the previous step via the linear relation : $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + (s - s_t)\boldsymbol{\eta}_t$ where $\boldsymbol{\theta} = [\boldsymbol{\beta}^\top, b]^\top$. The slope vector $\boldsymbol{\eta}$ is the solution of a linear system of size card($\mathcal{A}$) + card($\mathcal{E}$). $\mathcal{A}$ and $\mathcal{E}$ are respectively the sets of active variables ($\beta_j \neq 0$) and labeled margin points [4]. If μ and s are fixed, one can derived a similar algorithm based on the same mechanism leading to the smoothness path which computes the solutions $\{f_{\lambda_2}(x), 0 \leq \lambda_2 \leq \infty\}$.

Proposed methodology As the simultaneous determination of the hyper-parameters μ, s and λ_2 is a difficult task, we propose the following scheme: the λ_2-path is run assuming μ fixed, and no sparsity constraint (this is equivalent to $s = \infty$). The best non sparse model $f^*_{\lambda_2}(x)$ is identified in this way using a validation procedure. Then, the sparsity path is applied to the later model to yield the best sparse model $f^*_{s,\lambda_2}(x)$.

Results Applications of the procedure to the USPS handwritten digit database are reported in tables 1. ℓ labeled points per class where selected randomly. The algorithm was stopped when the error on the unlabeled data is of the same order as the error on the best original Laplacian SVM.

Table 1. Some results on USPS database. $|\mathcal{A}| = n$ for the original Laplacian SVM solution. Seven runs are done, the error is the average number of misclassified points. The values in brackets are standard deviations

(a) Training of class *2* against *5* - $n = \ell + u = 847$ training points

ℓ	16	32	64
$\lvert\mathcal{A}\rvert$	132 (104.17)	126 (154.18)	105.87 (114)
Test error	19 (8.73)	13 (2.78)	9 (3.33)

(b) Training of class *1* against *7* - $n = \ell + u = 954$ training points

ℓ	16	32	64
$\lvert\mathcal{A}\rvert$	348 (118.7)	317 (37.42)	408 (190.40)
Test error	36.8 (23.63)	23.25 (6.19)	13.75 (2.63)

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometrc framework for learning from label and unlabeled examples. *Journal of Machine Learning Research*, 1:1–48, 2006.

[2] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[3] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *JMLR*, 5:1391–1415, 2004.

[4] L. Wang, J. Zhu, and H. Zou. The doubly regularized support vector machine. *Statistica Sinica*, 16:589–615, 2006.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-87

Distributed Categorizer for Large Category Systems

Eric GAUSSIER [a] and Francois PACULL [b,1]
[a] *University Joseph Fourier, Grenoble, France*
[b] *Xerox Research Centre Europe, Meylan, France*

Abstract. We describe here a categorization system that allows to manage an arbitrary number of categories. It is based on a cascade of categorizers arranged as a tree. Categorizers are deployed on a set of machines to parallelize the processing.

Keywords. Categorizer, Large Scale System.

Introduction

The memory and time complexity is a crucial problem for categorization algorithms. Indeed, the complexity is somehow linked to the number of considered classes multiplied by the size of the vocabulary used during the training phase. Managing a large category system with a single standalone categorizer is most often impossible since these two parameters increase at the same time as the category system grows.

The solution proposed here allows to manage an arbitrary number of categories. It is based on a cascade of categorizers arranged as a tree. Each node of the tree is a categorizer the role of which is either to find the next categorizer(s) (for the *intermediary* categorizers) or the actual categories (for the *final* categorizers). From a practical point of view, the intermediary categorizers act as coarse-grained switches while the final categorizers play their classical roles and return the possible matching categories with the corresponding probabilities. These probabilities are then aggregated and unified to define the final result.

1. Definition of the category system

In figure 1-a, we considered a system based on a significant part of the Dmoz Open Directory Project *(http://www.dmoz.org)*: namely the categories corresponding to the topics *Arts, Business, Home, Reference, Science, Computers, News, Regional, Society, Health, Recreation* and *Sports*. For the decomposition into intermediary categorizers, we first tried the most intuitive that consists of considering the set of immediate sub-categories as shown in the figure 1-b. However, when categorizers of a level are not homogeneous in term of sub-categories and vocabulary, including further sub-categories has shown to improve the results.

[1] Corresponding Author: Francois.Pacull@xrce.xerox.com

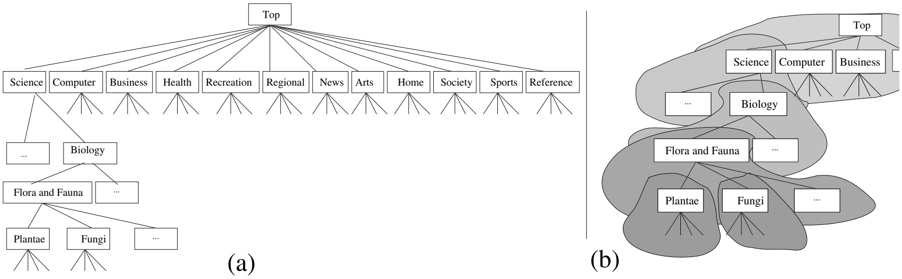


Figure 1. System based on a significant part of Dmoz Open Directory Project.

2. Cascading processing and aggregation of the results

The cascade approach allows several algorithms and strategies to define at each intermediary level which paths have to be followed. For instance, we can consider to follow several paths in parallel if we want to process documents containing different aspects (e.g. *Computer* and *Business* or *Sciences* and *Health*). Alternatively, we can decide to really focus on the main category and then to follow a direct path to a single final categorizer.

Since the results of the categorization returned by final categorizers are independent we have to aggregate them. To do so, we have used calibration [1] to re-rank the different results in order to provide an homogeneous result across the whole category system.

Conclusion

The main advantages of our architecture are: (i) it uses a classical categorizer (Probabilistic Latent categorizer [2]) as a black box for the nodes of the systems: it is not required to modify the code of the categorizer; (ii) it allows to consider a very large number of categories: we have successfully experimented systems based on the United States Patent and Trademark (USPTO *http://www.uspto.gov*) category system (15 000 categories) and a larger one (100 000 categories) based on Dmoz. (iii) the cascading approach is flexible enough to really cope with the desired results and the specificity of the category system. (iv) it is possible to put in place a very efficient and low cost system: because the different categorizers do not require a lot of resources and are independent, the whole system can be distributed on a set of basic desktops. Then processing several times the same document (up to 5 in practice) is highly compensated by the pipelining and the parallelisation of the processing. We reached almost six hundred documents per minute with the system of 100 000 categories deployed on just 5 machines.

References

[1] Cohen, I., Goldszmidt, M.: Properties and benefits of calibrated classifiers. In: Proc. of 15th European Conference on Machine Learning (ECML). (2004)

[2] Gaussier, E., Goutte, C., Popat, K., Chen, F.: A hierarchical model for clustering and categorising documents. In: Proc. of the 24th BCS-IRSG Colloquium on IR Research (ECIR'02). LNCS, Springer (2002)

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-89

Data stream management and mining

Georges HEBRAIL
TELECOM ParisTech, CNRS LTCI
Paris, France

Abstract. This paper provides an introduction to the field of data stream management and mining. The increase of data production in operational information systems prevents from monitoring these systems with the old paradigm of storing data before analyzing it. New approaches have been developed recently to process data 'on the fly' considering they are produced in the form of structured data streams. These approaches cover both querying and mining data.

Keywords. Data processing, data streams, querying data streams, mining data streams.

Introduction

Human activity is nowadays massively supported by computerized systems. These systems handle data to achieve their operational goals and it is often of great interest to query and mine such data with a different goal: the supervision of the system. The supervision process is often difficult (or impossible) to run because the amount of data to analyze is too large to be stored in a database before being processed, due in particular to its historical dimension.

This problem has recently been studied intensively, mainly by researchers from the database field. A new model of data management has been defined to handle "data streams" which are infinite sequences of structured records arriving continuously in real time. This model is supported by newly designed data processing systems called "*Data Stream Management Systems*". These systems can connect to one or several stream sources and are able to process "*continuous queries*" applied both to streams and standard data tables. These queries are qualified as continuous because they stay active for a long time while streaming data are transient. The key feature of these systems is that data produced by streams are not stored permanently but processed 'on the fly'. Note that this is the opposite of standard database systems where data are permanent and queries are transient. Such continuous queries are used typically either to produce alarms when some events occur or to build aggregated historical data from raw data produced by input streams.

As data stored in data bases and warehouses are processed by mining algorithms, it is interesting to mine data streams, i.e. to apply data mining algorithms directly to streams instead of storing them beforehand in a database. This second problem has also been studied and new data mining algorithms have been developed to be applicable directly to streams. These new algorithms process data streams 'on the fly' and can provide results either based on a portion of the stream or on the whole stream already seen. Portions of streams are defined by fixed or sliding windows.

Several papers, tutorials and books have defined the concept of data streams (see for instance [1], [2], [3], [4], [5]). We recall here the definition of a data stream given in [3]:

"A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor is it feasible to locally store a stream in its entirety."

The streams considered here are streams of items (or elements) in the form of structured data and should not be confused with streams of audio and video data. Real-time means here also that data streams produce massive volumes of data arriving at a very high rate. Table 1 shows an example of a data stream representing electric power consumption measured by many communicating meters in households (see [6]).

Table 1. Example of a data stream describing electric power metering

Timestamp	Meter	Active P (kW)	Reactive P (kVAR)	U (V)	I (A)
…	…	…	…	…	…
16/12/2006-17:26:12	86456422	5,374	0,498	233,29	23
16/12/2006-17:26:32	64575861	5,388	0,502	233,74	23
16/12/2006-17:26:11	89764531	3,666	0,528	235,68	15,8
16/12/2006-17:29:28	25647898	3,52	0,522	235,02	15
…		…	…	…	…

This paper provides an introduction to the data stream management and mining field. First, the main applications which motivated these developments are presented (telecommunications, computer networks, stock market, security, …) and the new concepts related to data streams are introduced (structure of a stream, timestamps, time windows, …). A second part introduces the main concepts related to Data Stream Management Systems from a user point of view. The third part presents some results about the adaptation of data mining algorithms to the case of streams. Finally, some solutions to summarize data streams are briefly presented since they can be useful either to process streams of data arriving at a very high rate or to keep track of past data without storing them entirely.

1. Applications of data stream processing

Data stream processing covers two main aspects: (1) processing queries defined on one or several data streams, possibly also including static data; (2) performing data mining tasks directly on streams, usually on a unique stream. For both of these aspects, the following requirements should be met:

- Processing is done in real-time, following the arrival rate in the streams and not crashing if too many items arrive
- Each item of the stream is processed only once (one-pass processing)

- Some temporary storage of items of the streams may be considered but with a bounded size

We divide the applications of data stream processing in two categories:

- **Real-time monitoring (supervision) of Information Systems**. Whatever the sector of activity, current Information Systems (IS) manage more and more data for supporting human activity. The supervision of these systems (for instance supervision of a telecommunication network) requires the analysis of more and more data. It is not possible anymore to adopt the ‘store then process’ strategy which has been used until now by maintaining large data warehouses. There is a strong need for real-time supervision applied on detailed data instead of batch supervision applied on aggregate data. This can only done by processing data on the fly which is the approach developed by data stream processing.

- **Generic software for operational systems managing streaming data**. There are many operational Information Systems for which the main activity is to process streams of data arriving at a high rate. It is the case for instance in finance where computerized systems assist traders by analyzing automatically the evolution of the stock market. Such systems are today developed without any generic tool to process streams, exactly as first Information Systems were developed using files to store data before data base technology was available. Data stream processing solutions will provide generic software to process queries on streams, analyze streams with mining algorithms and broadcast resulting streams.

Supervision of computer networks and telecommunication networks has been one of the first applications motivating research on data streams. Supervision of computer networks consists in analyzing IP traffic on routers in order to detect either technical problems on the network (congestion due to undersized equipment) or usage problems (unauthorized intrusion and activity). These applications fall into the first category of real-time monitoring of IS. The size of data to be analysed is huge since the utmost detailed item to be analyzed is an IP packet. Typical rate is several tens of thousands of records per second to be processed when raw detailed data is already sampled by 1/100. Typical queries on streams for these applications are the following: *find the 100 most frequent couples of IP addresses (@sender, @destination) on router R1, how many different couples (@sender, @destination) seen on R1 but not R2, and this during the last hour ?*

Finance is today the domain where first commercial data stream management systems (see [7] and [8] for instance) are used in an industrial way. The main application in finance is to support trading decisions by analyzing price and sales volume of stocks over time. These applications fall into the second category of operational systems whose first goal is to manage streams of data. Typical queries on streams for these applications are the following: *find stocks whose price has increased by 3% in the last hour and the volume of sales by 20% in the last 15 minutes.*

An interesting other (artificial) application is the “Linear Road Benchmark” (see [9]). The “Linear Road Benchmark” is a simulator of road traffic which has been designed to compare the performance of different Data Stream Management Systems. It simulates road traffic on an artificial highway network. Vehicles transmit continuously their position and speed: all data are received by the Data Stream Management System which is in charge of supervising traffic by computing for each vehicle a toll price depending on the heaviness of the traffic: high traffic on a highway portion leads to a high price. Computed toll are sent in real-time to vehicles so that they can adapt their route to pay less. This system – though only imaginary – is interesting because it falls in both of the categories described above: the result of a real-time supervision is directly re-injected in the operational system.

Other applications are described in [3] and concern web log analysis and sensor networks.

2. Models for data streams

2.1. Structure of data streams

As recalled in the introduction, a data stream is an infinite sequence of items (elements). Usually, items are represented by a record structured into fields (i.e. a tuple), all items from the same stream having the same structure. Each item is associated with a *timestamp*, either by a date field in the record (this is referred to *explicit* timestamping) or by a timestamp given when an item enters the system (this is referred to *implicit* timestamping). Still related to the representation of time in streams, the timestamp can be either *physical* (i.e. a date) or *logical* (i.e. an integer which numbers the items). Implicit timestamping ensures that items arrive in the order of timestamps while it is not true for explicit timestamping. Table 1 showed an example of explicit physical timestamp. Table 2 shows an example of data stream figuring IP sessions described by implicit logical timestamps.

Table 2. Example of a data stream describing IP sessions

Timestamp	Source	Destination	Duration	Bytes	Protocol
…	…	…	…	…	…
12342	10.1.0.2	16.2.3.7	12	20K	http
12343	18.6.7.1	12.4.0.3	16	24K	http
12344	12.4.3.8	14.8.7.4	26	58K	http
12345	19.7.1.2	16.5.5.8	18	80K	ftp
…	…	…	…	…	…

2.2. Model of data streams

Following [4], the contents of a stream describe observed values which have to be related to one or several underlying signals: the *model* of the stream defines this relationship. An observation is defined by a couple (t_i, m_i) where t_i is a timestamp and

m_i is a tuple of observed values. For instance in Table 2, $t_i=i$ since the timestamp is logical and m_{12342}=*(10.1.0.2, 16.2.3.7, 12, 20K, http)*.

There are several possibilities to define *one* underlying signal from observations. For instance in the stream of Table 2, one may be interested in:

- the number of bytes transmitted at each timestamp. This is called the *time series model* where each observation gives directly the value of the underlying signal.
- the total number of bytes going through the router since the beginning of the stream. This is called the *cash register model* where each observation gives a positive increment to add to the previous value of the signal to obtain the new value. When increments can be negative the model is called the *turnstile model.*

Another possibility is to define *several* signals from a single stream. This is the case when the stream contains information related to several objects: one signal per object may be extracted from the stream. For instance in Table 1, one may define one signal per meter and in Table 2, one may define one signal per @sender IP address or per couple (@sender, @destination). Again the model can be the time series, the cash register or the turnstile model for each underlying signal.

Depending on the model of the stream, the transformation of observations to the defined underlying signals may be easy or difficult. For instance, the transformation requires large computations when there is a very large number of underlying signals (for instance in Table 2 if there is one signal per @sender IP address).

2.3. Windows on data streams

In many applications, the end-user is not interested in the contents of the whole stream since its beginning but only in a portion of it, defined by a *window* on the stream. For instance, a typical query may be: *find the 100 most frequent @sender IP address during the last hour*. There are several ways to define windows on a stream:

- The window can be defined *logically* (in terms of number of elements, ex: the last 1000 elements) or *physically* (in terms of time duration, ex: the last 30 minutes).
- The window can be either a *fixed window* (defined by fixed bounds, ex: September 2007), a *sliding window* (defined by moving bounds, ex: last 30 minutes), or a *landmark window* (one fixed and one moving bound, ex: since September, 1st, 2007).

Finally, the end-user has to define the *refreshment rate* for the production of the results of queries/data mining tasks (ex: *give, every 10 minutes, the 100 most frequent @sender IP address during the last hour*). Again the refreshment rate can be defined either logically or physically.

3. Data stream management systems

Data Base Management Systems (DBMS) are software packages which help to manage data stored on disk. A model is available to define structures of data (usually the relational model) and a query language enables to define/modify structures and to insert/modify/retrieve data in the defined structures. The most popular query language is the SQL query language. A DBMS also offers other services like transaction processing, concurrency control, access permission, backup, recovery, … It is important to note that in a DBMS data is stored permanently and that queries are transient.

Though there is not yet a stable definition of a Data Stream Management System (DSMS), DSMSs can be seen as extensions of DBMSs which also support data streams. The structure of a data stream is defined in the same way as a table in a relational database (i.e. list of fields/attributes associated with a data type), but there is no associated disk storage with a stream. A data stream is connected to a source where data arrives continuously (for instance a TCP/IP port) and there is no control on the rate of arrival in the source. If the DSMS is not able to read all data in the source, the unread data is lost.

3.1. Continuous queries

The key feature of a DSMS is that it extends the query language to streams. The semantics of a query has to be redefined when it is applied to streams. The concept of *continuous query* has been introduced to do so. The result of a query on a stream can be either another stream or the maintenance of a permanent table which is updated from the contents of the streams. Thus queries are permanent and process data 'on the fly' when it arrives: this explains why queries on streams are called continuous.

Let us consider the following stream describing the orders from a department store, sent to this stream as there are created:

```
ORDERS (DATE, ID_ORDER, ID_CUSTOMER, ID_DEPT, TOTAL_AMOUNT)
```

The following query filters orders with a large amount:

```
SELECT DATE, ID_CUSTOMER, ID_DEPT, TOTAL_AMOUNT
FROM ORDERS
WHERE TOTAL_AMOUNT > 10000;
```

This query can be processed 'on the fly' (without storing the whole stream) if its result is output as another stream.

Let us now consider the following query computing the sum of sales per department:

```
SELECT ID_DEPT, SUM(TOTAL_AMOUNT)
FROM ORDERS
GROUP BY ID_DEPT;
```

The result of this query is typically a permanent table which is updated from the contents of the stream. Note that the stream does not need to be stored since the update of the result can be done incrementally. It is also possible to generate another stream from the permanent updated table by sending to this new stream at the refreshment rate either the whole table, the inserted tuples or the deleted tuples between the refreshment timestamps.

These two queries can be extended easily to join the stream to a permanent table describing customers, for instance for the last query:

```
SELECT ORDER.ID_DEPT, CUSTOMER.COUNTRY, SUM(TOTAL_AMOUNT)
FROM ORDERS, CUSTOMER
WHERE ORDERS.ID_CUSTOMERS=CUSTOMER.ID_CUSTOMER
GROUP BY ID_DEPT, CUSTOMER.COUNTRY;
```

Again, this query can be evaluated without storing the stream, by joining on the fly the tuples from the stream to the ones of the permanent table and maintaining incrementally the aggregates. This is true when the customer information is available in the permanent table at the time an order corresponding to the customer arrives in the stream.

But things are not always so easy. There are many cases where it is not possible to produce the result of the query without storing the whole stream. We define a second stream containing the different bills associated with orders, several bills being possibly associated with one order and arriving at different times:

```
BILLS (DATE, ID_BILL, ID_ORDER, BILL_AMOUNT)
```

Let us consider the following query which is also an example illustrating that joins of several streams can be handled by a DSMS:

```
SELECT ORDERS.ID_ORDER, TOTAL_AMOUNT - SUM(BILL_AMOUNT)
FROM ORDERS, BILLS
WHERE ORDERS.ID_ORDER = BILL.ID_ORDER
GROUP BY ORDERS.ID_ORDER, TOTAL_AMOUNT
HAVING TOTAL_AMOUNT - SUM(BILL_AMOUNT) != 0;
```

This query lists the orders for which the total amount has not been covered exactly by bills. To process this query, it is not necessary to keep in memory the elements of the BILLS stream once they have been joined to their corresponding ORDERS element. But it is necessary to keep all elements of the ORDERS stream: indeed, since the number of bills associated with an order is not known, the system always has to expect a new bill to appear for each arrived order. Such queries are called ‘blocking queries’ because they cannot be evaluated ‘on the fly’. There are several solutions to this problem and the most common one is to define windows on the streams. The user specifies on each stream a window on which the query applies, for instance:

```
SELECT ORDERS.ID_ORDER, TOTAL_AMOUNT - SUM(BILL_AMOUNT)
FROM ORDERS [LAST 30 DAYS], BILLS [LAST DAY]
WHERE ORDERS.ID_ORDER = BILL.ID_ORDER
GROUP BY ORDERS.ID_ORDER, TOTAL_AMOUNT
HAVING TOTAL_AMOUNT - SUM(BILL_AMOUNT) != 0;
```

This new query is restricted to the production of the result considering only the orders of the last 30 days and the bills of the last day. The query is then 'unblocked' and can be evaluated with a limited temporary storage of the contents of the streams (last 30 days for orders and at the most last day for bills). It is interesting to note that the end-user himself specifies how to limit the scope of the query by accepting that bills not issued 30 days after their orders will never be considered to produce a result in this query.

3.2. Main features of a DSMS

As mentioned before, a DSMS provides a data model to handle both permanent tables and streams. It enables the user to define continuous queries which apply to both permanent tables and streams and produce either new streams or update some other permanent tables. There are two main approaches to provide such a facility to the user: (1) the definition of an extension of the SQL language (see for instance the STREAM [10] and TelegraphCQ [11] systems), (2) the definition of operators applicable to streams which are combined to produce the desired result (see for instance the Aurora system [12]).

When designing a DSMS, several problems have to be solved, the most challenging ones being the following:

- Defining for a set of continuous queries an optimized execution plan which reduces temporary storage and allows shared temporary storage between the queries. Note that the typical use of a DSMS (see the Linear Road benchmark in [9]) leads to the definition of several queries producing intermediate streams which are then used by several other queries. Such execution plans have to be dynamic because the rate of arrival in streams may evolve and new queries may be defined by the user. Interesting solutions have been studied in the STREAM project (see [10]).
- Being able to face important variations in rates of arrival in input streams. As seen before, data arriving in data streams are either processed immediately or lost. DSMSs must ensure that there is no crash in this case and that the performance deteriorates in a controlled way. The main approach – called *load shedding* - is to place random sampling operators so that all elements of the streams are not processed when the system is overloaded. The placement of these operators is optimized dynamically to maximize a function of quality of service (see [12], [13] for instance).

3.3. Main existing DSMSs

Existing DSMSs fall into two main categories: (1) general purpose DSMSs, (2) specialized DSMSs dedicated to a particular domain.

General-purpose DSMSs

In this category, the most famous ones are the following research prototypes: STREAM ([10]), TelegraphCQ ([11]), and Aurora-Medusa-Borealis ([12]). STREAM is the result of a project at Stanford University: a new system has been built from scratch and has introduced many new concepts and solutions related to the management of data streams. In particular, the CQL language has been defined as an extension of the SQL language to handle continuous queries with windowing facilities. Though a demo is available on-line and the software can be downloaded, it is not really usable in practice (see [6]). TelegraphCQ has been developed at the University of Berkeley as an extension to the PostgreSQL system to support streams and continuous queries defined on streams. New continuous queries can be added dynamically to the system and it is usable for applications where there is no need to process very fast streams. The Aurora system has been developed at Brown University, MIT and Brandeis. In this system, queries are defined by combining operators which take streams as input and produce a new stream as output. A lot of interesting work has been done on load shedding by defining and optimizing a function of quality of service when the system becomes overloaded. The follow-up projects Medusa and Borealis focused on a distributed version of Aurora.

There are also a few commercial general-purpose DSMSs. The Aurora project led to the Streambase software (see [8]). The TelegraphCQ project led to the Aminsight/Truviso software (see [7]).

Specialized DSMSs

Specialized DSMSs have been developed to solve data stream problems from a particular domain of application. In these systems, specific operations of the domain are optimized. The most popular systems are Gigascope [14] (network monitoring), Hancock [15] (analysis of telecommunication calls), NiagaraCQ [16] (large number of queries on web contents), Statstream [17] (correlations between a large number of sensors) and Traderbot [18] (analysis of stock market).

A complete survey of DSMSs which was up to date in 2004 can be found in [19]. It also includes work on sensor networks which is related to data stream management.

4. Data stream mining

Data stream mining can be defined as *'applying data mining algorithms to one or several data streams'*, with the same constraints as DSMSs: (1) one-pass, (2) limited memory and CPU. From a user point of view, the data mining task can apply either to the whole stream since its beginning or to a part of it defined by a window. Depending on the type of window, the algorithms are different:

- Whole stream: algorithms have to be incremental. This is the case for instance for neural network algorithms which are incremental in nature. Some non-incremental algorithms can be modified to do so. This has been done for instance for decision trees (see below).

- Sliding window: algorithms have to be incremental and have the ability to forget the past of the stream though the corresponding elements are not accessible anymore. This is the case for additive methods like PCA (see below).
- Any past portion of the stream not decided in advance: algorithms have to be incremental and some summaries of the past must be kept in limited memory. Summaries of data streams can be done by several means; a solution based on micro-clustering is described below.

We do not cover here all stream mining algorithms but only illustrate these three cases by one example of each.

4.1. Building decision trees on data streams

Some early work has been done by Domingos and Hulten to build decision trees incrementally from data streams (see [20]). The idea is that it is not necessary to wait for all examples to decide of a split in the tree. A statistical test is performed to decide when a split can be done. The statistical test can be done by keeping track of appropriate statistics which can be computed incrementally in bounded space.

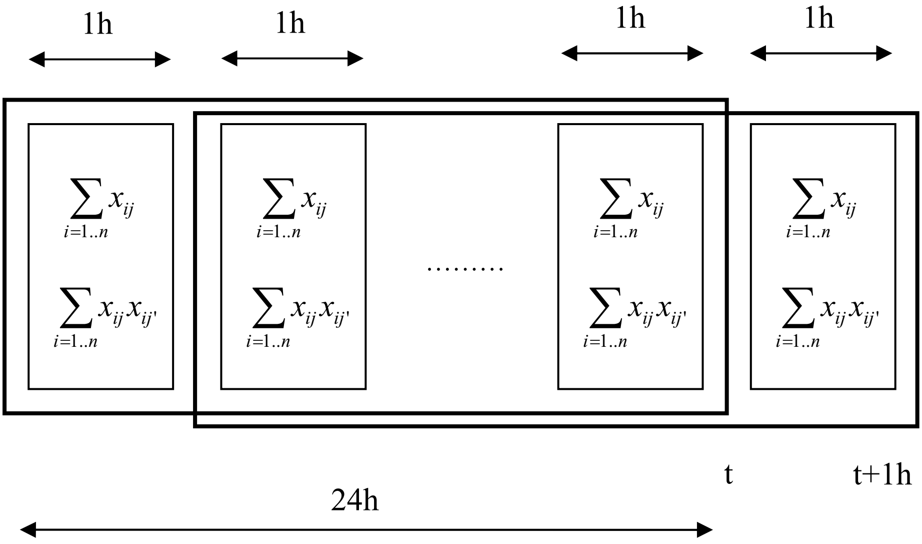

1h
1h
1h
1h
$\sum_{i=1..n} x_{ij}$
$\sum_{i=1..n} x_{ij}x_{ij'}$
$\sum_{i=1..n} x_{ij}$
$\sum_{i=1..n} x_{ij}x_{ij'}$
....
$\sum_{i=1..n} x_{ij}$
$\sum_{i=1..n} x_{ij}x_{ij'}$
$\sum_{i=1..n} x_{ij}$
$\sum_{i=1..n} x_{ij}x_{ij'}$
t
t+1h
24h

Figure 1. PCA on a sliding window

4.2. Performing PCA incrementally on a sliding window

Principal Component Analysis (PCA) is performed by singular value decomposition (*svd*) of the covariance (correlation) matrix between variables. It is easy to prove that the covariance matrix can be computed incrementally. Moreover, if the sliding window is for instance 24 hours and the refreshment rate is 1 hour, it is possible to maintain 25 covariance matrices as shown in Figure 1. At the end of each hour period, the covariance matrix can be computed from the last 24 ones and the *svd* is performed. This solution is applicable to any additive or pseudo-additive data mining method.

4.3. Performing clustering on any past portion of the stream

Several clustering algorithms have been developed to be applicable to data streams (see for instance [21]). The *Clustream* algorithm (see [22]) allows to perform clustering on any past portion of a stream where all fields are numerical. This algorithm works in two steps: (1) a clustering is performed on-line to build and maintain a large number of micro-clusters whose contents follow the contents of the stream, (2) a clustering of the micro-clusters is done off-line on demand to produce the final clustering.

The key concept of this approach is that micro-clusters are defined in such a manner that it is possible to 'substract' micro-clusters at date T1 from micro-clusters at date T2 to obtain micro-clusters describing the stream between T1 and T2. This is done by storing for each micro-cluster additive statistics in a structure called *Cluster Feature Vector CFV* (sum of values and sum of squares for every field, including the timestamp field). Snapshots of the micro-clusters are taken regularly and compacted logarithmically as micro-clusters get older: this approach is called *tilted time windowing* and ensures that the summary of the whole stream is stored with bounded memory (see Figure 2). The drawback is that there is less detail for older data than for recent data but this is often acceptable in applications.

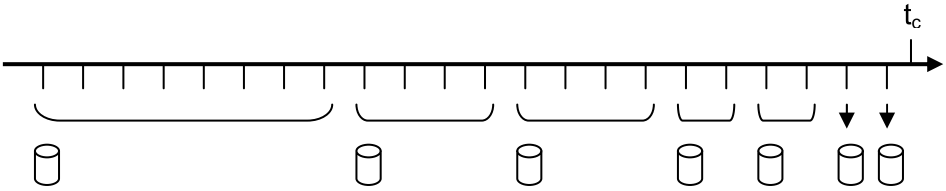


Figure 2. *Tilted time* structure: the size of storage decreases logarithmically when data gets older

5. Synopses structures

A strong constraint both in querying and mining data streams is to process data 'on the fly', facing variations in input streams. For many applications, it is nevertheless necessary to keep track of information related to a maximum number of elements of the stream but in bounded space. As seen in the previous sections, a common way to do so is to restrict the scope of the query (or the mining task) by windowing. In many applications, this solution is not efficient enough and much work has been done to summarize the contents of data streams in order to produce an approximate but controlled result.

Synopses structures can be used both for answering queries and applying data mining algorithms to streams. Many synopses structures have been studied. We present here a few examples of them.

We can distinguish between temporal and memory management approaches. The most famous approach to temporal management is the concept of *tilted time windows* (introduced in [23]) which compacts historical data by forgetting details in a logarithmic way with time, thus keeping a summary of the past with bounded memory. This approach of temporal management can be combined to any memory management approach. For instance, Clustream combines tilted time windows and micro-clustering.

As for memory management, many approaches have been studied, mainly: random samples, histograms and quantiles, micro-clusters and sketches. Since Clustream is an example of synopsis in the form of micro-clusters, we only develop below two approaches: random sampling and sketches.

Summarizing by random sampling

On-line random sampling from data streams is a problem because the size of the sample has to be bounded and the size of the data set from which elements are selected is not known when sampling begins, i.e. at the beginning of the stream. Some algorithms exist to overcome this problem: the *reservoir* algorithm (see [24]) maintains a fixed size uniform random sample from the beginning of the stream to current time. Some elements are selected randomly from the stream with a decreasing probability and replace randomly elements already in the sample. The decreasing probability ensures that the sample is uniform over the period.

Some extensions and new approaches have been developed to extract random samples from a sliding window defined on the stream (see [25]). In this case, the problem is that elements of the sample have to be removed and replaced by new ones when they expire from the window, under the constraint of still keeping uniform the sample.

A combination of random sampling and tilted time windowing is described in the paper of Baptiste Csernel in this book.

Summarizing with sketches

Sketches are data structures dedicated to a specialized task which can be updated incrementally to maintain in small space summarized information. They are often based on the use of hash functions which project randomly stream data on the small structure. Approximate answers to queries can be obtained from sketches with probabilistic bounds on errors. Since data streams produce data with a high rate, the probabilistic bounds are quite tight. Popular sketches are the Flajolet sketch which counts the number of distinct objects appearing in a stream (see [26]) and the count sketch algorithm (see [27]) which maintains the *k* most frequent items in a stream. These two approaches provide the result with storage much smaller than the number of distinct items appearing in the stream.

Conclusion

This paper has introduced the main concepts of data stream management and mining. The two main directions are the development of Data Stream Management Systems (DSMS) which enable to query one or several streams, and the development of data stream algorithms which enable to mine data streams without storing them entirely. Synopses are basic tools which enable to process streams using a smaller storage space: they can be used both in query and mining tasks.

As for perspectives on DSMSs, there is some current work on event processing systems which provide a more convenient querying language for streams describing events (see for instance the Cayuga [28] and SASE [29] projects). The first commercial DSMSs appear on the market and are used mainly in financial applications.

As for perspectives on data stream mining, one challenge is to build summaries of the whole stream in order to be able to apply data mining tasks to any past portion of the stream. Another challenge is to apply data mining algorithms to several streams, without joining them but keeping a summary of each stream which preserves the relationship between the streams. The paper of Baptiste Csernel in this book describes an approach to do so.

References

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom, *Models and Issues in data stream systems*, PODS’2002, 2002.

[2] M. Garofalakis, J. Gehrke, R. Rastogi, *Querying and Mining Data Streams: You Only Get One Look. A tutorial*, Tutorial SIGMOD'02, Juin 2002.

[3] L. Golab, M.T. Özsu, *Issues in Data Stream Management*, Canada. SIGMOD Record, Vol. 32, No. 2, June 2003.

[4] S. Muthukrishnan, *Data streams: algorithms and applications*, In Foundations and Trends in Theoretical Computer Science, Volume 1, Issue 2, August 2005.

[5] C.C. Aggarwal, *Data streams: models and algorithms*, Springer, 2007.

[6] T. Abdessalem, R. Chiky, G. Hébrail, J.L. Vitti, *Using Data Stream Management Systems to analyze Electric Power Consumption Data*, International Workshop on Data Stream Analysis (WDSA),. Caserta (Italie), March 2007.

[7] Amalgamated Insight/Truviso, http://www.truviso.com

[8] Streambase software, http://www.streambase.com

[9] A. Arasu, M. Cherniack, E. Galvez, D. Maier, A.S. Maskey, E. Ryvkina, M. Stonebraker, R. Tibbetts, *Linear Road: A Stream Data Management Benchmark*, Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004. http://www.cs.brandeis.edu/~linearroad/

[10] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava, J. Widom. *STREAM: The Stanford Data STREAM Management System*, Department of Computer Science, Stanford University. Mars 2004. Available at: http://www-db.stanford.edu/stream

[11] S. Chandrasekaran, O. Cooper, A. Deshpande, M.J. Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, M. Shah, *TelegraphCQ: Continuous Dataflow Processing for an Uncertain World*, CIDR 2003. http://telegraph.cs.berkeley.edu/telegraphcq/v2.1/

[12] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik, *Aurora: A New Model and Architecture for Data Stream Management*, In VLDB Journal (12)2: 120-139, August 2003.

[13] B. Babcock, M. Datar, R. Motwani, *Load Shedding for Aggregation Queries over Data Streams*, 2004. Available at:http://www-db.stanford.edu/stream

[14] T. Johnson, C.D. Cranor, O. Spatscheck, *Gigascope: a Stream Database for Network Application*, ACM SIGMOD International Conference on Management of Data, June 9-12, 2003.

[15] C. Cortes, K. Fisher, D. Pregibon, A. Rogers, *Hancock: a language for extracting signatures from data streams*, 2000 KDD Conference, pages 9–17, 2000.

[16] J. Chen, D. J. DeWitt, F. Tian, Y. Wang, *NiagaraCQ: A Scalable Continuous Query System for Internet Databases*, SIGMOD Conference 2000: 379-390.

[17] Y. Zhu, D. Shasha, *StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time,* VLDB, Hong Kong, China, August, 2002.

[18] Traderbot home page, http://www.traderbot.com (closed since July 2007).

[19] V. Goebel, T. Plagemann, *Data Stream Management Systems - Applications, Concepts, and Systems*, Tutorial MIPS'2004, 2004.

[20] P. Domingos, G. Hulten, *Mining High-Speed Data Streams*, In Proceedings of the 6th ACM SIGKDD conference, Boston, USA, pages 71-80, 2000.

[21] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, *Clustering data streams*, In IEEE Symposium on Foundations of Computer Science. IEEE, 2000.

[22] C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu, *A framework for clustering evolving data streams*, In Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003.

[23] Y. Chen, G. Dong, J. Han, B. Wah, J. Wang, *Multidimensional regression analysis of time-series data streams*, VLDB Conference, 2002.

[24] J. Vitter, *Random sampling with a reservoir*, ACM Trans. Math. Softw., 11(1) :37–57, 1985.

[25] B. Babcock, M. Datar, R. Motwani, *Sampling from a moving window over streaming data*, In Proceedings of the thirteenth annual ACM-SIAM SODA, 2002.

[26] P. Flajolet, *Probabilistic Counting Algorithms for Data Base Applications*, In Journal of Computer and System Sciences, Volume 32, Issue 2, page 182-209, Sept.1985

[27] M. Charikar, K. Chen, M. Farach-Colton, *Finding frequent items in data streams*, Theor. Comput. Sci., 312(1) :3–15, 2004.

[28] L. Brenna, A. Demers, J. Gehrke, M. Hong, J. Ossher, B. Panda, M. Riedewald, M. Thatte, W. White, *Cayuga: A High-Performance Event Processing Engine*, ACM-SIGMOD 2007.

[29] D. Gyllstrom, E. Wu, H.J. Chae, Y. Diao, P. Stahlberg, G. Anderson. *SASE: Complex Event Processing over Streams*. Demo in CIDR 2007, Asilomar, CA, January 2007.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-103

Modelling and Analysing Systems of Agents by Agent-aware Transition Systems[1]

Marek BEDNARCZYK [a,2], Luca BERNARDINELLO [b], Wiesław PAWŁOWSKI [a] and Lucia POMELLO [b]

[a] *Institute of Computer Science, Polish Academy of Sciences, Gdańsk, Poland, and Institute of Informatics, the University of Gdańsk, Poland*
[b] *Dipartimento di informatica, sistemistica e comunicazione, Università degli studi di Milano–Bicocca, Italia*

Abstract. We propose a method to specify, in a modular way, complex systems formed by interacting agents. The method is based on the notion of *view*, that is a partial representation of the system, reflecting one of its specific aspects. By composing the different views, we get the overall system, described as a special kind of transition system. By means of a suitable logical language, we can express interesting properties of the system; model-checking techniques can then be used to assess their validity. Views can be specified using different languages or notations, provided they can be translated in so-called agent aware transition systems. The method is explained with the help of a simple, but non trivial example.

Keywords. Transition systems, agents, views, model checking, modularity

Introduction

Large scale data mining is usually performed on data that are distributed in space and heterogeneous in formats. It is therefore natural to think of setting it in a framework allowing for parallel and coordinated action of several agents (see [1]).

The research in this field is active, as witnessed, for instance, by [2], where the reader can find contributions dealing with the specific issues raised by the use of agents in data mining.

In this contribution, we focus on a more general matter, and devise a method for modelling potentially complex systems, formed by "agents" who can interact to perform their tasks. The term *agent* is used here in a very generic way, and simply denotes a component of a system which bears an identity and exists throughout the time span of the system.

Modelling can be used for two reasons: either for designing a new system, or for describing an existing system that we want to analyze. In both cases, we need rigorous

[1] Partially supported by PAN and CNR, project CATNET, and by MUR.

[2] Corresponding Author: Marek Bednarczyk, IPI-PAN Gdańsk; E-mail: m.bednarczyk@ipipan.gda.pl.

techniques allowing us to examine the model, in order to derive information on the behaviour and on the structural and dynamical properties of the real (or to be built) system.

That need requires the use of formal techniques, so that the model be amenable to automatic analysis by means of efficient algorithms. In the present context, our most general reference model is given by *transition systems*, that is structures defined by a set of states and by a set of possible state transitions. Each transition is labelled by the name of an “action”, or event, whose occurrence triggers the change of state. The same action can obviously occur in different states, but generally is constrained by specific preconditions that must be satisfied in a state.

Such transition systems lend themselves to a kind of formal analysis, namely *model-checking*, on which we will return later. Suffice it to say, by now, that such analysis requires the specification of properties to be checked in a logical language, usually based on temporal logics. Specific algorithms examine the state space of the system and check whether the given property is satisfied or not. These algorithms are now very efficient, and allow the analysis of very large state spaces (for an introduction to the subject, see [3]).

Although highly efficient algorithms for model-checking are available, using transition systems as a way to specify or describe real systems runs across a fundamental obstacle, particularly for systems composed of a large number of interacting components, which run in parallel; the problem consists in the so-called *state explosion*: the overall, or global, state of such a system is given, intuitively, by the combination of the local states of all components. The number of possible combinations of local states grows exponentially with the number of components, and can soon become unmanageable.

We are thus led to look for modelling techniques which allow to develop a specification in a modular way. A typical solution would consist in defining a language for specifying single components, and then rules of interaction. This solution has been used, for instance, in association with *process algebras*, languages in which sequential processes (components) can be specified by using different control structures (like sequence, choice, repetition), and then composed by a parallel composition operator (see, for instance, [4], [5], and [6]). A different basic strategy underlies *Petri nets*, where a system is directly defined in terms of a set of local states and a set of local state transitions. Here, components can be recovered as higher level structures within a net (for a wide review on the theory and applications of Petri nets, see [7]).

In this paper, we will follow a different path, based on the idea of *views*. By *view* we mean, intuitively, a sort of projection of the system to be described, as if seen along one of its many dimensions. Such a dimension may coincide with the observation of the system by one of its agents, or components, but is a more general concept. One view might, for instance, express some constraints, like legal or physical constraints, which limit the behaviour of agents; another one might represent spatial relations among agents, which govern their possibility to interact, and the way in which agents can move in space. Still other views can be used to express organizational features of the system.

With this approach, the designers can apply, while developing their models, principles of separation of concerns which help them in managing complexity.

In general, a view is a partial representation of a system. Each view is concerned with only some of the agents and with only some of the actions. Of course, the same agent and the same action can be observed in several views.

A view can be studied in isolation; however, the potential behaviour of a view is in general constrained when the view is composed with other views. Hence, only safety properties (namely, properties expressed by statements of the kind "no *bad* state will be reached") are preserved after composition. For liveness properties (expressed by statements of the kind "something good will eventually happen") we need to design views respecting stronger conditions.

Combining all the views, we get the overall system, whose observable behaviour results from superimposing the constraints coming from each view.

This kind of modular representation derives from ideas developed first within the theory of basic Petri nets [8], and later recast in a new formal setting, where agents and their changing hierarchical relations are explicitly represented [9]. The resulting model, called *hypernet*, can be seen as a sort of compact notation for Petri nets. A rather drastic generalization, or rather abstraction, led us to the idea of *agent aware transition system* [10], which is the formal tool described in this paper.

The main aspects of the framework we propose will be introduced by means of an example. The example is informally described in the next section, together with a short discussion on the kind of formal analysis that we would like to perform on the corresponding model.

Section 2 takes the step from the verbal description to a formal representation. The notions of agent aware transition system, view, and so on, will be gradually introduced with some comments on their applicability.

In Section 2.6 we discuss the kind of structural and dynamical properties that we can express in a suitable logical language, and suggest how to check whether those properties are satisfied.

Finally, in Section 3, we summarize our approach, and try to briefly assess its merits, drawbacks, and limitations.

1. An Example: Dynamic Coalitions

The example chosen to illustrate our ideas is, for obvious reasons, quite artificial, and much simplified with respect to a realistic setting. However, we hope that even such a simple model can convey the main ideas that underlie our approach. The example shows the main features of the formal framework: the components of the system (abstractly called agents in the following) are classified in several types. Mutual relationships among the agents can change in time as a result of the execution of actions. Some actions involve a number of agents; we refer to such a situation by talking of a *synchronization* of the involved agents. However, agents can also interact in more indirect ways, like exchanging messages.

The system we want to model is formed by a set of *players* who communicate within dynamically forming and changing *coalitions*. Here, we do not attach a specific meaning to the word *coalition*; the reader might think, for instance, of groups of interest, but other interpretations are possible. What matters is the constraint that an actual exchange of information can happen only between two players which currently belong to the same coalition.

A player can belong to several coalitions, can promote new coalitions, and can leave a coalition. Joining a coalition is only possible after receiving an *invitation* from a player which is already a member of that coalition.

Within such a general description, we can imagine several properties that a designer might want to check, or to enforce. For instance, we might be interested in proving that two given players will never belong to the same coalition, or to show an admissible sequence of events that will violate such a property; or, we might want to prove that, given three distinct players a, b, c, whenever a and b belong to the same coalition γ, c does not belong to γ. Later we will suggest how to express such properties, and how to check them.

2. Dynamic Coalitions: Formal Setting

In this section, we translate the plain English description of the system into a formal definition, and introduce, along the way, the needed notions.

2.1. Agent aware transition systems

As a first step along the way, we define the general concept of agent aware transition system. In the end, we want to be able to specify our system as such an object, possibly through a more expressive specification language.

Definition 1 *An* agent aware transition system *(AAS), is a tuple $\mathcal{S} = \langle A, T, S, \delta, s_0 \rangle$, where A is a finite set of* agents*, S is a finite set of* states*, T is a finite set of* actions*, δ is the transition function, and $s_0 \in S$ is the initial state. A single transition is specified by an action, the set of involved agents, the starting state, and the final state. More formally, the transition function is a partial map*

$$\delta : T \times 2^A \times S \longrightarrow_* S$$

where 2^A denotes the set of subsets of A.

The map δ is defined as partial to reflect the fact that in certain states, a given action cannot be performed by a given set of agents.

With respect to classical transition systems, an AAS introduces agents, and the specification of the set of agents which take part in a transition. The same action can thus be performed by different groups of agents.

In the formal model that we are going to develop for the system of players and coalitions, each view will be specified as an AAS. In the next section we will define views, but first we must answer a basic question. How can we build up the entire system from the collection of views? As suggested in the introduction, the different views must be superimposed. Formally, this superimposition is defined in a way which resembles the usual synchronous product for transition systems, where an action can occur in the composite system only if all components which have that action in their alphabet are ready to execute it. This mechanism gives an abstract form of synchronization, or interaction, among components.

The corresponding operations can be defined as a method to compose n AASs, irrespective of their interpretation as views of the same system.

Here, we will content ourselves with an intuitive description of the operation. The reader interested in formal definitions is invited to look at [11].

Assume then to have several AASs, $\mathcal{S}_i$, with i varying over a set I. We want to define an AAS, call it $\mathcal{S}$, as the composition of the $\mathcal{S}_i$. The states of $\mathcal{S}$ are all combinations of states of the components. On the other hand, actions and transitions are defined as the union of actions and transitions, respectively, of the components.

Two, or more, transitions, taken from different components, are superimposed, or synchronized, when they are consistent. Consistency means that they can be seen as a sort of projections of a more general transition. By *projection* of a transition t on a component $\mathcal{S}_i$, we mean a transition with the same label, taken from T, and such that the set of agents involved is exactly the intersection of the set of agents involved in t and the set of agents belonging to $\mathcal{S}_i$.

The underlying idea can be intuitively explained as follows. A state transition is, in general, a complex entity; several agents can participate in a transition. We can observe the transition from different standpoints, corresponding to the different views in which the system is articulated. When looking at the transition from a standpoint corresponding to view i, we can only see agents belonging to that view; so in view i we must find the corresponding transition, labelled by the appropriate set of agents.

2.2. Agents, Actions, and Views

From the description of the system, we can derive the entities that must be explicitly represented in the model. They can be classified in three categories: *players*, representing people who gather in groups of interest and exchange information, *coalitions*, representing the groups themselves, which have players as members, and *messages*, which are used by players in order to invite other people to join a coalition; notice that these messages do not represent information exchanged within a coalition: in our simple example, we do not deal with the actual exchange of information, which will be represented by a generic action called *talk*.

Players, coalitions, and messages are the agents of the AAS that we are going to define. Each view, defined later will deal with a subset of the agents.

We can now identify the views through which we look at the behaviour of agents. We define three of them.

The first view is concerned with the knowledge that players have of coalitions. Remember that a player can join a coalition only after receiving an invitation from another player; we will assume that each player has knowledge of a subset of coalitions, and can enlarge its knowledge by receiving an invitation (receiving an invitation is a prerequisite for joining the coalition, but a player can ignore an invitation if not interested in joining that coalition). We also provide for an action by which a player can forget a coalition. After forgetting a coalition, a player can not join it, unless she receives a new invitation.

The second view defines the rules by which two players can talk to each other, namely that they belong together to some coalition. This view should keep track of membership in coalitions.

Finally the third view governs the "mechanics" of invitations. It describes the correct sequences of actions involving messages (for instance, that a given message can be received only after being sent) and associates to a pending message its content, that is the coalition it refers to.

Views will now be defined more precisely, in a sort of operational style. For each of them, we will define the set of states, the set of agents, and the set of actions; then we will give the conditions that allow a transition to occur at a given state, and the effect of its occurrence.

In the following, P denotes the set of all players, C the set of all coalitions, M the set of all messages. These sets are fixed from the beginning: in AASs, we cannot create or delete agents. We will comment on this restrictions in the conclusion of the paper.

2.3. View 1: Knowledge of Coalitions

This view represents one kind of relation between players and coalitions, expressed by the statement *player* p *knows about coalition* c.

With respect to the general form of an AAS, we have

$$\mathcal{S}_1 = \langle A_1, T_1, S_1, \delta_1, s_{01} \rangle$$

where $A_1 = C \cup P$. States of this view associate to each player the set of coalitions she knows (this is not the same as the set of coalitions that she is a member of); formally, states are functions from the set of players to subsets of coalitions:

$$S_1 = \{K \colon P \longrightarrow 2^C\}$$

The actions relevant to this view are those actions which depend on the knowledge of a player, or which affect that knowledge:

$$T_1 = \{\mathit{join}, \mathit{send}, \mathit{receive}, \mathit{start}, \mathit{forget}\}$$

We have now to define the transition function. To this aim, we list a set of generic clauses that implicitly define δ_i. To keep the notation compact, we adopt the following general convention: let $f \colon X \to Y$ be a map; then, by $f\,[z/x]$ we denote a new map from X to Y which coincides with f for all elements in the domain except for x, where it takes the value z.

There are five rules for this view. The first rule states that a player can start a new coalition only if she has knowledge of it (we can say: if she knows its name).

Let K be a state of this view.

- $K \overset{\mathit{start}\{c,p\}}{\longrightarrow} K$ if and only if $c \in K(p)$

The second rule allows a player to join a coalition provided she knows about it already.

- $K \overset{\mathit{join}\{c,p\}}{\longrightarrow} K$ if and only if $c \in K(p)$

The third rule allows a player who knows about a coalition to send an invitation for it.

- $K \overset{\mathit{send}\{c,p\}}{\longrightarrow} K$ if and only if $c \in K(p)$

Notice that the first three rules do not change the state of this view (the same actions will show up in other views also).

The fourth rule states that a player receiving an invitation can enlarge her knowledge. This does not exclude that a player receives an invitation for a coalition she already knows.

- $K \overset{receive\{c,p\}}{\longrightarrow} K\left[K(p) \cup \{c\}/p\right]$

The state of the view changes after performing an action *receive*, reflecting the fact that the receiver now knows about coalition c.

The final rule for this view governs the occurrences of action *forget*. By this action, an agent voluntarily forgets about a coalition.

- $K \overset{forget\{c,p\}}{\longrightarrow} K\left[K(p) \setminus \{c\}/p\right]$

2.4. View 2: Coalition Membership

This view is concerned with the relation of membership between players and coalitions. The agents involved in this view are, like in the first view, players and coalitions:

$$A_2 = P \cup C$$

but the states are different. A state of this view must keep track of actual membership. It is then natural to define states as maps from C to subsets of P:

$$S_2 = \{s : C \longrightarrow 2^P\}$$

with the intended meaning that $p \in s(c)$ if player p is a member of coalition c in state s.

Relevant actions for this view include actions which change membership, and also the generic action, here called *talk*, representing an exchange of information between two players.

$$T_2 = \{start, join, drop, talk\}$$

Rules:

A player can start a coalition only if that coalition has no members. After performing the action, the promoter is the only member of the coalition (she will be able to invite other players later).

- $s \overset{start\{c,p\}}{\longrightarrow} s'$ iff $s(c) = \emptyset$ and $s' = s\left[\{p\}/c\right]$
- $s \overset{join\{c,p\}}{\longrightarrow} s'$ if and only if $p \notin s(c)$, $s(c) \neq \emptyset$ and $s' = s\left[s(c) \cup \{p\}/c\right]$
- $s \overset{drop\{c,p\}}{\longrightarrow} s'$ if and only if $p \in s(c)$ and $s' = s\left[s(c) \setminus \{p\}/c\right]$
- $s \overset{talk\{p,p',c\}}{\longrightarrow} s$ if and only if $p, p' \in s(c)$

2.5. View 3: Mechanics of Invitations

This view deals with invitations. A player can invite someone to join a coalition by sending a message bearing the name of the coalition. In defining this view, we must face a constraint given by the formal framework in which we operate. In order to apply the analysis techniques briefly described in a later section, the model of the system must be finite. Since messages are agents in our model, we cannot create and destroy them freely. Hence, we choose to assume that there are enough messages in the initial states, and reuse them as needed. A "quiescent" message is like an empty box. On sending a message, the sender associates it with a coalition. On receiving the message, the addressee removes its content, so that the message can be reused.

A typical state for this view is decribed by a map from messages to coalitions, associating to each message its content. Since a message can be empty, the map is only partial. By checking if the map is defined for a given message, we can also decide whether a message is traveling from a player to another.

To keep the model simple, we do not explicitly represent the addressee of a message. This means that any player can catch a pending message. Several properties of a system of this kind are actually independent of that information.

The agents involved in this view are messages and their contents, namely coalitions:

$$A_3 = M \cup C$$

States are partial maps from messages to coalitions:

$$S_3 = \{s \colon M \longrightarrow_* C\}$$

Relevant actions include sending and receiving a message. Notice that in this view there is no way to know which player sends or receives a message, since this is irrelevant in this context. Such information is available in the first view. After composing views, a send action will be associated to a message, a coalition, and a player.

$$T_3 = \{\mathit{send}, \mathit{receive}\}$$

The rule concerning *send* requires that a message is empty before it can be sent. In performing *send*, it is filled with the name of a coalition.

- $s \overset{\mathit{send}\{c,m\}}{\longrightarrow} s'$ if and only if $s(m)$ is undefined and $s' = s\left[c/m\right]$

The rule for *receive* is symmmetrical to the former: a message can be received only if it has been sent, that is if it has a content. The act of receiving deletes its content. The effect of receiving a message on the knowledge of the receiver is represented in view 1.

- $s \overset{\mathit{receive}\{c,m\}}{\longrightarrow} s'$ if and only if $s(m) = c$ and $s' = s\left[\perp /m\right]$

One last ingredient is needed to fully specify the views composing our system: the initial state. For each view, we must specify the initial situation. In particular, the initial state should “initialize” the knowledge of each player, so that each coalition is known by at least one player.

Building the complete description of the system as a single AAS can now be done automatically, by applying the synchronization operation informally explained above.

2.6. Analysis

In this section we hint at the kind of properties of a system that one would like to check on the corresponding model. Properties are expressed in a logical language endowed with temporal operators.

The logical language we propose is based on a fixed set of symbols, denoting predicates on agents, and relations among agents. One obvious relation for our example could, for instance, express the fact that a given player is a member of a given coalition. A simple predicate can assert that a given coalition is empty. Formulae built on the fixed set of symbols, the set of names of agents, and on variables, can be evaluated at arbitrary states, after fixing a valuation function which associates each variable with an agent. The validity of a formula can obviously change from state to state.

The temporal operators allow us to specify a rich set of formulae of the kind "for each admissible sequence of actions from the initial state, eventually the formula ϕ will be valid", or "there exists an admissible sequence of actions from the initial state such that formula ϕ is valid until ψ becomes valid, and so on.

For instance, assume that in the example there are three players, p_1, p_2, and p_3, and we want to check, maybe for security reasons, that in no state they are members of the same coalition c. Then we can write the formulae

$$\alpha = \mathrm{in}(p_1, c) \wedge \mathrm{in}(p_2, c) \Rightarrow \neg \mathrm{in}(p_3, c)$$

$$\beta = \mathrm{in}(p_2, c) \wedge \mathrm{in}(p_3, c) \Rightarrow \neg \mathrm{in}(p_1, c)$$

$$\gamma = \mathrm{in}(p_1, c) \wedge \mathrm{in}(p_3, c) \Rightarrow \neg \mathrm{in}(p_2, c)$$

where *in* is a symbol denoting the membership relation between a player and a coalition.

The property we want to check can then be expressed as follows, where s_0 is the initial state of the system, and □ denotes the temporal operator "always".

$$s_0 \models \Box(\alpha \wedge \beta \wedge \gamma)$$

3. Conclusion

With the help of a simple example, we have outlined the main ideas of a method for designing formal models of complex systems. The method is based on the notion of *agent aware transition system*, which can be seen as a generalization of the standard notion of transition system. A distinguishing feature of AASs is its explicit treatment of agents as entities in the model; the actions that make such a system evolve are always referred to the set of agents involved.

In order to apply usual model-checking techniques, the models we build are bound to finite sets of states. Consequently, we do not allow creation of agents. This can be a strong limitation in the expressive power of the model in some fields of application. The extension to infinite state spaces will be explored, in association to model-checking algorithms for such classes of systems.

A central feature of or approach is the idea of view. While this is certainly not a new idea in general, the presence of agents gives it some special features, that we think may be useful while developing the specification of a distributed system.

As a generalization of labelled transition systems, AASs are a very general notion. When designing a real systems, a designer can work on more expressive or compact notations, like hypernets, Petri nets, or process calculi, provided there is an automatic way to translate them into AASs.

References

[1] H. Kargupta, I. Hamzaoglu, and B. Stafford. Scalable, distributed data mining using an agent based architecture. In *Int. Conf. on Knowledge Discovery and Data Mining, pp. 211-214, August 1997*, 1997.

[2] Vladimir Gorodetsky, Chengqi Zhang, Victor A. Skormin, and Longbing Cao, editors. *Autonomous Intelligent Systems: Multi-Agents and Data Mining, Second International Workshop, AIS-ADM 2007, St. Petersburg, Russia, June 3-5, 2007, Proceedings*, volume 4476 of *Lecture Notes in Computer Science*. Springer, 2007.

[3] Edmund Clarke, Orna Grumberg, and Doron Peled. *Model checking*. MIT Press, 1999.

[4] C.A.R. Hoare. *Communicating sequential processes*. Prentice-Hall, 1985.

[5] Robin Milner. *Communicating systems and the π-calculus*. Cambridge University Press, 1999.

[6] Wan Fokkink. *Introduction to process algebra*. Texts in theoretical computer science. Springer, 1999.

[7] Wolfgang Reisig and Grzegorz Rozenberg, editors. *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*, volume 1491 of *LNCS*. Springer, 1998.

[8] Marek A. Bednarczyk, Luca Bernardinello, Benoit Caillaud, Wiesław Pawłowski, and Lucia Pomello. Modular system development with pullbacks. In Will van der Aalst and Eike Best, editors, *24th International Conference on Applications and Theory of Petri Nets, Eindhoven, The Netherlands, June 2003*, volume 2679 of *LNCS*, pages 140–160. Springer-Verlag, 2003.

[9] Marek A. Bednarczyk, Luca Bernardinello, Wiesław Pawłowski, and Lucia Pomello. Modelling mobility with Petri hypernets. In José Luiz Fiadeiro, Peter D. Mosses, and Fernando Orejas, editors, *Recent Trends in Algebraic Development Techniques. 17th International Workshop WADT 2004, Barcelona, Spain, March 27-30, 2004. Revised Selected Papers*, volume 3423 of *LNCS*, pages 28–44. Springer-Verlag, 2005.

[10] Marek A. Bednarczyk, Wojciech Jamroga, and Wiesław Pawłowski. Expressing and verifying temporal and structural properties of mobile agents. In Ludwik Czaja, editor, *Proceedings of the Concurrency, Specification and Programming Workshop CS&P'05*, pages 57–68, 2005.

[11] Marek A. Bednarczyk, Luca Bernardinello, Tomasz Borzyszkowski, Wiesław Pawłowski, and Lucia Pomello. A multi-facet approach to dynamic agent systems. In L. Czaja, editor, *Proceedings of the International Workshop on Concurrency, Specification and Programming (CSP07), Vol. 1, Łagów, Poland, 26-30 September 2007*, pages 33–47, 2007.

Search

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-115

The "Real World" Web Search Problem: Bridging The Gap Between Academic and Commercial Understanding of Issues and Methods

Eric GLOVER
SearchMe Inc.
800 W. El Camino Real, Suite 100
Mountain View, CA 94040

Abstract. In the area of web search, there is a disconnect between some of the research in the field and what is actually done within a large-scale commercial search engine. Everything from measuring the wrong things to the objectives of the research, have the potential to produce work that although academically interesting, has little commercial value. This paper attempts to reduce this gap by summarizing my experience as both a researcher and an engineer at commercial search engines.

This paper focuses on the theoretical components of a commercial search engine and then describes several of the real-world issues and challenges faced by commercial search engines. Next, it explores briefly the issue of "relevance" and some challenges specific to studying and improving relevance in a commercial search engine.

Keywords. web search engine architecture, web search relevance

Introduction

It is easy to view a search engine using an IR methodology, as a system which takes in a query and returns an ordered list of the ten most *relevant* results ("ten blue links"). However, a modern web search engine is much more than "ten blue links", and the nature of a Web Search Engine (WSE) is very different from classical Information Retrieval (IR) systems; the way a modern search engine should be evaluated is not based simply on what fraction of the results returned are relevant to some binary notion of relevance.

To reduce the gap of understanding between a WSE and researchers in the area, this paper summarizes my personal experiences in this field as both an academic and an engineer working for commercial search engines. This paper summarizes the challenges currently faced by commercial search engines, as well as common mistakes made by researchers in the area of web search related problems. Section 1 delves into a simple model of the components of a WSE. Section 2 will explore some limitations of the simplistic models of the components, and some significant challenges faced by a WSE. Section 3 is dedicated to *relevance*, how it is evaluated on classical IR systems and which unique properties of the web make ranking documents difficult. Section 4 will very briefly discuss some approaches search engines use to address some of the problems, and some academic work which could be promising.

The Modern Search Engine - more than ten blue links

The ultimate goal of a commercial WSE is to make money. To accomplish this, a WSE must offer a more compelling value proposition than their competitors. Modern WSEs aim to provide a variety of information or answer a variety of information needs, than simply returning "ten blue links". A modern search engine is very likely to be able to answer the current value of ten Euros in US dollars, or the current temperature in Milan Italy, or if your flight to Miami is on-time. In fact, many people may not be aware that if you enter a UPS or FedEx package number in several commercial search engines, it is recognized as such and relevant information is offered. Likewise, if you enter the name of a company, you are likely to find their official site as the top result. In addition to textual queries where web pages are the 'answer', most WSEs also offer image or video search.

Over the past ten years, there have been significant advances in commercial WSEs, not least of which is the User Interface (UI). For example, in 2006, Ask.com launched their new Ask-3D which combines results for many different information sources, and media types. Yahoo! and Google both offer differentiated UIs for popular queries including links to editorial resources, popular links under an official homepage, images, and suggestions for alternate queries. SearchMe Inc., announced in March 2008, a visual interface, with the ability to choose from a dynamically generated set of "categories" to help narrow the focus of your search - while viewing the result pages as a whole, not limited to the immediate query-term context.

A WSE is very different from the IR systems of the late 1980s. Unlike a traditional IR system for library books, or even corporate document retrieval, the scale of a WSE is much larger. Not only is the indexable web larger than 25 Billion pages, but large WSEs must manage tens of thousands of machines to effectively process the web and handle the query load. This does not include the likely hundreds of Billions of URLs Google probably "knows about", but does not index. Unlike other IR systems, a WSE must deal with active targeted attacks against their algorithms and systems - including creation of content designed to cause problems and automated attacks designed to tie up their query-engines with junk queries. This does not include the "attacks" against advertising products and services, such as Google's AdSense or AdWords.

1. Components of a Web Search Engine

Although a modern WSE is complex, the basic goal of a web search engine is to map user-input queries to results; where a result is a web page that is somehow "relevant" to the user's query. The great power of the web as a large decentralized forum where anyone can be a publisher has a great disadvantage of no centralized mapping of content to pages - hence the existence of WSE.

The modern Web lacks an inherent centralized content map, it is necessary to analyze the web to find potential future results. The data produced from this analysis must

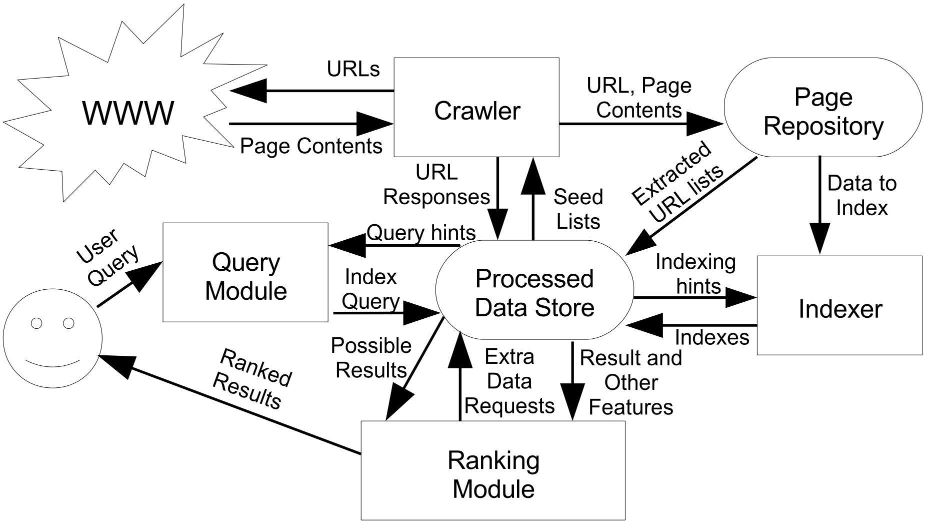


Figure 1. Architecture of a WSE

be stored in an effective form, and some type of retrieval-engine must be built around this data to enable retrieval. At query time, the user's query must be received and analyzed and then the potentially relevant candidate results must be retrieved and processed and then the "best" results returned to the user.

Published works view the search engine has having a variety of components. Figure 1 summarizes these works as six components. Papers such as [1], and Books such as [2], offer more details and describe sub-components. The six components are: crawler, indexer, processed-data-store, page-repository, query-module and ranking module.

1.1. Finding Potential Results - the crawler

A web search engine must be able to return relevant results to a user, but where do the results come from? Since there is no reverse map of the web, a search engine uses a crawler, also called a spider, to traverse the web, producing a mapping of URL to page contents, which can be used for retrieval. A crawler will start with a set of seed URLs, and fetch their contents. From those contents, new URLs will be discovered and likewise crawled. A commercial WSE might make on the order of one hundred million page requests a day. The crawler crawls pages in-advance of search results being returned, so a crawler is considered part of the off-line components of a search engine. Chakrabarti describes more details about a crawler in Chapter 2 of [3].

In addition to downloading "new URLs", a crawler must revisit previously crawled pages to ensure it has the most recent content. Returning cnn.com for a query which was in the news last week is considered an embarrassing error. The requirement of revisiting old pages, as well as the sheer size and dynamic nature of the web necessitate very resource intensive crawlers.

The crawler receives its URL lists from the *Processed Data Store* and saves the page results to the *Page Repository*. The crawler also saves the result status for each attempted crawl. If a page request generates an error, that information should be saved. In addition, some WSE might save the header response codes, or other low-level network data about a URL or domain (i.e. what IP address served it, was the response fast, etc.)

1.2. Storing The Pages - The Page Repository

The *Page Repository* provides storage for the crawled pages. The input to the page repository is the stream of $< URL, content >$ tuples. The Page Repository is used to feed other components - including the *Indexer*, Subsection 1.3, which converts the contents into a more usable form, and the *Processed Data Store*, Subsection 1.4 which stores the crawl lists (or the data used to determine them) and other pertinent information.

In theory, if we use only "words", the Page Repository might not store the entire web page, but rather just the words on the page.

The basic challenges are related to the physical size of the contents and the rate of access. Storing 10 Billion web pages can take 50 Terabytes if each web page is 5 KB. Likewise, refreshing the entire page contents every ten days, would be 5 TB/day of transfer.

1.3. Inverting the Web - The Indexer

Having a large database with contents of all web pages is not sufficient to make a WSE. A WSE must ultimately map from a *query* to a set of possibly relevant web pages, R. The simplest system would map from each query word w_k to a set of possible results R_k, and then take the intersection of the sets to produce R.

The indexer is responsible for building the data to facilitate this task, in an efficient manner. The resultant structure is called an *inverted index*, and the process of doing this at large scale is well understood [4]. The input is the extracted text from the set of crawled web pages, and the output is the inverted index.

The Indexer receives the raw page data from the Page Repository, Section 1.2, and saves its output to the *Processed Data Store*, Section 1.4. In addition, to the page data, the *Processed Data Store* can provide extra data, which I call *indexing hints*, that can be used to improve performance, and index quality. The extra data can do this by improving the ability to select what to index and what to skip, as well as how to restructure the resultant data for maximum effectiveness and performance.

As described in Section 2.3, the problems and challenges come from a combination of the scale of the problem and the variation of quality on the web, and are related to deciding what to index, what features to use, and how to make incremental changes, while maintaining performance at very large scale.

1.4. Storing The Index Data

The *Processed Data Store* stores the inverted index, as well as other useful data, and it supplies this data to the other components to facilitate the search. The Indexer builds the inverted index, the Processed Data Store saves this index and provides an interface to return data effectively to other modules. In addition to the regular inverted index, other information such as the web graph, text summary data, general link-information with inbound anchortext, or new seed sets might also be stored in the Processed Data Store. The Processed Data Store can be used to provide data directly required as well as the indexing hints such as frequency information or intermediate features that can be used to improve performance or effectiveness of other components. It is better to plan the execution of a query before querying the engine, and hint data such as knowing how popular the query is, or if certain words are likely to constitute a phrase.

In a commercial WSE, the *Processed Data Store* will typically be distributed over many machines, and there is not always a single storage for all data types. Some engines might have many different stores, each optimized for different types of tasks.

The challenges of a processed data store are the interface, performance, and reliability. Simply having an inverted index or a web graph is not sufficient. If this inverted index can't be queried hundreds of times per second, can't survive disk or hardware failures, or handle queries that return one million results, then it is not likely useful for a web search engine.

1.5. Query Module

A search engine takes a user's query, and utilizes this query to obtain, and then rank the set of results. The *Query Module* is the component responsible for taking the "natural language" query and obtaining the possible result set R from the *Processed Data Store*, Section 1.4.

The simplest way is to convert the user's query to a set of words, and then submit an *AND* query for the words, producing the possible result set R. So if the user's query were three terms, $q = w_1 w_2 w_3$ we can think of $R = R_1 \bigcap R_2 \bigcap R_3$, where R_k is the possible result set for the k_{th} term.

More advanced query modules might incorporate techniques to alter the words sent, such as: stemming, synonyms, automatic-phrasing, stopword removal or other "linguistic tricks". Likewise, more advanced systems might implicitly understand 'concepts' or phrases. A general name for the alteration of the input query to improve the search, is *query rewriting*. The fundamental challenges for a WSE *Query Module* relate to preservation of meaning of the user's intention, combined with constructing effective queries to the *Processed Data Store* - effective for both performance and result quality. An advanced *Query Module* might utilize significant extra knowledge, which I call *Query Hints*, from the *Processed Data Store*, to plan a better search strategy.

1.6. Ranking Module

The possibly relevant set R from the *Query Module* is unordered, or at best weakly ordered, and could be large. The *Ranking Module* is responsible for utilizing the data available to determine which results from R to keep and how to rank or present them.

The Ranking Module utilizes the *Processed Data Store*, Section 1.4, to obtain the "features" used to rank the pages. When ranking results in a web search engine, as opposed to ranking results in a traditional IR system, it is desirable to utilize the features of the web that separate it from traditional physical books or articles. In addition to *words on page*, a web page has local structure such as title, URL, anchortext, and each page is part of the larger web graph.

There are many algorithms and features, identified in the literature, which can be used to rank, or filter R. [5] has a very good summary of the traditional text-based IR methods, including TFIDF [6]. Langville [2] has a good summary of graph-based methods including PageRank [1].

Table 1. Properties of a Commercial Web Search Engine

Millions of heterogeneous users
Web is very large (practically infinite)
Page contents can vary over time
No quality control on pages (quality varies)
Commercial Search Engines must consider maliciousness
Result ranking is not an independent problem
Content of web page not always sufficient to imply meaning
Real-time/fast expectations
UI is extremely important
Goal is to make money

In addition to features discovered by the crawler (or analysis of crawled pages), there has also been published work on using "clickstream" data to improve relevance [7].

2. Realities of a Commercial Search Engine

It is tempting to apply traditional IR-world approaches towards designing and evaluating a WSE. However, it is important to understand the properties of a commercial search engine that are different from traditional IR retrieval systems, and consider these differences when designing and testing a WSE.

Table 1 defines several properties of a WSE. The scale of the problem at each stage, combined with the requirements of fast performance, and the existence of intentional manipulations of data, can alter the requirements for each component.

Due to the large size of the web and high performance requirements, indexing becomes even more difficult. Active attacks engineered to take advantage of known algorithms, implies more complex analysis, ranking and indexing. In addition, the lack of quality control presents entirely new challenges. A page might be textually relevant, but is not necessarily useful for a given user [8].

Researchers might also overlook the importance of the UI, which has implications for the design and evaluation of a search engine. If a user cannot effectively judge a page as being useful, then the WSE has failed. It is a non-trivial computation to produce a query-dependent summary of a multi-term query.

This Section will examine each component and present several real-world problems faced.

2.1. Crawler

Table 2 presents several challenges for a crawler.

The wide variation of content (both legitimate and manipulated), necessitates a smart crawler. The fundamental challenge for any WSE crawler is effectively balancing expansion of the page-horizon and re-crawling. A web site likely to yield low-quality content might be crawled, after re-crawling a popular site with frequently changing content. If the index has 25 Billion pages worth keeping, it is easy to imagine there are 100 Billion known URLs which were either not crawled, or crawled and not indexed due to low

Table 2. Commercial Web Search Crawler Challenges

Balancing the acquisition of new pages with re-crawling old pages
Selecting which pages are worth following (good vs useless content)
Keeping track of what has been crawled and status of responses
Dealing with and detecting “data-attacks”, or server/page content manipulations
Crawling politeness rules (robots.txt)
Dealing with dead or ’comatose’ pages
Parsing complex pages: Redirects, Javascript, AJAX

value. The problem of managing pages you don’t crawl can be as difficult as the problem of managing the set you do keep.

An additional problem is the fact that there is not a one-to-one mapping between URLs and content. URLs such as *http://www.mydomain.tld/index.html* and *http://mydomain.tld/* could be identical. It is important to be able to determine URLs are identical, to prevent wasting of crawler resources, and a consistent web graph.

A WSE crawler must deal with the fact that as your crawl-set increases, there is decreasing marginal value for crawling each new URL. Fetterly et al [9] did a study starting from a “clean seed-list”, concluding more than half of the pages from over a 400 Million page seed list were either errors or spam. Not starting with a clean seed-list, could have a higher fraction of spam and errors.

WSE Crawlers are intentionally manipulated or attacked. One kind of manipulation called a *Spider Trap*, is a dynamically generated site that is a virtually infinite set of pages, each linking to more pages (all dynamically generated by a single script). A second type of manipulation is called cloaking where a site provides different content to a WSE than to a human browser.

In addition to the problems of overload and manipulation, there is the challenge of discovering valid URLs from pages which use formats other than HTML. Additionally, many pages use Javascript, AJAX, or Flash, hiding their URLs. Some search companies use toolbars, plugins, ad-networks, or purchase user browser patterns from Internet providers to increase their set of known URLs. Unfortunately, these methods present possible privacy issues, and are not available to average researchers.

Other complications include the use of feeds or dumps. Wikipedia, a large site with millions of pages, offers a dump file. This file must be parsed and re-mapped back to URLs, using a different code-path. There are also many standards which provide hints of what to crawl on frequently updated sites, such as RSS.

2.1.1. Robots Exclusion Standard and politeness rules

A crawler for a commercial search engine is expected to follow the rules specified by the *Robots Exclusion Standard*, which specify how a crawler may access a website (politeness rules). An aggressive web robot can slowdown a website. Each site owner decides which robots (crawlers) are or are not allowed, which paths are acceptable, and what is the maximum crawl rate. For instance, mysite.tld might not want Google to crawl their /privateData/ directory. Likewise, mysite.tld might request that each robot limit their crawls to one page every minute.

Such rules create many challenges for both WSEs and researchers, and surprisingly several of these issues are business or policy related as opposed to technical. An organization may decide to block a specific WSE, for one reason or another (e.g.., due to business deals, or fear of web robots). Sites allowing only some commercial WSE crawlers, creates a problem for researchers who wish to crawl the web the same way a WSE might. Some companies might make secret deals to get permission to crawl a site, or to crawl at a faster rate, an option not available to small academics.

A second challenge is crawling large sites. A site like Amazon.com might have tens of thousands of pages. Although they want to be crawled, they might set a rate limit to one page per minute. At that rate, it could take months to view each page only once. Amazon pages might not change too frequently, but imagine a site like cnn.com whose content changes often. Considering the assumption of non-static page contents, the rules limiting access rate to a site can create severe challenges.

2.2. Page Repository Challenges

It is easy to view a Page Repository as a simple database where pages are stored associated with a URL. For a simplistic WSE this works, but in reality there are challenges for even such a simple component. Specifically, the scale creates a challenge - both scale in terms of the number of documetrnts which need to be stored, and in the frequency of access. If a crawler is distributed across many machines the rate of in-bound accesses might be large. Likewise, if the indexing process runs quickly, then there is a large volume of out-bound data.

2.3. Indexer

The indexer produces the structured data used by the *Query Module* to obtain the *Possibly Relevant Set*. One key challenge is deciding what not to index. It is desirable to skip pages which are of no value - i.e. spam pages, or intentional manipulations. Second, when doing indexing, it is important to consider the way the resultant index will be used. A user's queries might not always be reasonable to treat as a 'set of independent words'. For both runtime performance and relevance reasons, it makes sense to consider word position - either explicitly or by indexing phrases (or some combination). The scale of words alone is daunting, especially if you consider one-off strings, the scale of phrases is even greater. This does not even include the large number of "false words", i.e. binary data mistaken as text, or intentional manipulations sending hundreds or thousands of letter characters - indexing them can create problems, but if they are legitimate, failing to index can make a WSE look bad.

For example, "Hubert Blaine Wolfeschlegelsteinhausenbergerdorff", is a real person's name with a 35 character surname. Limiting the index to only dictionary words, or some maximum word length of under 35 characters would make it difficult to find relevant pages for this query.

Besides the selection of what to index based on individual page value, what about as a set. Is it necessary to index 10,000 Wikipedia mirror sites? In order to decide what to and not to index, substantial additional data, and corresponding infrastructure is required.

In theory, an inverted index is merely an unordered set of URLs that contain a given keyword or phrase. In reality, for performance reasons, it might make sense to order or partition the resultant records. If the 'best result' is ranked ten million for a particular query word, the computational costs go up significantly. Low-popularity pages might be treated (indexed) differently from high-popularity page.

In addition, the indexer might "index" information beyond the term to document map, such as position, or formatting info which could be useful for ranking. For example, an engine might first look for documents with a title-only match. The concept of feeds and RSS change the way the indexer needs to operate, since contents change frequently. Indexing using the traditional methodology for current weather data feed, or the current news summary might not make sense.

2.4. Processed Data Store

A WSE *Processed Data Store* is a high-performance sub-system designed to handle high-volume requests with millisecond response times. Commercial WSEs have many subsystems for providing data to the *Query Module*; this paper lumps them all into one logical component for brevity of discussion.

The large scale nature of the web and high query volume differentiate such a WSE from traditional databases. A simple inverted index has problems when the size of the intermediate R lists is very large. When querying for *the white house* there could be a hundred-million possible results for each term. The processed data store needs to efficiently store and allow access to the required data. If the score of a web page is determined by the term positions on the page, inbound anchortext, and local-graph structure, simple methods can become too computationally expensive for real-time operation. A commercial WSE has milliseconds to consider millions of possible results. One approach is to plan a query execution path that reduces the expected size of the consideration set, possibly initial filtering on a gross level, or use of query rewrite rules. A WSE's Processed Data Store needs to be distributed, redundant and optimized for high performance.

The Processed Data Store stores more than the inverted index. It also saves other page and query-related data, such as graph and category information, and possibly editorial judgments. The individual pieces must be built to work in real-time with high reliability, and resistance to attacks, as well as easy to update.

Some search engines might have multiple Processed Data Stores, and dynamically choose at query time based on the query or system load. One could be for "popular domains" and another for "lower quality domains" for example.

Given the wide variation of users and queries, it is critical that the Processed Data Store be able to remain functional under all circumstances. There should be no possible query which can cause the Processed Data Store to fail or excessively slow down.

2.5. Query Module

The *Query Module* takes a query and determines the *possibly relevant set*, R. The simplest way to do this, is to break the query into individual terms, take the result set for each term and then take the intersection - effectively taking an AND of all of the query terms. In order for this to work, there are two assumptions: First, that the words on a page are indicative of the purpose of the page. Second, that the words in the query are identical to the words on a relevant page.

The Query Module should locate all possibly relevant results, or in IR terms, high recall, before passing them on to the *Ranking Module*. Like IR systems for non-web documents, Web Search inherits the typical problems of *synonymy* and *polysemy*, but with a few new twists. First, unlike other IR systems, there are intentional attempts at manipulation (in the form of spam, page cloaking, "google bombs", or other attacks) - although from a theoretical standpoint retrieving extra documents is acceptable, if the set of 'low quality' results is too great, there is a performance issue. Second, there is a tremendous variety of users and needs on the web. Given the multitude of possible meanings and intentions, and the variety of skill levels, retrieval is even more difficult.

For many types of queries, including navigational, the "best page" might not even contain the query terms anywhere on the page. Consider, for instance, that few web search engines include the text "web search engine" in the text of their home page. Several classes of queries are especially difficult for traditional methods, and it is critical that the Query Module, and the rest of the system handle such queries without problems. Queries which are especially difficult include duplicated words: *bora bora* (a place), queries with stopwords which matter *the who* (the band), numbers *24* (TV show), or worse combinations *the the* (a band). Also some users may intentionally or accidentally send bad data as a query, such as attempts at buffer overflow, or binary data, or random characters. Most popular WSE are designed to operate in many languages, which can further complicate the task of query processing and planning.

Another complicating factor is the integration of non-textual/non-web results. In a simple WSE, the user's query is sent to a large inverted (text) index. When a user asks a query about a recent event, the results might come from a third-party provider as opposed to the traditional index. Likewise, if a user asks for pictures of something, a simple inverted index might not be sufficient.

2.6. Ranking Module

The *Ranking Module* is probably the most complex, and the *relevance* problem is so important, that Section 3 is devoted to understanding the problems and current methods for evaluation of relevance for a WSE.

A simple Ranking Module takes the set of possibly relevant results, R, from the Query Module and produces an ordering. Many of the problems presented in Section 2.5 apply here. Such as, the issue of words-on-page and the limited ability of words alone to predict the usefulness to a query.

A WSE combines data from many features to select and rank results. Some possible "features" include: words on page and location, classification related features such as quality, topic, query-intention-prediction, use of human editorial judgments, use of the link-graph (PageRank type computations and in-bound anchortext), clickstream data, Natural Language Processing (NLP), structured data, enhanced page parsing (i.e. for local searching to extract addresses), user query history and others which are not made public. Each commercial search engine has its own unique set of features and its own specific way of combining and utilizing them. Some companies use specialized machine learning, while other prefer manual control over how various features are combined.

2.7. Missing Components

It is important to keep in mind that there are several tasks important for a search engine, but are not shown as explicit components in Figure 1. Table 3 lists several additional tasks or functions which are important for a WSE. In addition, there are other areas of difficulty, which are not discussed in this paper including dealing with multiple character

Table 3. Important tasks and functions not included in the simple architecture model

Caching - between most modules, such as query to results, results to metadata, descriptions
Graph processing - Data from Page Repository to Processed Data Store
HTML Parsing - Extraction of text and links from pages
Spam detection - Prediction of pages or domains which are spam
Duplicate detection - Identify pages which are near or exact duplicates
Mirror site detection - Identify entire domains or paths which are mirrors
Feed integration - Allow feed data to be searched and indexed
Manual editorial control - Human editorial influence of results and ranking
User logs - collection and integration (clickstream, etc...)

encodings, or dealing with multiple languages (especially languages such as Japanese or Chinese which require segmentation).

3. Relevance

Everyone talks about *Relevance*, and how important it is - but despite numerous papers on approaches for improving web relevance, there still seems to be a limited understanding of the problems affecting a WSE.

3.1. Academic Evaluation Methods

The primary concepts used for measuring relevance of academic search systems are *precision* and *recall* or derivatives from them. Precision is loosely defined as the fraction of *retrieved* results that are *relevant*, while recall is the fraction of *relevant* results that are *retrieved*. Roughly speaking precision is a measure of what the user sees that is judged as relevant, while recall is the measure of the ability of a search system to retrieve all relevant results. Precision is further extended to P@X or the Precision of the first X results, and extended to MAP or Mean Average Precision, which is a measure that weights the precision of top ranked documents more, and takes the average over a set of queries. There are other measures such as F1, also called the *f*-measure, which is a combination of precision and recall. For good references on measures in IR see [10], [11], [12], [6].

Precision, Recall, P@X, and MAP all assume that relevance is a binary judgment. MRR or Mean Reciprocal Rank is a measure of the inverse of the rank of the best result, such that the worse the rank, the lower the score.

It should be noted that recently, both commercial search engines and academics have realized the deficiency of using a binary notion of relevance for evaluations. The measure Normalized Discounted Cumulative Gain (NDCG)[12] is now used by commercial search engines to do internal evaluations. Cumulative Gain measures such as NDCG are especially useful when combined with machine learning partially because they try to compare how far from “perfect” a particular set of results is. Roughly speaking for a single query, NDCG (at some number of results) is 1 for a perfect ranking. Typically search engines will use a five point scale for relevance for NDCG calculations.

3.2. A Perfect Relevance Function

To understand some of the challenges faced by a modern WSE in terms of *relevance*, let us begin with a theoretically perfect relevance function, $s = F(q, u)$. Where the score is s, user's query is q and u is a URL or possible result. This "perfect" relevance function has the following properties:

$\forall u_x, u_y \in U, q \quad s_1 = F(q, u_x)$, and $s_2 = F(q, u_y)$ a user prefers u_x to u_y $iff s_1 > s_2$

Let us assume for the moment, that there is only one user (i.e. function F is tailored to this user), and that this user has consistent (transitive) preferences across all URLs in the set U for any query q.

Assuming we used user's judgments for relevance, and that we could map score to a binary judgment, then all measures should in theory be perfect. All relevant results would rank higher than all non-relevant results, and the best result would always be ranked first.

Even though, such a magical function F is unreasonable, there are issues and problems. Such a function would have a perfect score for the above measures, but it would not be sufficient to make a good commercial search engine. The main problems arise from the false assumption that user assessed quality/usefulness is independent of the set and order of the other results. Not only does changing the order affect the overall value, but it affects the value of other shown results.

The first problem is that on the web there are many cases where two different URLs have identical content, such as www.X.tld and X.tld. There are also 'near duplicates' such as sites which include the identical AP-Newswire source article and only change the advertisements and nav-bars around it. There are also many mirror sites which can be annoying if shown to users.

Even after duplicate removal, there is still the notion of marginal value. A user might prefer a news result to a picture page, but that does not mean they want all news results before all picture pages. The usefulness of the second result is not necessarily independent of the first result.

The third problem is that a perfect ranking is only useful if the user can identify a result as relevant. Most major search engines enhance their UI to facilitate a user identifying a useful result. Being able to figure out what description or text to display is as important as figuring out which results to include. This includes a recognizable title, recognizable URL (when there are duplicates, pick the one the user recognizes), meaningful description or summaries. Most major search engines modify the title, URL and or summary from the 'text on page'.

3.3. Realities and complications

Although the perfect relevance function described in 3.2 would have value, it is unrealistic. The problem is users are heterogeneous. The same query might mean something very different to two different users. For example, "Michael Jordan" is both a famous basketball player and a professor at Berkeley. One meaning (the basketball player) dominates, but that does not mean a search engine should show only results for that one meaning. Dealing with multiple meanings for ambiguous queries is a hot topic for both commercial search engines and academics, which is briefly discussed in Section 4. Variation of users and their needs make studying search engines difficult, and further demonstrate the limited value of utilizing text-only features or binary relevance judgments.

Like the other components of a web search engine, the methods used for ranking results are also actively attacked. Fundamentally, any ranking decision comes down to features - either location or frequency of terms, or other features such as link-graph related features, in-bound anchortexts, clickstream analysis etc. If it becomes public the exact ways in which a commercial search engine does ranking, then it is likely attacks will quickly appear. There are pages which are generated using markov-text generators, but designed to give key placement to specific words. There are entire sites made to "look like" an expert site on a single topic. There are organizations which specialize in helping people to get in-bound links to raise their PageRank, and articles written on how to rank higher in Google.

4. Dealing With the Problems

Search Engines generally work, so clearly there exist approaches to minimize these problems, or at least hide them from users. In addition, there are many academics who are actively contributing to both the science as well as the effectiveness of commercial search engines. There are several interesting approaches described in other chapters in this book. Surprisingly, many solutions employed by commercial search engines might be academically boring, such as using a simple list of trusted hosts or human editors to flag spam.

A basic approach to many commercial web search problems is to minimize the user-perceived effects - without necessarily solving the problem. This can be done in many ways - including biasing the evaluations to be consistent with real users. Likewise, search engines are aggressively using clickstream data as a means to leverage massiveness as a mechanism to help hide the limitations of the other algorithms. It might be easy to foil a link-based algorithm, but if millions of users pick the 'best result' for a query, it can reduce the effect of link-spam attacks for that particular query. This is an important point for academics - just because a particular algorithm or approach has a limitation, and might appear to evaluate poorly, does not mean it will fail in the commercial search world (as long as there exist known methods to cover up or eliminate that limitation).

4.1. Crawler

WSEs prefer simple solutions. Priority crawl lists and general crawl prioritization can be used to focus time and resources on the sites which are likeliest to be clean and useful. Robots.txt related problems can sometimes be addressed through business deals; calling up companies asking them to unblock you or alter the priority. Researchers in this area can focus on how to effectively balance and detect content changes and develop efficient ways to manage this information at web scale. [13] is dedicated to this area.

4.2. Page Repository and Processed Data Store

Page repository issues are ideal for research in the areas of distributed storage and computation, as long as the work assumes real-world search engine data sizes (petabytes). A successful page repository will be able to handle the bandwidth of a crawler and indexer simultaneously, be resilient to hardware failures, and be able to have some backup/restore/history capabilities.

The Processed data store, like the page repository must be high-performance and optimized for the desired task and performance. In general most engines will have multiple sites and multiple distributed copies of their data. Likewise, this distribution might not be symmetric; an important site or page might have more copies than one deemed low-quality.

4.3. Indexer

The issues of the index are related to the scale of the web, and lack of quality control - which map to deciding what to index and what not to index - both pages and words on a page. There are two issues: ability to selectively index (i.e. excluding certain pages, or certain words from pages), and ability to capture complex concepts while preserving efficiency (this could include auto-phrasing, stemming or concept expansion). Some engines might have multiple indexes, one for high-quality pages, and another for everything else. Some engines might use some lists of words or phrases and treat them differently. Unfortunately, the value of a change to the indexer is not easy to evaluate in isolation, and depends on how the index is used.

There have been significant improvements in NLP (Natural Language Processing) with regards to performance, although it is unclear exactly where and how NLP should be applied to commercial web search. Effective incorporation of NLP could be done through a specialized field of entities that are extracted, as opposed to ‘in-place’ extraction. Separating out the entities from the index also mixes well with manual or clickstream enhancement. There are specialized search sites, such as `http://www.zoominfo.com/` which have their own web crawlers and use specialized entity extraction - in the case of ZoomInfo, they focus on people and company relations.

4.4. Query Module and Ranking Module

The query module and ranking module have substantial active research - which could be useful for commercial search engines. Commercial search engines employ many techniques for enhancing the perceived relevance, they are not discussed in detail in this paper.

Improving the retrieved set can be done through *query rewriting* or integrating other sources or partner sites (as well as other methods). Improved understanding of the query, through semantic analysis, statistical analysis, spell checking, clickstream mining, or other methods can be used to enable more effective querying of the *Processed Data Store*. When querying most major search engines, it is obvious that the engine did not treat your query as a ’bag of words’. This is most visible with automatic spell correction and sometimes with the engine inserting synonyms or alternate forms of your query words.

Major areas of semantic analysis include identifying specific entity types, such as products, companies, or person names; vanity and company navigational searches are a significant percentage of web search traffic. There are several search engines now specializing in the area of semantic analysis including `http://www.hakia.com/` and `http://lexxe.com/`. There is also a lot of academic work on ’query understanding’ and ’entity extraction’ including [14] which utilizes corpus statistics to improve accuracy of semi-supervised relation extraction.

Improving the ranking can be done through many methods. Specifically, the problem of ambiguous queries is addressed in many ways. Most major search engines provide some type of clustering of results - for highly ambiguous queries, this clustering is likely to map to the popular meanings. Most major engines also offer some type of 'query suggest' mechanism where alternate queries are listed to aid the user in constraining their search. SearchMe Inc. approaches the problem through classification of the whole web into an ontology and presenting the user with "categories" to chose from, allowing them to explicitly narrow by topic or page type. There are many academic works and commercial engines which try to enhance search through "semantic understanding", result or page classification, and alternate User Interfaces.

5. Conclusion and Overview

This paper summarized the logical components of a web search engine. Then described several of the challenges faced by real search engines, and how the simplistic models are insufficient. Several common problems include the implications of the scale of users and data, as well as the effects of intentional attacks against the fundamental workings of a search engine. In addition, many of the models and methods used to evaluate IR systems break down when applied to existing commercial web search engines.

References

[1] Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30**(1–7) (1998) 107–117

[2] Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press (2006)

[3] Chakrabarti, S.: Mining The Web. Morgan Kaufmann (2003)

[4] Witten, I.H., Moffat, A., Bell, T.C.: Managing Gigabytes: Compressing and Indexing (second editon). Morgan Kaufmann Publishing (1999)

[5] Jones, K.S., Willett, P.: Readings in Information Retrieval. Morgan Kaufmann (1997)

[6] Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill (1983)

[7] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Transactions on Information Systems (TOIS) **25**(2) (2007)

[8] Glover, E.: Using Extra-Topical User Preferences to Improve Web-Based Metasearch. PhD thesis, University of Michigan (2001)

[9] Fetterly, D., Manasse, M., Najork, M.: Spam, damn spam, and statistics. In: Proceedings of the Seventh International Workshop on the Web and Databases. (2004)

[10] Singhal, A.: Modern information retrieval: A brief overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering **24**(4) (2001) 35–43

[11] Wikipedia: Information retrieval performance measures `http://en.wikipedia.org/wiki/Information_retrieval#Performance_measures` (2007)

[12] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)

[13] Cho, J.: Crawling the Web: Discovery and maintenance of large-scale web data. PhD thesis, Stanford University (2001)

[14] Rosenfeld, B., Feldman, R.: Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. (2007)

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-130

Website Privacy Preservation for Query Log Publishing [1]

Barbara Poblete [a], Myra Spiliopoulou [b] and Ricardo Baeza-Yates [c]
[a] *Web Research Group, University Pompeu Fabra, Barcelona, Spain*
[b] *Otto-von-Guericke-University Magdeburg, Germany*
[c] *Yahoo! Research, Barcelona, Spain*

Abstract.
In this article we introduce and discuss briefly the issue of privacy preservation for the publication of search engine query logs. In particular we present a new privacy concern, *website privacy* as a special case of *business privacy*.

Keywords. Query Logs, Websites, Data Mining, Privacy Preservation

Introduction

Query logs are very rich sources of information, from which the scientific community can benefit immensely. These logs allow among other things the discovery of interesting behavior patterns and rules. These can be used in turn for sophisticated user models, for improvements in ranking, for spam detection and other useful applications. However, the publication of query logs raises serious and well-justified privacy concerns: It has been demonstrated that naively anonymized query logs pose too great a risk in disclosing private information.

The awareness towards privacy threats has increased by the publication of the American Online (AOL) query log in 2006 [2]. This dataset, which contained 20 million Web queries from $650,000$ AOL users, was subjected to a rather rudimentary anonymization before being published. After its release, it turned out that the users appearing in the log had issued queries that disclosed their identity either directly or in combination with other searches [3]. Some users even had their identities published along with their queries [4]. This increased the awareness to the fact that query logs can be manipulated in order to reveal private information if published without proper anonymization.

Privacy preservation in query logs is a very current scientific challenge. Some solutions have been proposed recently [5,6]. Similarly to the general research advances in privacy preserving data mining, they refer to the *privacy of people*. Little attention has been paid to another type of privacy concern, which we consider of no less importance: *website privacy* or, more general, *business privacy*.

[1] For the full paper refer to [1].

Indeed, important and confidential information *about websites and their owners* can be discovered from query logs and that naive forms of URL anonymization, as in [3], are not sufficient to prevent adversarial attacks. Examples of information that can be revealed from query logs include accesses to the site's documents, queries posed to reach these documents and query keywords that reflect the market placement of the business that owns the site. Such pieces of information *are* confidential, because websites serve as channels for advertisement, communication with potential customers and often sales to them. Hence, the traffic recorded in them delivers a picture of customer-company interaction, possibly for the whole product portfolio. A thorough analysis of this traffic with a data mining method may then deliver information like insights on the effectiveness of advertising campaigns, popular and less popular products, number of successful and failed sale transactions etc.

Example 1 (Disclosing confidential company related information) *Websites* A *and* B *are on-line book stores and use the Web as their only sales channel.*

Website A *knows from their own website log that* $m_a\%$ *of the visits from external search engines lead to a purchase. They also know that* $n_a\%$ *of their buyers come from search engine* X*. They further know the set of frequent queries used by these visitors to click at their site. These pieces of information are obviously confidential: They are private data to company A and are valuable for planing marketing campaigns, investing in on-line advertisements and addressing users directly.*

Assume that the owner of the search engine X publishes their query log L with the same anonymization procedure as used by AOL [3]. Site A *can use this* published *log together with their own* private *website log of the same period* L_A *to derive confidential information about website* B*. In fact,* A *can derive the* pieces of private data *about* B *as they have for their own site!*

Site A *can analyze* L *and website* B*'s public website, to disclose confidential information about* B*. This involves finding all the queries in L from which users clicked on a document in site* B*. Next,* A *uses these queries to compute the absolute number of accesses to the website* B*,* n_b*. Since* A *and* B *address the same target group,* A *can use* n_a, n_b *and the percentages* $n_a\%, m_a\%$ *to compute* $n_b\%, m_b\%$*. Further, if they know that the behavior of visitors coming through search engine* X *is representative of all visitors to site* A*, then they deduce that the same holds for the visitors to site* B*. Thus, they can identify frequently accessed features, such as authors and genres for* B*'s book sales and derive their competitive advantages and disadvantages in the business.*

One may argue that a site's traffic is only recorded at the site's server and therefore not public. However, the traffic delivered to a website by major search engines accounts for an important part of the site's overall traffic. If this part is undisclosed, it will be a very accurate approximation to the complete access log of the website.

The protection of such confidential information is different from conventional privacy preservation. One reason for this difference is that an adversary can reveal confidential website information by aggregating a published query log with other legally owned private data. In particular, consider an adversary which is a company interested in disclosing information about its competitors. This adversary could use its own background knowledge and the data of its own site in combination to the published query log data, to infer the competitor's private data. This includes but is not limited to popular queries that reach *both* the adversary's site and that of the competitor. The log of the adversary can then be used to de-anonymize a part of the published query log. Depending on the amount and quality of the information revealed, *industrial espionage* or malicious intent could be argued by the affected parties against the company that published the query log.

Although query log anonymization does not look promising in the near future, especially from the user privacy perspective, we believe that reasonable measures can be taken to preserve website privacy. The work presented in [1] discusses some of the existing threats and ways to prevent them, this helps to set a precedent for data mining applications on logs, and future query log publishing. The goal is that the information generated from query log mining applications is inspected to prevent privacy leaks. Although [1] focuses on website privacy, we believe that this approach also contributes to user privacy, because much of the sensitive information about users comes from assessing the pages they have visited. The contributions of [1] are: (1) to introduce a new privacy issue for query logs, *website privacy*. (2) Describe attacks that disclose confidential information from query logs, and ways to prevent them. (3) Propose a heuristic graph-based method that removes those parts of the log that may lead to information disclosure.

1. Related Work

The rapid development of advanced techniques for data collection and propagation, along with the fast growth of the Web, have increased the awareness to the use of private information. This has lead to a new field of research in the context of analyzing private or confidential information – the domain of *privacy preserving data mining* [7].

Privacy preserving data mining aims at analyzing databases and data mining algorithms, identifying potential privacy violations and devising methods that prevent privacy breaches. Preventive measures involve the hiding or modification of sensitive raw data like names, credit card numbers and addresses, and the exclusion of any further type of sensitive knowledge that can be mined from the database. It is important to note that many privacy preserving algorithms are based on heuristics. This is because of the premise that selective data modification or sanitization is an NP-hard problem.

Some research on privacy preservation in databases deals with privacy preserving data publishing that guarantees utility for data mining [8,9]. There are studies on preventing adversarial data mining in relational databases, when data fields are correlated [10]. Samarati and Sweeney proposed *k-anonymity*, in which data is released in such a way that each query result (and each attempt for data disclosure) returns at least k entities [11]. The principle of k-anonymity is quite effective but it cannot be directly applied to data that expands across multiple databases, as is the case of website privacy preservation.

In the context of Web mining, one of the prominent areas for privacy preservation is the protection of user privacy in query logs of search engines. Among the advances in privacy preserving Web mining, most relevant to our work are the studies of Kumar et al [5] and of Adar [6]. Kumar et al propose token-based hashing for query log anonymization [5]; The queries are tokenized and a secure hash function is applied to each token. However, the authors show how statistical techniques can be used to disclose private information despite the anonymization; they also show that there is no satisfying framework to provide privacy in query logs [5].

In [6], Adar explains many aspects of the AOL query log problem, and shows that traditional privacy preservation techniques cannot be applied in a straightforward way to protect privacy in a search log. Further, Adar argues that k-anonymity is too costly for rapidly changing datasets like query logs. Then, Adar proposes two user anonymization methods for query logs, which attempt to balance log utility for research and privacy [6].

2. Challenges for Query Log Anonymization

Anonymizing query logs for data mining is very challenging for several reasons. First, the attributes of the query log are not independent. An adversary may use these dependencies to deduce the value of an anonymized field. For example, queries in search engines are known to exhibit a remarkable frequency distribution: Kumar et al exploited this property to decrypt anonymized queries by studying the frequency and co-occurrence of terms in a non-anonymized reference log [5]. Moreover, query logs have *sequential* records: Rearranging or shuffling them for anonymization purposes would blur or eliminate important temporal and order-dependent information, such as user sessions.

Despite these observations, we should keep in mind that data mining focuses mostly on extracting knowledge in pattern form and does not always require exact values for each attribute: Such values can be replaced by an anonymized value that preserves their distribution. However, for Web query mining, it is difficult to determine which attributes should be anonymized or hidden: *all* attributes in the log are of potential use – depending on the purpose of the analysis. Thus, the minimization of the private information that could be disclosed by an adversary while maintaining enough information for data mining becomes a complex optimization problem.

References

[1] Poblete, B., Spiliopoulou, M., Baeza-Yates, R.: Website privacy preservation for query log publishing. In: Proceedings of the PinKDD Workshop on Privacy, Security and Trust in Data Mining at the ACM SIGKDD Int. Conf. on Data Mining and Knowledge Discovery (PinKDD'07).

[2] AOL research website, no longer online. (http://research.aol.com)

[3] Arrington, M.: AOL proudly releases massive amounts of private data. http://www.techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/

[4] Barbaro, M., Zeller, T.: A face is exposed for aol searcher no. 4417749. New York Times (2006)

[5] Kumar, R., Novak, J., Pang, B., Tomkins, A.: On anonymizing query logs via token-based hashing. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM Press (2007) 629–638

[6] Adar, E.: User 4xxxxx9: Anonymizing query logs. In: Query Log Analysis: Social and Technological Challenges, Workshop in WWW '07. (2007)

[7] Verykios, V., Bertino, E., Fovino, I., Provenza, L., Saygin, Y., Theodoridis, Y.: State-of-the-art in privacy preserving data mining. SIGMOD Record **33**(1) (2004) 50–57

[8] Chawla, S., Dwork, C., McSherry, F., Smith, A., Wee, H.: Toward privacy in public databases. Theory of Cryptography Conference (2005) 363–385

[9] Kifer, D., Gehrke, J.: Injecting utility into anonymized datasets. Proceedings of the 2006 ACM SIGMOD international conference on Management of data (2006) 217–228

[10] Aggarwal, C., Pei, J., Zhang, B.: On privacy preservation against adversarial data mining. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (2006) 510–516

[11] Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report (1998)

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-134

Fighting Web Spam

Marcin SYDOW [a1], Jakub PISKORSKI [b], Dawid WEISS [c], Carlos CASTILLO [d]
[a] *Web Mining Lab, Polish-Japanese Institute of Information Technology, Warsaw, Poland*
[b] *Joint Research Centre of the European Commission, Ispra, Italy*
[c] *Poznań University of Technology, Poznań, Poland*
[d] *Yahoo! Research Barcelona, Spain*

Abstract. High ranking of a Web site in search engines can be directly correlated to high revenues. This amplifies the phenomenon of Web spamming which can be defined as preparing or manipulating any features of Web documents or hosts to mislead search engines' ranking algorithms to gain an undeservedly high position in search results. Web spam remarkably deteriorates the information quality available on the Web and thus affects the whole Web community including search engines. The struggle between search engines and spammers is ongoing: both sides apply increasingly sophisticated techniques and counter-techniques against each other.

In this paper, we first present a general background concerning the Web spam phenomenon. We then explain why the machine learning approach is so attractive for Web spam combating. Finally, we provide results of our experiments aiming at verification of certain open questions. We investigate the quality of data provided as the Web Spam Reference Corpus, widely used by the research community as a benchmark, and propose some improvements. We also try to address the question concerning parameter tuning for cost-sensitive classifiers and we delve into the possibility of using linguistic features for distinguishing spam from non-spam.

Keywords. Search Engines, Web Spam, Machine Learning, Linguistic Features

1. Introduction

Web spamming is any form of manipulating the content, link-structure [1] or other features [2] of Web hosts and documents to mislead search engines in order to obtain undeservedly high ranking in search results. Since high ranking in search engines is positively correlated with high revenues, the motivation is almost purely economical.

Web spam combating has been regarded as the most urgent problem in the Web information dissemination process for many years [3] because it significantly deteriorates the quality of search results and thus affects the whole Web community. In addition, unhappy users can turn to competition and this translates to significant revenue cuts for search engines. Since Web spamming has such significant social and economic impact, the struggle between search engines and spammers is an "arms race": both sides apply increasingly sophisticated techniques and counter-techniques against each other.

Rapid progress of spamming techniques, an increasing number of factors to consider, and the adversarial nature of the spam phenomenon require novel techniques of

[1] The first author was supported by the Polish Ministry of Science grant: N N516 4307 33

dealing with the problem. Recently, machine learning has been successfully applied to support Web spam combating [4,5].

1.1. Outline of the paper

The outline of this paper is as follows. We start by presenting the background in Section 2 which mentions the dominating role of the search engines in the Web and emphasises the role of ranking algorithms in the search process (Section 2.1). Then, we briefly describe the Web economic model (Section 2.2) which clearly explains the motivations behind the Web spamming phenomenon.

In Section 3, we describe what is usually regarded as Web spam, present a Web spam taxonomy (Subsection 3.1) and give some remarks on strategies for combating Web spam (Subsection 3.2).

Section 4 outlines the state of the art in the field of Web spam detection. We mention the reference corpus (Subsection 4.1)[2] prepared recently to help the research community in a systematic comparison of automatic Web spam detection methods and related activities (Subsection 4.2).

Next, we discuss various approaches concerning the use of machine learning with respect to Web spam detection (Subsection 4.3).

The application of the concept of *trust* is separately discussed in Subsection 4.4 due to its important role in automatic spam detection.

Section 5 introduces several open questions concerning usefulness of linguistic features in the context of Web spam classification and unbalanced training class sizes. Some of these questions stem from previous work on the subject (most notably [5]), but we also investigate an unexplored direction of using linguistic features in Web spam detection. The remaining part of Section 5 contains the description of experiments and results achieved.

We conclude and discuss possible future work in Section 6.

1.2. Contribution

Applications of machine learning techniques for fighting Web spam have been in the centre of attention recently (see Section 3). Our contributions presented in this publication are listed below.

- We explore the possibility of using linguistic features contained on Web pages for detecting and classifying spam. In section 5.2 we present the attributes we computed and added to the previous attribute set. Preliminary results show that some of the features are potentially promising and they exhibit some discriminative power in automatic Web spam classification. To our best knowledge, such features have not previously been used in the context of Web spam detection (although they have been applied in other fields).

- We observed that inconsistent labelling present in the reference corpus (see Section 4.1) may lead to unnecessary deterioration of the classification quality. Our results (5.4) indicate that cleaning the data by removing non-univocally human-labelled training examples makes the resulting decision trees much simpler while

[2] A new larger corpus is currently being prepared by the research community, to be available in 2008.

the classification accuracy is not deteriorated. These results seem to be important and applicable to the preparation of future versions of the reference Web spam corpus.

- We repeated classification experiments for a wide range of cost values used in the cost classifier, trying to complete previous research done in [5]. Our results (Section 5.3) shed more light on the impact of the cost parameter on the size of decision trees, accuracy of classification and selection of significant attributes.

2. Background

The Web is a large source of information encompassing petabytes of publicly available data on innumerable Web pages. While it is not possible to tell exactly the size of the Web (due to the existence of dynamic documents and the impossibility of taking an instant snapshot of the Web), there are techniques for estimating the size of the "indexable" Web [6]. At the time of writing the number of indexable Web documents is estimated as 25 billion.[3]

2.1. The Role of Ranking in Search Engines

To make any use of that huge amount of available information, Web users use search engines, which became the de facto main gate to the Web. Search engines themselves are huge and complex systems, answering hundreds of millions search queries over hundreds of terabytes of textual corpora daily. Discussion of main architectural and technical issues concerning large search engines can be found in [7],[8], or [9].

Processing huge amounts of data on enormous load rates is a challenge, but the most difficult problem in search technology emerges from the very fact that most users look only at the first page of search results, containing typically 10–20 results. Thus, the primary task of a search engine is to automatically sort all the results according to their relevance, authority and quality so that the best results are at the top of the matching list of thousands or millions matching candidates. This is the task of the *ranking system* module — perhaps the most secret, key component of each search engine.

Ranking algorithms, which determine the order of the returned results, use all the aspects of the information explicitly or implicitly connected with Web documents to decide the ranking position of a search result among the others. These aspects of information include (but are not limited to):

- textual contents of the body, meta-tags and URL of the document, as well as anchor text (the text snippets assigned to the links pointing to the document),
- the link structure of the whole Web graph,
- various statistics derived from query logs,
- estimates of Web traffic.

[3] http://worldwidewebsize.com

Textual components of ranking techniques are derived mostly from classic IR systems [10,11] and are based on variants of the term-frequency-inverse-document-frequency (*tfidf*) score. More sophisticated link-structure ranking algorithms were introduced to automatic search systems more recently (around 1998; e.g., [12,13]) and are still being intensively developed.

2.2. *What drives the Web?*

Search engines are the heart of the Web [14]. One can ask about the business model which makes commercial search engines do their business. The answer is that the main source of income for search engines is *advertising*. Search-related advertising can be divided into two main categories: *sponsored links* (paid links to commercial destination pages shown alongside search results) and *contextual ads* (shown on third party Web sites). Both types of advertising rely on the search engine's technology of matching between keywords provided by the advertiser and the context: in the first case an ad is matched against the user query, in the second, against the contents (and other contexts) of the hosting Web page.

The income of search engines increases with the number of advertising customers (ad hosting Web pages share this profit proportionally). There are several different models of charging for ad's appearance: the number of impressions (cost-per-mille; CPM model) of an ad, the number of actual clicks on an ad (cost-per-click; CPC model) or the number of actual transactions made as a consequence of clicking on the ad (still the least popular model, but of increasing interest). Note that search engines try to achieve the best possible matches to make participation in ad programmes commercially attractive, but also to increase their own profit (well targeted ads are likely to be clicked on).

The total income of search-based ads in 2006 in the USA was around \$6.7 billion, and constitutes 40% of the total internet-based advertising revenue.[4] Furthermore, this figure grows at a very fast rate—35% in 2006.

As the very important consequence of the Web economic model described above is that Web traffic directly turns into real profit, due to the existence of contextual advertising programs.

Bearing in mind that search engines constitute the actual "main gate" to the Web, we can make the following statements:

- ranking algorithms of search engines determine and influence the actual visibility of particular Web pages,
- the more a Web page is visible (better ranking in search queries) the more traffic goes to it (more users will eventually visit the page due to its presence among the top search results),
- more traffic on the page means more potential income (due to the contextual ad programs).

Thus, to conclude the above: *it is commercially attractive to boost ranking positions in search results*. This is the main rationale behind the existence of the Web spamming phenomenon.

[4] Internet Ad Revenue Reports, http://www.iab.net/.

3. Web spam

Web spam (or search engine spam) can be described as any deliberate manipulation of Web documents intended to mislead ranking algorithms of search engines in order to artificially boost the ranking position *without* actually improving the information quality for (human) Web users. Another, somehow extreme, description is: “Web spamming is everything Web authors do only because search engines exist”.

The above descriptions are obviously not strict definitions — they leave much room for ambiguity which is inherent to the issue. In practice, most search engines provide their guidelines (for webmasters) to reduce the ambiguities to the minimum about what is considered spam and what is not. Note that spam is consequently punished, usually by removing the documents or hosts from indexes, thus reducing the visibility to zero.

3.1. Spam taxonomy

In [2], spam techniques are classified into two broad categories: boosting techniques and hiding techniques. Boosting techniques influence the ranking used by search engines by altering the contents, links or other features of the pages or hosts. Hiding techniques serve as camouflage for other spamming techniques (e.g., hidden text or links) or provide two different views of a page to the user and to the search engine, e.g. by means of quick and automatic redirect of the user from the page indexed by the search engine to another spam page.

In general, spam techniques aim at modifying the view of documents that search engines use for indexing the pages by modifying the contents, links, etc. of the pages. Content-based techniques include copying or repetition of phrases perceived by the spammer to be relevant for their business, and in some cases, hiding such terms by using hidden text (either very small, or the same colour as the background of the page[5]), or use non-visible parts of the HTML code such as meta tags or alternate descriptions for multimedia objects which are embedded in the HTML.

Link-based techniques aim at link-based ranking algorithms such as PageRank [12] or HITS [13] by manipulating the in-links of a page. This can be done by creating a *link farm*: a tightly-knit community of pages linked to each other nepotistically [15]. The components of the link farm can be pages under the control of the spammer, pages that agree to enter a link-exchange program with the spammer, or external pages in sites that allow world-writable content. The latter is the case of wikis, forums and blogs where, if the appropriate anti-spam measures are not taken, spammers typically post links to sites that they want to boost.

Web spam affects the whole Internet community given that it deteriorates the quality of search results, and thus breaches the trust relationship between users and search engines. It also affects search engines themselves given that it forces them to waste network and storage resources indexing content that is not valuable for its users.

3.2. Fighting Web spam

Defeating Web spam does not require perfection, but only to alter the economic balance for the would-be spammers [16]. A Web site owner will spam if she or he perceives that

[5] This technique used to be popular but now it is quite easy to be detected.

it is economically justified to pay more to spend a certain amount of money in spamming a search engine instead of spending the same amount of money in improving his or her Web site. While a group of spammers perhaps is able to make profit in the short term, this is not true in general and certainly not true in the long term. The first steps decreasing the amount of spam on the Web is to educate users about how to improve their Web sites to make them more attractive to users without using deceptive practises.

Search engines can also explain to users what is regarded as spam and what is not. For example, [17] advocates that search engines develop a clear set of rules and equate these rules to the "anti-doping rules" in sport competitions. Following the same analogy, search engines should increase the cost of spamming by demoting pages that are found to be using spamming techniques. Search engines also maintain spam-reporting interfaces that allow the users of the search engine to report spam results.

Numerous factors of Web documents have to be analysed to decide whether a given document is spam or not. In addition, the process of inventing new spamming techniques by spammers and subsequent updating of the ranking algorithm by search engines (in response) clearly resembles a never ending arms race.

4. Web spam detection

The development of an automatic Web spam detection system is an interesting problem for researchers in the data mining and information retrieval fields. It concerns massive amounts of data to be analysed, it involves a multi-dimensional attribute space with potentially hundreds or thousands of dimensions, and is of an extremely dynamic nature as novel spamming techniques emerge continuously.

4.1. Public corpus of spam data

The lack of a reference collection was one of the main problems affecting research in the field of spam detection. This often obliged researchers to build their own data sets to perform experiments, with a twofold drawback. First of all, the data sets were generated to constitute a good representative of the phenomenon researchers were investigating and so, in many cases, had been biased towards it. Second and more importantly, techniques cannot be truly compared unless they are tested on the same collection.

The *webspam-uk2006* dataset described in [18] and available on-line[6] is a large, publicly available collection for Web spam research. It is based on a large crawl of Web pages downloaded in May 2006 by the Laboratory of Web Algorithmics, University of Milan.[7] The crawl was obtained from the `.UK` domain, starting from a set of hosts listed in the Open Directory Project and following links recursively in breadth-first mode.

The labelling was the result of a collaborative effort. A group of volunteers was shown a list of Web sites — the "host" part of the URLs — and asked for each host, if there were *spamming aspects* in the host. A list of typical spamming aspects were available to guide the assessment. Some of the aspects often found in spam hosts were: large sets of keywords in the URL and/or the anchor text of links, multiple sponsored

[6] http://www.yr-bcn.es/webspam

[7] http://law.dsi.unimi.it/

links and/or ad units, plus text copied from search engine results. Eventually, the corpus[8] contained 8123 hosts tagged as *normal*, 2113 hosts tagged as *spam* and 426 tagged as *undecided* (borderline).

4.2. The Web Spam Challenge

Using this collection, the Web Spam Challenge series was started.[9] Two challenges were ran during 2007. The first Web Spam Challenge took place simultaneously with AIRWeb 2007[10]; six teams participated and were given a graph, the contents of a sample of 400 pages for each host, and a list of features described in [16,17]. Participants were allowed (and encouraged) to compute their own features and build classifiers that were later tested on a test set obtained from the same collection.

The second Web Spam Challenge took place during GraphLab 2007.[11] This second challenge was aimed mostly at machine learning research groups. Six teams participated (2 that also had participated in the first challenge) and were given just the graph and a set of features. Participants were not allowed to use any external source of information.

The set of features used in the first Web Spam Challenge was composed of 236 features. These features included content-based features such as average word length, number of words in the title, content diversity, term popularity and others proposed in [16]; as well as link-based features such as PageRank, number of neighbours, and others proposed in [17].

4.3. Web spam and machine learning

It has been observed that the distribution of statistical properties of Web pages can be used for separating spam and non-spam pages. In fact, in a number of these distributions, outlier values are associated with Web spam [19]. Several research articles in the last years have successfully applied the machine learning approach to Web spam detection [16,20,21,4].

Building a Web spam classifier differs from building an e-mail spam classifier in a very important aspect: aside from statistical properties from the contents of the messages/pages, we also have a directed graph on the data. Furthermore, there are linking patterns that can be observed in this graph: for instance, non-spam hosts rarely link to spam hosts, even though spam hosts do link to non-spam hosts.

In the scientific literature, there are several ways in which this Web graph has been exploited for Web spam detection.

A first option is to analyse the topological relationship (e.g., distance, co-citation, etc.) between the Web pages and a set of pages for which labels are known [22,23].

A special group of Web-graph topology-based techniques, which deserves for a separate discussion, is based on a notion of *trust* which originates from the social network analysis. This topis is discussed in a subsection 4.4.

Another option is to extract link-based metrics for each node and use these as features in a standard (non-graphical) classification algorithm [17]. Finally, it has been

[8] Counts of `webspam-uk2006-set1-labels.txt` and `webspam-uk2006-set2-labels.txt` combined.

[9] http://webspam.lip6.fr/

[10] http://airweb.cse.lehigh.edu/2007/

[11] http://graphlab.lip6.fr/

shown that the link-based information can be used to refine the results of a base classifier by perturbing the predictions done by the initial classifier using propagation through the graph of hyperlinks, or a stacked classifier [5,24].

4.4. The Concept of Trust in Web Spam Detection

Among the best features used in the machine-learning approach to Web spam classification are those based on the notion of *trust* or *distrust*. The concept is widely known in the social-network research community. A general survey of the trust management techniques can be found in [25].

Due to the adversarial nature of the Web, making use of the concept of trust or distrust when assessing the quality of linked Web pages proved to be a successful idea. In particular, it concerns automatic identification of the Web spam documents.

In the context of directed graphs representing virtual social networks (similar to that of the linked Web pages), a systematic approach for computing or *propagating* the trust through the edges of the graph is discussed in [26]. Various schemes for trust and distrust propagation, which are mathematically represented by the properly modified adjacency matrices and some multiplicative operations are proposed and experimentally studied with the use of some real datasets concerning virtual communities.

While in social networks the concept of trust concerns the users of the system, and models the degree of belief about the honesty of other users, in the context of the Web, the idea is slightly different. Namely, the link between two pages p and q is simplifically interpreted as the belief of the author of the page p about the *good quality* of the page q. An alternative approach, however, was proposed in [27], where an extended linking language is proposed with some experiments done with the use of the `Epinions.com` dataset. The latter approach proposes to distinguish between the "appreciating" and "criticising" links between the pages by a proper extension of the markup language.

One of the first works concerning the application of the notion of trust in successful automatic identification of Web spam documents is [28]. The paper proposes an algorithm called "TrustRank" which uses a seed set of some "trusted" pages (which practically mean the pages labelled by human experts as non-spam pages) and the trust propagation algorithm derived from the classic PageRank [12] algorithm. The idea is based on the observation that non-spam pages usually link to other non-spam pages. Noteworthy, the values computed by the TrustRank algorithm (or derived from them) are found to be among the best attributes used in the machine-learning approach to Web spam classification.

The extension of the ideas discussed in [26] and [28] concerning various methods of trust and distrust propagation in the context of Web spam detection is presented in [29]. In particular, the paper proposes novel methods for splitting trust and distrust through the links as well as for aggregating the incoming values.

The "Topical TrustRank" algorithm is proposed in [30]. It overcomes two vulnerabilities of the TrustRank algorithm [28]: its bias towards more tightly-knit Web pages in the Web graph and the problem of the usual under-representation of the various categories of Web document in the human labelled, trusted seed set. The experimental results in that paper prove that introducing the topical context into the trust-computation framework significantly improves the original TrustRank's idea.

An interesting transformation of the TrustRank algorithm, which propagates the "trust" forward, through the links between "non-spam" pages is presented in [31].

Namely, the idea is similar but inverted here. The proposed algorithm, named AntiTrustRank, is based on the analogous observation: spam pages are usually *linked by* other spam pages in the Web graph. Thus, the algorithm proposes to propagate distrust backward, through links incoming to initially labeled spam pages. The experimental evaluation [31] proves that such approach outperforms that of the TrustRank algorithm.

5. Experiments

This section reports on our explorations of deploying linguistic features for Web spam classification using a machine learning paradigm. Further, we investigate issues concerning unbalanced training class sizes and we analyze the learned decision trees.

5.1. Questions and goals

We outline the questions and goals driving the experiments presented here. Many of these questions arose as a consequence of previous research on the subject — the *webspam 2006 challenge* and [5].

1. Linguistic text features (lexical diversity, emotiveness; more details in the next section) provide very different class of information compared to graph and traditional content features. They should be good discriminators of "real", human-written content and automatically generated (or structured) gibberish. If we add linguistic features to the set of input attributes, will they help to improve the classification accuracy? What is the distribution and relationship between certain linguistic features vs. spam-normal classes?

2. A number of hosts and pages in the *webspam-uk2006* corpus are marked as "borderline" or received an inconsistent note from human evaluators. We suspect that training a classifier on this "noisy" data can mislead the learning algorithm, resulting in poorer performance and proliferation of attributes which are not truly relevant to evident spam hosts. Would initial pruning of the training data (by selecting "strong" examples of non-spam and spam hosts) improve the classification results? What will happen to the size of the resulting decision trees?

3. The two classes of Web sites (spam and normal) are highly unbalanced in size. In [5] authors use a cost-sensitive classifier to cater for this problem, suggesting that cost coefficient R equal to 20 worked best in their case.[12] How sensitive is the classification depending on the actual setting of R? Given the same input data, is $R = 20$ really the best value to pick and why?

To address the above questions we decided to perform several new experiments using the training and test data obtained from the *webspam-uk2006* corpus. Using this particular reference data also lets us compare against the results reported in [5].

The remaining sections describe the arrangement and results of each experiment.

[12] Cost-sensitive classifiers take into account the minimum expected misclassification cost. In our case the cost coefficient R is the cost given to the spam class, and the cost of the normal class is fixed to 1.

Table 1. Selected linguistic features used in our experiments. The "number of potential word forms" used for computing lexical validity and text-like fraction of the text refers to the number of tokens which undergo morphological analysis — tokens representing numbers, URLs, punctuation signs and non-letter symbols are not counted as potential word forms. The term "number of tokens which constitute valid word forms" refers to the number of potential word forms, which actually are valid word forms in the language, i.e., they are recognized by the morphological analyser as such word forms.

feature name	formula	value range
Lexical diversity	$= \frac{\text{number of different tokens}}{\text{total number of tokens}}$	$[0, 1]$
Lexical validity	$= \frac{\text{number of tokens which constitute valid word forms}}{\text{total number of potential word forms}}$	$[0, 1]$
Text-like fraction	$= \frac{\text{total number of potential word forms}}{\text{total number of tokens}}$	$[0, 1]$
Emotiveness	$= \frac{\text{number of adjectives and adverbs}}{\text{number of nouns and verbs}}$	$[0, \infty]$
Self referencing	$= \frac{\text{number of 1st-person pronouns}}{\text{total number of pronouns}}$	$[0, 1]$
Passive voice	$= \frac{\text{number of verb phrases in passive voice}}{\text{total number of verb phrases}}$	$[0, 1]$

5.2. Linguistic features

There is a number of aspects that can be measured and extracted from the text apart from simple occurrence statistics. Certain language features, such as expressivity, positive affect, informality, uncertainty, non-immediacy, complexity, diversity and emotional consistency (discussed in [32]), turned out to have some discriminatory potential for human deception detection in text-based communication. Intuitively, they might also be useful in differentiating Web spam from legitimate content and, to our best knowledge, so far they have not been exploited in this context.

There are various ways of how the aforementioned features can be computed. For instance, for estimating a text's complexity, the average sentence length or the average number of clauses per sentence could be considered. In case of expressiveness, one could give a preference for certain part-of-speech categories to others (e.g., giving higher weight to adjectives and adverbs). Further, non-immediacy is indicated by usage of passive voice and generalising terms.

For our experiments, we have selected and adapted a subset of feature definitions described in [32]. In particular, we considered only features, whose computation can be done efficiently and does not involve much linguistic sophistication since the open and unrestricted nature of texts on the Web indicates that utilization of any more error-prone higher-level linguistic tools would introduce more noise. Table 1 lists the features and formula's used to calculate their value.

Two NLP tools were used to compute linguistic features: *Corleone* (Core Linguistic Entity Extraction) [33], developed at the Joint Research Centre, and Alias-i's *LingPipe*.[13] We processed only the summary version of the *webspam-uk2006* collection. It contains circa 400 pages for each host. It is important to note that solely the body of each page was taken into account. The aggregate for a host was calculated as an arithmetical average

[13] http://www.alias-i.com/lingpipe

Table 2. Results of classification with and without linguistic features on a full data set (all instances) and on a data set from which instances with unknown attribute values have been removed.

	full data		data w/o missing values	
	with l.f.	without l.f.	with l.f.	without l.f.
instances	8 411	8 411	6 644	6 644
attributes	287	280	287	280
classified ok	91.14%	91.39%	90.54%	90.44%
misclassified	8.85%	8.60%	9.45%	9.55%

of values of all its pages. Interestingly, it turned out that 14.36% of the pages had no "textual" content at all and many pages simply indicated HTTP errors encountered during the crawl (404, page not found).

Classification with linguistic features

In our first experiment, we have tested the usability of linguistic features simply by adding them to the set of existing features in the *webspam-uk2006* collection. Surprisingly, adding these new features did not yield significantly different results (see Table 2). In case of the full data set, adding linguistic features degraded classification accuracy slightly. We were able to get a small improvement in quality by pruning the data set from instances with empty values of attributes (see Table 2), but the improvement is very little.

The first intuitive conclusion was that the new features are not good discriminators or there is some strong correspondence between them and the "original" content-based features included in *webspam-uk2006* collection (e.g., compression ratio is in a way similar to lexical diversity). We decided to take a closer look at the distribution of linguistic features with regard to the input classes.

Distribution of linguistic features in the data set

To get a better understanding of our previous results and the relationship between spam and linguistic features, we explored the distribution of those features in the corpus. Figure 1 depicts the distribution of lexical diversity, lexical validity, text-like fraction, emotiveness, self-referencing, and passive voice respectively.

Each diagram in Figure 1 on the following page consists of a bar graph and a line graph. The bar graph reflects the distribution of a given feature in the input corpus of Web pages. The horizontal axis represents a set of feature value ranges (bins). For example, in the first diagram on the left, the first range holds the pages, whose lexical diversity is between 0.0 and 0.05. The values on the left vertical axis correspond to the fraction of pages that fell into a particular range. The right vertical axis corresponds to the graph line, and represents the fraction of pages in each range that were classified as spam.

As can be observed, not all of the features seem to be good discriminators in spam detection. In particular, emotiveness and self referencing do not seem to be good indicators of spam, i.e., the line graph appears to be quite noisy. Certain value ranges for lexical diversity (0.65–0.80) and passive voice (0.25–0.35) might constitute a weak indication of non-spam. The spam-percentage line for lexical validity seems to have a clear

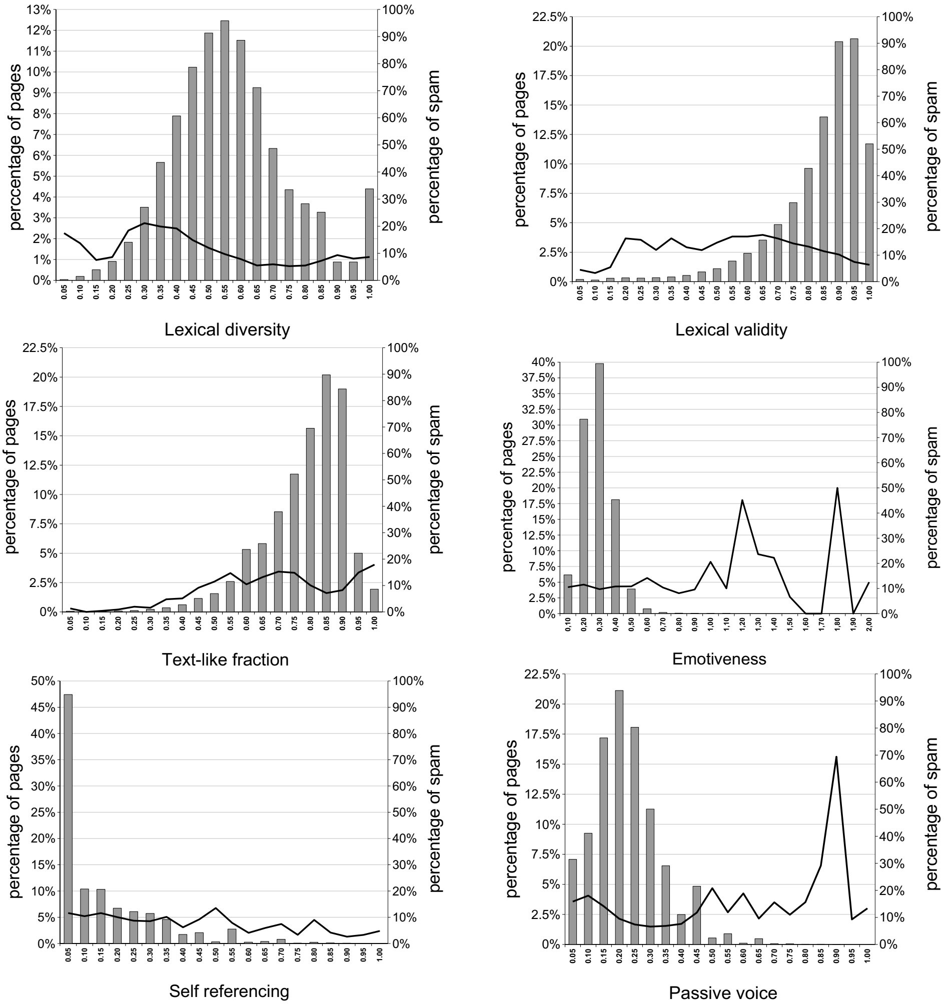


Figure 1. Prevalence of spam relative to linguistic features. The bar graph in each diagram reflects the distribution of a given feature in the input corpus of Web pages. The horizontal axis represents a set of feature value ranges (bins). The values on the left vertical axis correspond to the fraction of pages that fell into a particular range. The right vertical axis corresponds to the graph line, and represents the fraction of pages in each range that were classified as spam.

downward trend in the rightmost part of the corresponding diagram. In case of text-like fraction feature, values below 0.3 correlate with low spam probability.

Since many of the pages contained in the "summary" collection happen to be just short messages indicating HTTP errors, we recalculated the distributions discarding all pages with less than 100 tokens. Figure 2 depicts the recomputed distribution for text-like fraction and lexical validity. Some improvement can be observed: left and right-boundary values for lexical validity, as well as text-like fraction values lower than 0.25, correlate with higher probability of non-spam, whereas text fraction of more than 95% implies higher prevalance of spam (50%). However, the assessment of the usefulness of each feature (the line) should take into account the number of documents in particular range (the bar).

For the sake of completeness, we also provide in figure 3 direct comparison of histograms for the latter attributes in spam and non-spam pages in the reduced corpus.

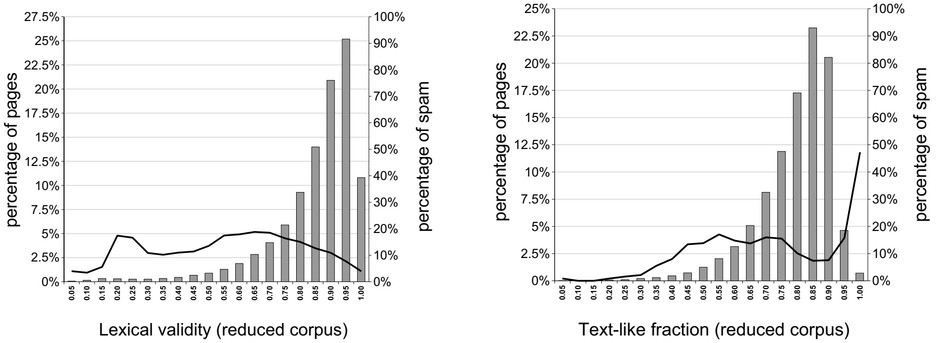


Figure 2. Prevalence of spam relative to lexical validity and text-like fraction of the page in the reduced input corpus.

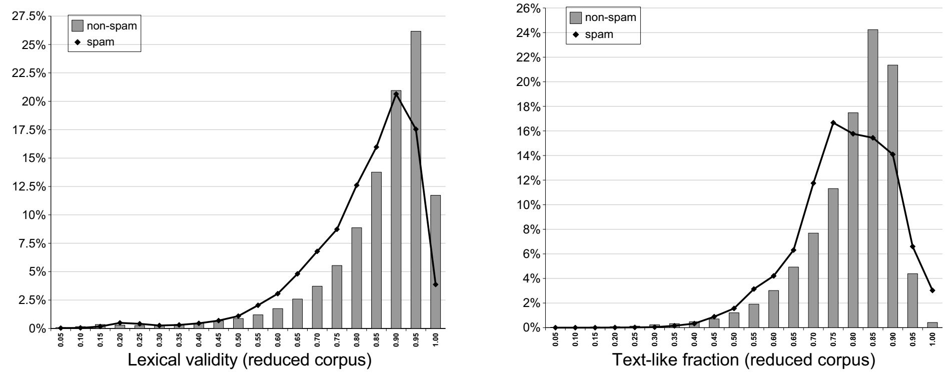


Figure 3. Histograms of the lexical validity and text-like fraction in spam and non-spam pages in the reduced input corpus.

The last experiment shows that a more-sophisticated way of computing some of the linguistic features might be beneficial. In general, however, the linguistic features explored in this article seem to have less discriminative power than the content-based features described in previous work on the subject [34]. This may be a result of the fact that spammers reuse some existing Web content (inject spam content inside legitimate content crawled from the Web).

5.3. Looking for the optimum cost value of the cost classifier

Inspired by the results reported in [5], we wanted to shed some more light on the characteristic of the cost ratio (R) between spam and normal classes, given to the cost-sensitive classifier (see footnote 3 on page 9).

Table 3. The number of hosts that received an identical number of votes for the "pure" and "purest" sets. The denotations NN, NNN, NNNN mean that a host received a univocal "normal" label from (all) 2,3 or 4 human assessors (respectively). Similarly, the SS denotation concerns the host that obtained "spam" label from both the human assessors.

data set	votes			
	NNNN	NNN	NN	SS
pure	13	843	1850	323
purest	13	843	—	323

We trained and tested the cost-sensitive classifier (with underlying J48 algorithm) for R values ranging between 1 and 200. For each value of R, a single training/ testing round was executed until the resulting decision tree does not depend on the order of input data or other parameters.

As it turned out, adjusting the cost of misclassifying a spam page as normal (R) does not affect the f-measure as much as one could think (see Figure 4 on page 16). Increasing R beyond 70 does not change the results significantly at all. True positive/ false positive ratio curves are more insightful compared to f-measure — it seems sensible to strike the balance between TP (true positive) and FP (false positive) ratios of spam and normal classes and this happens for value of R somewhere around 20, just as previously reported in [5].

5.4. Classification accuracy for classifiers trained on "clean" data

In this experiment we start from the assumption that the label assigned to training examples (spam/ normal) is not always correct. This was motivated by the analysis of the training data — even though the labels were assigned by humans, there were frequent cases of inconsistent labels between judges, most likely caused by pages or hosts that mixed legitimate content with spam [18]. Instead of training the classifier on this "borderline" data, we decided to extract just the strongest examples of spam and normal hosts and use this subset for learning.

We processed the "judgement" files from the *webspam-uk2006* collection, splitting hosts into subsets that exhibited full agreement of judges. For each host we concatenated all votes it received so, for example, a host marked with `SS` received two "spam" votes, a `NNN` host received three "normal" votes and so on. We then created two sets — "pure" and "purest", consisting of hosts with the following labels:

- `NNNN`, `NNN`, `NN`, `SS` hosts ("pure" set),
- `NNNN`, `NNN`, `SS` hosts ("purest" set).

The number of the hosts in each group is given in Table 3.

Finally, we trained a cost-sensitive classifier on each of these filtered sets, for changing cost value R — this time between 1 and 40. The resulting decision trees were evaluated against the original set of hosts (including those that received mixed votes) to keep the results consistent and comparable with previous experiments.

Figure 5 illustrates the F-measure, area under curve (AUC), true positive (TP) and false positive (FP) ratios for three training data sets — pure, purest and the original unfiltered set. Before we compare the results note that, regardless of the data set, the "optimal" value of the cost R seems to be around the value of 20 — this is where TP/FP meet and the F-measure/ AUC reach their peak values. As for pruning the training data, we can observe a slight boost of quality (F-measure, AUC) for the "pure" set. However, further pruning ("purest" input) does not yield any improvement, even degrades the performance of the final decision tree (note high values in the sub-figure showing true positives).

Summing up, removing the noisy borderline elements from the training data contributes slightly to the accuracy of the final classifier, although leaving out just the strongest examples results in borderline cases to be classified as spam. Not depicted in Figure 5, but of interest and relevance, is the size of the final decision tree, discussed in the next section.

5.5. Analysis of the output decision trees

Figure 6 shows the size of the tree and the number of attributes used for 3 different inputs. We may see that the cleaner the input data, the fewer attributes are needed to properly distinguish between spam and non-spam. Taking into account the fact that there was little difference in quality between the decision tree trained on all instances compared to the "pruned" set, this may mean redundancy of some attributes in the original tree.

We performed the following analysis. For each data set (unfiltered, pure, purest) and for each value of R between 1 and 40, we counted the attributes occurring in the final decision tree (conditions on branches). We then calculated which attributes were used most frequently to decide between a spam host and a regular host. Among the most influential attributes[14], regardless of the value of R, were:

- logarithm of the number of different supporters (different sites) at distance 4 from the site's home page,
- logarithm of the trust rank of a given host's home page,
- length of host name,
- top 100 corpus recall, fraction of popular terms that appeared on the page (*STD_83*),
- top 100 corpus precision, fraction of words in a page that appeared in the set of popular terms (*STD_79*),
- compound features such as *log_OP_trustrank_hp_div_indegree_hp_CP* (various coefficients such as trust rank, in degree etc., combined into a single formula).

Actual conditions on these attributes were quite sensible; for example, if length of the host name exceeds 20 characters or the page contains many popular terms (*STD_83*), then it is most likely a spam host.

Note that these attributes were not only the most frequently used for choosing between spam and normal hosts, but were also stable with respect to the change of cost parameter R (as visually depicted in Figure 7).

[14] See [5] for details concerning how these attributes were computed.

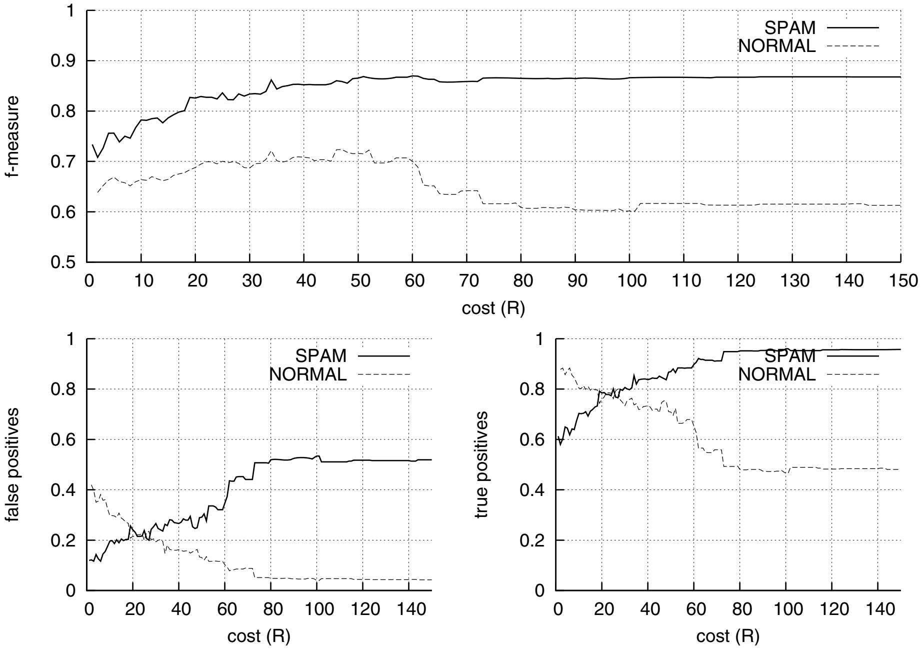


Figure 4. F-measure and TP/FP rates for changing misclassification cost R (continuous lines for clarity).

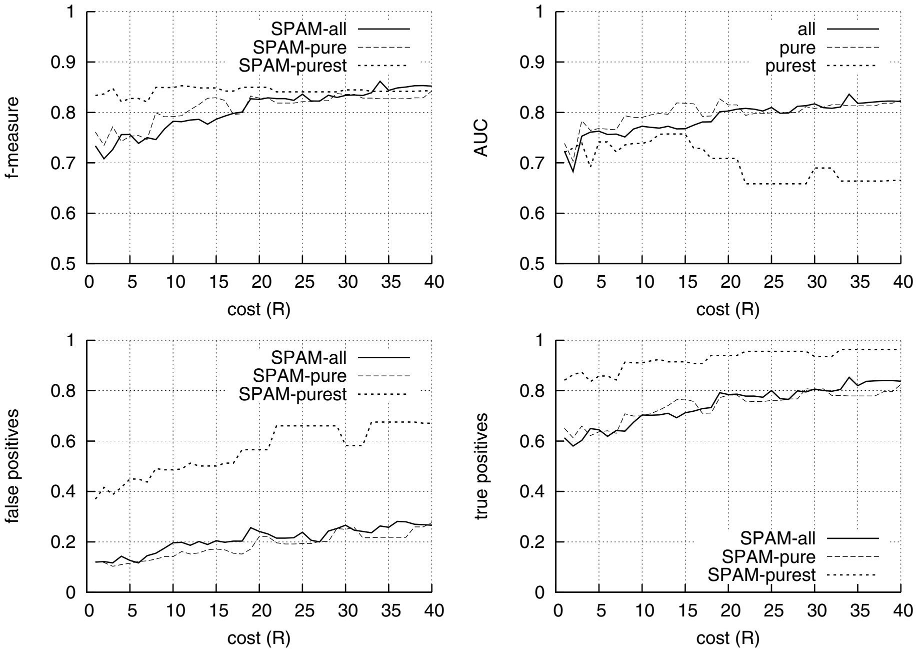
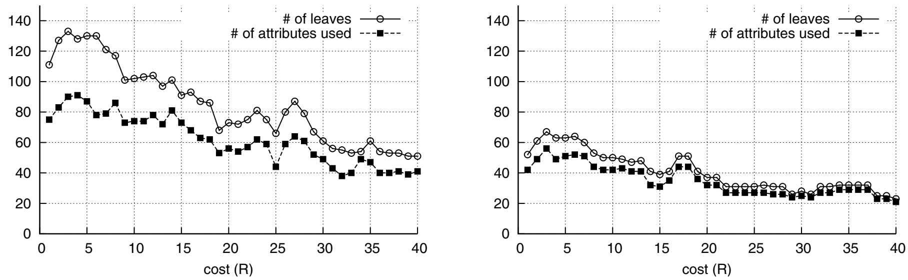


Figure 5. F-measure, TP/FP rates and AUC for changing misclassification cost R, results shown only for the spam class, all three inputs on one chart: pure, purest and unfiltered training set.



a) unfiltered training data

b) “pure” set of hosts

Figure 6. Size of the decision tree (number of leaf nodes and attributes used) depending on the training set.

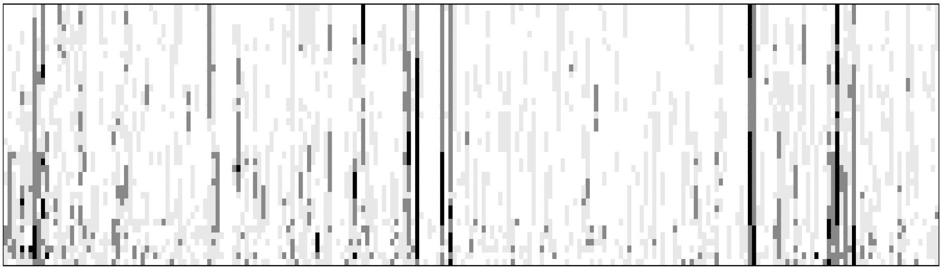


Figure 7. Visualisation of attributes used inside decision trees. Each “column” in the horizontal axis represents a single attribute, vertical axis reflects changing values of R — 1 on the bottom, 40 on the top. A grey square at each intersection indicates the number of times the attribute was used, aggregating over the three training sets; light grey: 1 time, grey: 2 times, black: 3 times. Note nearly solid black vertical lines — these attributes were almost always used for prediction.

6. Summary and conclusions

Web spam is any form of manipulation of Web documents intended to mislead ranking algorithms of search engines in order to artificially boost the ranking position without improving the information quality for Web users. Fighting Web spam is being considered as one of the most urgent problems in the Web information dissemination process since it significantly deteriorates the quality of search results and thus affects the whole Web community. In this article, we gave a short overview of the Web spam phenomenon and state-of-the-art techniques for combating it. Further, we tried to answer and verify several open questions by applying machine learning techniques.

First, we explored whether linguistic features, which go beyond classical content-based features (used by others), have any discriminatory power for classifying spam. In particular, we experimented with features like lexical validity, lexical diversity, emotiveness, text-like fraction, passive voice and self reference, which proved to be useful in the process of detecting human deception in text-based communication [32]. Various experiments on including these features for training a classifier did not show any significant improvement in the accuracy, although some of the corresponding distribution graphs revealed some discriminatory potential.

Our second endeavour focused on experimenting with training the classifier on “cleaned” data, i.e., data pruned via removing the “borderline”, which were neither classified as spam nor legitimate pages, and non-univocally labelled instances in the training corpus. Removing such noisy data yielded significantly simpler decision trees without deteriorating the classification accuracy. Presumably some attributes in the trees computed from the original data were redundant. We also observed that there were some attributes which were most influential disregarding the cost coefficient and training dataset used. These included: logarithm of the trust rank of hosts home page, length of host name, logarithm of the number of different supporters, top-100 corpus recall, top-100 corpus precision and some compound features like trustrank combined with indegree.

A continuation of the application of light-weight linguistic analysis in machine learning approach to Web spam detection is envisaged. Most likely, the linguistic features studied in our work duplicate information of the traditional content-based features. In the next step, we intend to train the classifier solely using linguistic features in order to verify the latter assumption. Further, we also intend to explore more sophisticated features like for instance positive affect, syntactical diversity, etc.

The current and future results of our work related to the application of linguistic features for web spam detection will be available at the following URL:
`http://www.pjwstk.edu.pl/~msyd/lingSpamFeatures.html`

7. Acknowledgements

The work was additionally supported by the EMM project carried out at the Joint Research Centre of the European Commission (`http://emm.jrc.it/overview.html`) and by the internal grant of Polish-Japanese Institute of Information Technology: ST/SI/06/2007.

References

[1] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 517–528, Trondheim, Norway, 2005.

[2] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, pages 39–47, Chiba, Japan, 2005.

[3] Monika R. Henzinger, Rajeev Motwani, and Craig Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[4] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Boston, MA, USA, July 2006.

[5] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of SIGIR*, Amsterdam, Netherlands, July 2007. ACM.

[6] Ziv Bar-Yossef and Maxim Gurevich. Efficient search engine measurements. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 401–410, New York, NY, USA, 2007. ACM.

[7] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. Searching the web. *ACM Transactions on Internet Technology (TOIT)*, 1(1):2–43, 2001.

[8] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[9] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data.* Morgan-Kauffman, 2002.

[10] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[11] Gerard Salton and Michael McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, 1984.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[13] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[14] Ian H. Witten, Marco Gori, and Teresa Numerico. *Web Dragons: Inside the Myths of Search Engine Technology (The Morgan Kaufmann Series in Multimedia and Information Systems).* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[15] Brian D. Davison. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, pages 23–28, Austin, Texas, USA, July 2000. AAAI Press.

[16] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the World Wide Web conference*, pages 83–92, Edinburgh, Scotland, May 2006.

[17] Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Link-based characterization and detection of Web Spam. In *Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Seattle, USA, August 2006.

[18] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[19] Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the seventh workshop on the Web and databases (WebDB)*, pages 1–6, Paris, France, June 2004.

[20] Tanguy Urvoy, Thomas Lavergne, and P. Filoche. Tracking web spam with hidden style similarity. In *Second International Workshop on Adversarial Information Retrieval on the Web*, Seattle, Washington, USA, August 2006.

[21] Gilad Mishne, David Carmel, and Ronny Lempel. Blocking blog spam with language model disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, May 2005.

[22] András Benczúr, Károly Csalogány, and Tamás Sarlós. Link-based similarity search to fight web spam. In *Adversarial Information Retrieval on the Web (AIRWEB)*, Seattle, Washington, USA, 2006.

[23] Dengyong Zhou, Christopher J. C. Burges, and Tao Tao. Transductive link spam detection. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 21–28, New York, NY, USA, 2007. ACM Press.

[24] Qingqing Gan and Torsten Suel. Improving web spam classifiers using link structure. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 17–20, New York, NY, USA, 2007. ACM.

[25] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the iTrust 3rd International Conference on Trust Management, 23–26, May, 2005, Rocquencourt, France*, pages 77–92. Springer-Verlag, LNCS 3477/2005, May 2005.

[26] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.

[27] Paolo Massa and Conor Hayes. Page-rerank: Using trusted links to re-rank authority. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 614–617, Washington, DC, USA, 2005. IEEE Computer Society.

[28] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating Web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 576–587, Toronto, Canada, August 2004. Morgan Kaufmann.

[29] Baoning Wu, Vinay Goel, and Brian D. Davison. Propagating trust and distrust to demote web spam. In *Workshop on Models of Trust for the Web*, Edinburgh, Scotland, May 2006.

[30] Baoning Wu, Vinay Goel, and Brian D. Davison. Topical trustrank: using topicality to combat web spam. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 63–72, New York, NY, USA, 2006. ACM.

[31] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *ACM SIGIR workshop on Adversarial Information Retrieval on the Web*, 2006.

[32] A. Zhou, J. Burgoon, J. Nunamaker, and D. Twitchell. Automating Linguistics-Based Cues for Detect-

ing Deception of Text-based Asynchronous Computer-Mediated Communication. *Group Decision and Negotiations*, 12:81–106, 2004.

[33] Jakub Piskorski. Corleone - Core Language Entity Extraction. Technical Report (in progress). Joint Research Center of the European Commission, 2008.

[34] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of WWW 2006, Edinburgh, Scotland*, pages 83–92, 2006.

This page intentionally left blank

Social Networks

This page intentionally left blank

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-157

Emergent patterns in online coactivity

Dennis M. Wilkinson [1]
HP Labs, Palo Alto, CA, USA

Abstract. The Internet has enabled people to coactively create, share, rate and classify content on an unprecedented scale. Examples of coactive systems include open source software development, wiki communities, social news aggregators, and many others. This paper discusses regularities in online coactive systems of thousands or millions of participants. First, user participation levels are shown to follow a heavy-tail power-law distribution over their entire range, so that a small number very active users make the vast majority of contributions. The power law arises from a simple rule where the probability a person stops contributing varies inversely with the number of contributions. The power law exponent is moreover demonstrated to discriminate between systems according to the effort required to contribute. Next, the level of activity per topic is shown to follow a heavy-tailed distribution, this time lognormal, generated by a simple stochastic popularity reinforcement mechanism. The vast majority of activity thus occurs among a small number of very popular topics. The trends are demonstrated to hold for four large independent online communities with different scopes and purposes.

Keywords. online data, social systems, empirical studies

Introduction

The Internet provides a unique forum for interaction on a very large scale. The past decade has seen the emergence of coactive online efforts in which a large amount of content is created, shared, promoted, and classified by the interrelated actions of a large number of users. Examples include open source software development, collections of wikis (web pages users can edit with a browser), social bookmarking services, news aggregators, and many others. Coactive systems now comprise a significant portion of the most popular websites [11] and it is reasonable to assume that they will continue to grow in number, scope, and relevance as Internet use becomes more and more widespread.

Large coactive systems are complex at a microscopic level because there is a high degree of variability in people's decisions to participate and in their reactions to others' contributions. The number of possible interactions is also very large, increasing as the square of the number of participants, and the barrier to interaction online is often lower than in traditional social systems. Nevertheless, as we show, macroscopic regularities can be distinguished given a large enough population and explained in terms of simple individual-level mechanisms. Electronic activity records, being extensive, exhaustive, and easy to analyze, are very valuable for this approach.

[1] 1501 Page Mill Rd, ms 1139, Palo Alto, CA 94022 USA; E-mail:dennis.wilkinson@hp.com.

Beyond providing interesting descriptions of people's behavior, macroscopic regularities in coactive systems are of significant practical relevance. For example, the basic principle of Internet search is that high quality pages can be differentiated by having accumulated far more visibility and reputation, in the form of incoming links [3]. Another example is the popular success of Wikipedia, which is at least partially due to the correlation between greater user participation and higher article quality [19]. It is rather remarkable that coactivity on such a large scale is able to produce successful results; in many offline applications, result quality plateaus or decreases as the number of collaborators increases past a certain level (e.g. [4,7]).

Two key challenges in the study of large social systems are to distinguish between general and system-dependent trends, and to provide an explanation for how the trends come about. Empirical regularities which go beyond one particular system or which arise from simple dynamical rules reflect deeply on people's behavior and may be reasonably extended to similar or future instances. A good example of this is the study social networks, where comparisons of structural properties across a number of disparate networks (e.g. [12]), along with theoretical mechanisms for network formation (e.g. [17]) have combined to provide valuable insight. Other examples include the law of Web surfing [10] and the growth dynamics of the World Wide Web [9].

This paper demonstrates strong macroscopic regularities in four online coactive systems. The systems we examine are Wikipedia, an online encyclopedia anyone with a web browser can edit; Bugzilla, a system for reporting and collaborating to fix errors in large software projects; Digg, a news aggregator where users vote to identify interesting news stories; and Essembly, a forum where users create and vote on politically oriented resolves. While all large, these systems range broadly in scope, size, and purpose, from Wikipedia with its broad range of topics and many millions of contributors down to Essembly which focuses on politics and has twelve thousand members.

I examine two fundamental trends: the distribution of levels of user participation, ranging from dedicated, core contributors to casual or one-time participants; and the distribution of activity per topic, ranging from highly popular and visible to very obscure and rarely viewed. The regularities we observe in these distributions are consistent across the four systems, and I show that both trends are attributable to simple, individual-level rules or mechanisms. This suggests that the trends we observe are quite relevant to the study of coactive participation and collaboration in social systems.

The organization of the paper is as follows. Section 1 describes the social systems we analyze and our data sets. We examine the distribution of participation per user and how it is determined by participation "momentum" and the effort required to contribute in section 2. Section 3 discusses the distribution of activity per topic and the simple generative model for it, and section 4 is the conclusion.

1. Systems and data

The results in this paper were observed in data from four online systems which vary greatly in their scope and purpose. The data sets from these systems are in all cases exhaustive, in the sense of including all users and topics within the specified time frame; in three of the four cases, the time frame extends back to the system's inception. A summary is provided in table 1.

Table 1. Data sets in this paper. "Topics" refers to articles in Wikipedia, bugs in Bugzilla, stories in Digg, and resolves in Essembly. "Contributions" refers to non-robot edits in Wikipedia, comments in Bugzilla, "diggs" or votes in Digg, and votes in Essembly.

System	time span of our data	users	topics	contributions
Wikipedia	6 years, 10 months	5.07 M	1.50 M	50.0 M
Bugzilla	9 years, 7 months	111 k	357 k	3.08 M
Digg	6 months	521 k	1.32 M	31.2 M
Essembly	1 year, 4 months	12.4 k	24.9 k	1.31 M

The disparate focus and scope of the systems analyzed in this paper is of importance to the generality of our results. The difference in scope is demonstrated in the table. As far as focus, Wikipedia is as broad as possible, Bugzilla is quite narrow and esoteric, Digg is rather broad but centers on tech and sensational news, and Essembly is strongly political in nature. It is thus reasonable to assume that the population of contributors to each system represents a different cross section of Internet users.

Wikipedia [2] is the online encyclopedia which any user can edit. It consists of a large number of articles (as of this writing, over 9 million [6]) in wiki format, that is, web pages users can edit using a web browser. There is thus virtually no barrier to contribution for any reader. All previous article versions are cached and users can review these as well as exchange comments on the article's dedicated talkpage. When editing, people are encouraged to follow a code of principles and guidelines, and in the worst cases of misuse, volunteer administrators may step in and ban a particular editor for a short time. Users can locate Wikipedia articles using a search function, and the articles are also hyperlinked together when related terms appear in the text.

Our data set contains user ID, article ID and timestamp for all the edits made to the English language Wikipedia between its inception in January 2001 and November 2, 2006. We processed the data to exclude disambiguation and redirect articles, as well as the 5.2 million edits made by robots, as described in [19]. The numbers are presented in table 1.

Essembly [3] is an open online community where members propose and vote on politically oriented resolves, post comments, and form friendships, alliances and anti-alliances ("nemesis links"). The site's welcome page states that its goal is to allow users to "connect with one another, engage in constructive discussion, and organize to take action," although experience suggests that voting and commenting on resolves is the dominant activity. All the resolves expect for ten "initial" resolves were created by site users. Voting is done on a four point scale ranging from strongly agree to strongly disagree, and one's votes are tallied anonymously and also visible to neighbors in the friends, allies and nemesis networks. Within Essembly, multiple mechanisms exist for users to learn about new resolves, including lists of recent popular or controversial resolves and votes within users' social and preference networks, none which is particularly dominant [8].

Our data set contains randomized user ID, randomized resolve ID and timestamp for all votes cast between Essembly's inception in August 2005 and December

[2] www.wikipedia.org

[3] www.essembly.com

12, 2006. We excluded votes on the ten "initial" resolves on which all users are encouraged to vote upon signing up.

Bugzilla is an online service for reporting errors and collaborating to fix them in software development efforts. Any large software project can have its own bugzilla; our data comes from the Mozilla Bugzilla[4]. (Mozilla is an open-source suite of Internet tools including a web browser, email client, and many others, and is a large project involving many thousands of developers.) Within Bugzilla, each reported bug has its own page where users can post detailed information, examples, patches and fixes, and exchange comments. The comments typically discuss technical matters. A comment almost always accompanies a patch, fix or other resolution. Bugzilla is equipped with a search function to help users find bugs, and lists of related or dependent bugs exist for some bugs.

Our data set contains randomized user ID and bug ID for the 3.08 million comments posted under the first 357,351 reported Mozilla bugs, from April 1998 through November 22, 2006.[5] This data set is arguably exhaustive in some absolute sense among the population of Mozilla programmers.

Digg [6] is a social news aggregator where users nominate and "digg," or vote for, online news stories they find interesting. We will refer to "diggs" as votes. A Digg vote can only be positive, and indicates that the users finds the story interesting. The stories appear in form of a short summary and the URL link. Fifteen popular recent stories appear on the front page, according to a secret algorithm, and beyond this users must use a search function to find stories.

Our data set consists of randomized user IDs, story IDs and timestamps for all the votes cast between Jan. 1 and July 1, 2007. We also present results from a previous study [20] which used a different data set of all the 29,684 front-page stories from 2006.

2. User participation

In every social unit, there is a range in the amount of participation by different members, from a dedicated core group to a periphery of occasional or one-time participants. The distribution of user participation in online social systems is of practical relevance to how these communities evolve. As we show, participation follows a heavy-tail power law distribution in which a small number of very active users account for most of the activity. A heavy tail trend was previously noted in chat room posts [18], but the distribution was not formally studied or extended to other online communities.

In our observations, we measured participation in the following ways. For Wikipedia, we counted the number of edits made by each non-robot user, meaning each time a new version of an article was uploaded more than 10 seconds after the previous edit [7].

[4] https://bugzilla.mozilla.org/

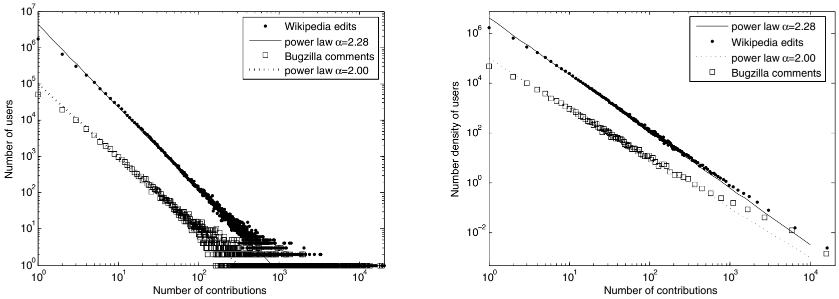
[5] This figure excludes some 3500 bugs which we required special authorization to access, most likely because of security concerns, and include 54 older bugs imported from Netscape bug lists.

[6] www.digg.com

[7] The 10 second cutoff was chosen to exclude edits by users who occasionally functioned as bots, editing hundreds of times in a few seconds. Any actual human edits excluded by this cutoff were not likely to have been significant contributions of content.

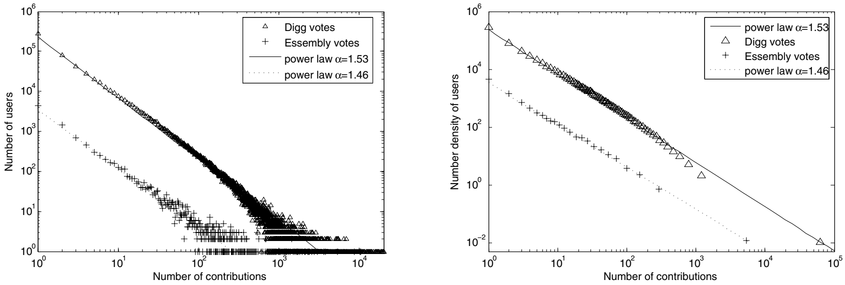
Wikipedia users could and often did submit multiple edits to a single page in one session. In Bugzilla, we counted the number of comments, including those accompanying patches or other resolutions, posted by each user. Users often commented numerous times on a single bug. In Digg and Essembly, we counted the number of votes (in Digg known as “diggs”) each user made. Users were only allowed to vote once per resolve or story. As previously mentioned, there is only one possible Digg vote (saying “yes, that is interesting”), whereas Essembly users vote on a four-point scale.

In counting the number of contributions per user, we include only those users who have become inactive. This provides a useful comparison across systems of different ages and will allow us to draw conclusions about users’ probability of quitting the system and the relation of this probability to the effort required to contribute and the number of previous contributions. In practice, we restricted the analysis of this section to users who were inactive for at least six months (Wikipedia and Bugzilla) or three months (Digg and Essembly) prior to the date of data capture. The smaller interval for Digg and Essembly was used because they have a shorter characteristic time between contribution and because of the shorter time span of these data sets.



(a) Raw data (b) Binned data

Figure 1. Number of Wikipedia edits or Bugzilla comments per user



(a) Raw data (b) Binned data

Figure 2. Number of Digg or Essembly votes per user

The number of edits per Wikipedia contributor and comments per Bugzilla contributor are presented in the log-log plot of figure 1. The left panel shows the raw data, and the right panel shows binned data [8] producing a number density curve, which is equivalent to a probability density function multiplied by the total number of users. Figure 2 is analogous for Digg and Essembly votes per user. A power law line is fit to the data in each case, demonstrating their heavy-tail nature. A log-likelihood G-test [14] supports the hypothesis that the power law is generative for $k \geq 10$ contributions at $p > 0.5$ for all systems except Digg ($p > 0.1$; $k \geq 15$). The deviation at the high end is not statistically significant because of the small number of counts in this range.

The power law's excellent approximation of the true distributions over their entire range warrants closer examination. A power law distribution means that the number of people participating k times is given by

$$N(k) = Ck^{-\alpha}.$$

Alpha is the only parameter and corresponds to the slope of the line in a log-log plot; C is simply a normalization constant. It is rare to see a power law fit a distribution even at the low end of its range. In fact it is common practice to use the term even when only the rightmost tail of the empirical distribution appears straight on a log-log plot, without any statistical justification (for a discussion of this practice and many examples see [5]).

The power law distribution in user participation means that the more people participate, the less likely they are to quit, as we now show. The probability that a user stops after his kth contribution is equal to the number of users contributing exactly k times divided by the number of users contributing k or more times:

$$P(\text{stop after } k) = \frac{Ck^{-\alpha}}{C\sum_{b=0}^{\infty}(k+b)^{-\alpha}} = \frac{1}{\sum_{b=0}^{\infty}(1+b/k)^{-\alpha}}. \tag{1}$$

Observe that in the large k limit,

$$\frac{1}{k}\sum_{b=0}^{\infty}\left(1+\frac{b}{k}\right)^{-\alpha} \longrightarrow \int_0^{\infty}(1+x)^{-\alpha}dx + O(1/k) = \frac{1}{\alpha-1} + O(1/k)$$

where we have used the formal definition of Riemann integration with step size $1/k$. In fact, since the maximum slope of the function $(1+x)^{-\alpha}$ on $(0,\infty)$ is $-\alpha$, the error term is bounded above by $\alpha/2k$ [1]. Returning to equation 1, we have that

$$P(\text{stop after } k) = \frac{\alpha-1}{k} + O(1/k^2) \tag{2}$$

where the error term is bounded above by $\alpha(\alpha-1)^2/2k^2$ and is thus very small for k as small as 5 or 10.

Equation 2 indicates that people have a "momentum" associated with their participation, such that their likelihood of quitting after k of contributions decreases inversely with k. This rule holds for any power law, independent of the value of the exponent.

As for the power law exponent α, we now turn to the question of its interpretation. Larger values of α in equation 2 indicate that, at every opportunity, a contributor is more likely to quit. When the effort required to to contribution is higher, we thus expect a larger value of α. Participation in Wikipedia, as we measure it, requires the user to contribute

[8] We used equal-count binning, where the number of counts per bin in each data set was proportional to the total number of counts for that set

original content with at least some background effort. By contrast, voting in Digg and Essembly can be done quickly with far less personal investment. This quick approach was used by at least some Digg and Essembly users, as evidenced by the rapid accumulation of votes in both systems immediately following the appearance of a resolve or story [16]. We therefore expect to find a higher value of α for Wikipedia than for Digg and Essembly. Bugzilla provides an intermediate case.

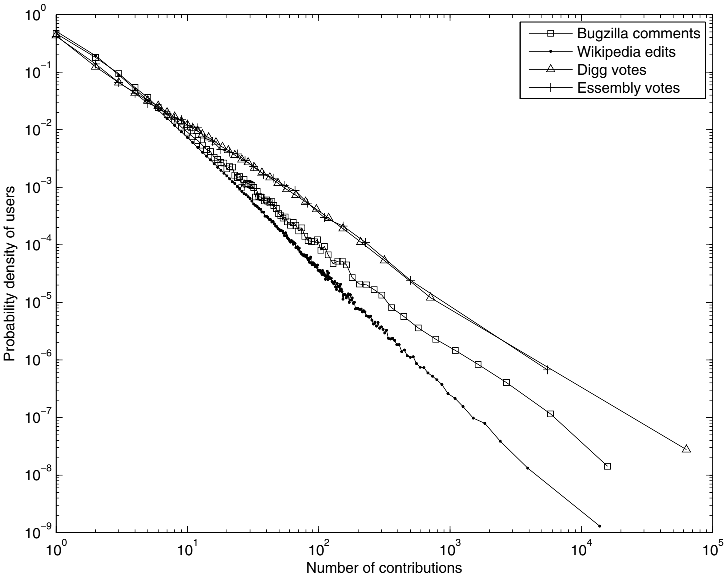


Figure 3. Empirical probability density function for number of contributions per user for all four systems.

This expectation is confirmed by the data, as shown in Figure 3. In this figure, we have produced an empirical probability density function for each system by dividing each user's counts by the total for that system and binning as before.

The similarity between Digg and Essembly in figure 3 is striking. Indeed, similarity of the barrier to participation is one of of the few common properties of these systems. This suggests that the barrier to participation is the dominant element in determining the rate of participation dropoff.

3. Coactive reinforcement of topic popularity

This section examines how interest and participation levels vary across different topics. This subject is of significant practical importance, as demonstrated by the examples of Google search and Wikipedia quality we mentioned in the introduction. Just as for user participation levels, the distribution has a heavy tail of very popular topics which attract a disproportionately large percentage of participation and interest. In this case, however, the exact form is lognormal, not power law.

In this section, contributions are counted as before, and "topics" refers to Wikipedia articles, Essembly resolves, and Digg stories. As a measure of the level of interest of popularity of a topic, the procedure was to simply count the number of contributions to it. For Wikipedia, this metric was shown to correlate strongly to page views [19,15]. The Bugzilla data are not included in this section because the distribution of comments to bugs, while very heavy tail, does not follow the mechanism or distributions described in this section.

This process of how participation accumulates on a given topic is very complex at the level of individual contributions, as discussed in the introduction. However, as we show, we can account for individual idiosyncrasies and the varying effect of many interactions with a simple white noise term, as follows.

Consider the number of new edits to a Wikipedia article, or votes to an Essembly resolve or Digg story, made between time t and time $t + dt$, an interval of minutes or hours. Because of the complicated nature of the system, this number will vary a lot depending on the time period and topic. However, the overall average amount of new activity will be directly related to the visibility or popularity of the topic. We account for the effect of coaction in the system in the simplest possible way, by assuming that contributions to a topic increases its popularity or visibility by some constant amount, on average, with deviations away from the average absorbed into a noise term. The number of contributions to a given topic will thus be proportional to the number of previous contributions, and the dynamics of the system expressed simply as:

$$dn_t = [a + \xi_t]n_t dt. \tag{3}$$

In this equation, n_t is the number of contributions on the topic up until time t; dn_t is the amount of new activity between t and $t + dt$ for some suitably small dt; a is the average rate of activity, independent of the topic or time; and ξ_t is a mean-zero white noise term. The noise term embodies the vagaries of human behavior, the varying effect that one person's contribution has on other people's participation, and the varying effect each contribution has on topic popularity. It might seem too good to be true that the noise term be so simple; it is, but only in a trivial way. That is, we do observe a short autocorrelation length in the noise, which has the cosmetic effect of affecting our measurement of the time scale. However, the general conclusions, including the application of the central limit theorem below in the solution of this stochastic differential equation, still hold [2].

For Wikipedia and Essembly, this equation is sufficient to describe the dynamics. For Digg, it must be modified by introducing a discount factor to account for the decay in novelty of news stories over time [20]. In Digg, the basic equation is thus

$$dn_t = r(t)[a + \xi_t]n_t dt.$$

where $r(t)$ is a monotonically decreasing function of age. Even with the novelty factor, the final distribution of votes per story can be shown to follow a lognormal distribution, but the age dependence of is more complex. It is also important to mention that this mechanism only functions in Digg for stories which are shown on the front page, because the site interface so heavily favors these in terms of visibility.

According to equation 3, the amount of activity $n(t)$ that a topic of age t has accrued may take on a range of values, because of the presence of the noise. The solution to this stochastic differential equation is the probability density function

$$P[n(t)] = \frac{1}{n\sqrt{2\pi}\sqrt{s^2 t}} \exp\left[-\frac{(\log n - at)^2}{2(s^2 t)}\right], \tag{4}$$

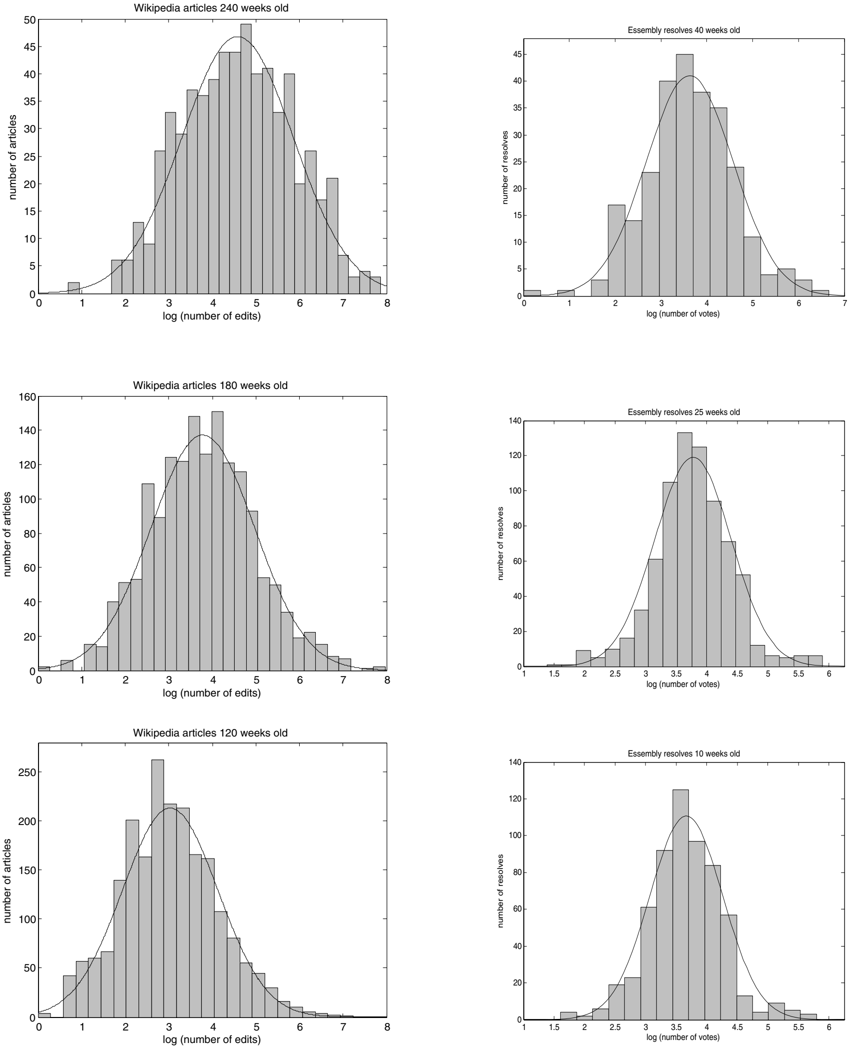
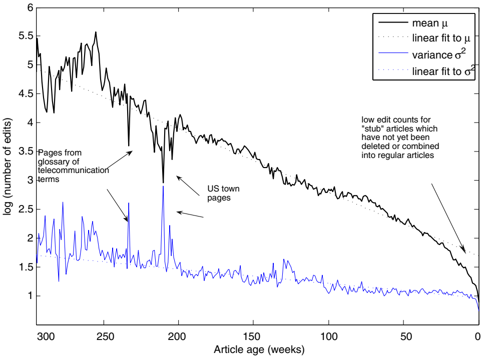


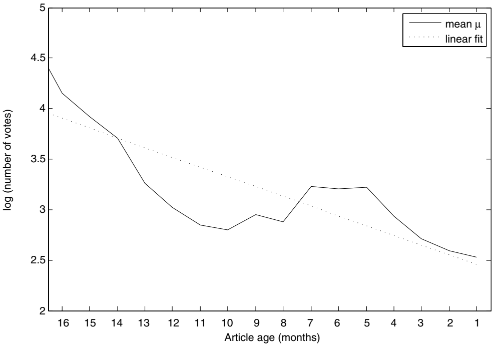
Figure 4. Distributions of the logarithm of the number of Wikipedia edits and Essembly votes for several articles or resolves within several time slices. Since the number of participations is lognormally distributed, the logarithm is normally distributed. The best fit normal curve is included for comparison.

where again s^2 is the variance of the $\xi(t)$ and a is the average rate of accumulation of edits or votes [13]. This equation describes a lognormal distribution with parameters $\mu = at$ and $\sigma^2 = s^2 t$ that depend linearly on the age t of the topic. Note that μ and σ^2 represent the mean and variance, respectively, of the log of the data, and are thus related to but not equal to the distribution mean and variance.

Our model thus predicts that among Wikipedia articles, Essembly resolves, or Digg stories of the same age, the number of edits per article, or votes per resolve or story, will follow a lognormal distribution. These predictions are confirmed by the data. A log-likelihood ratio test on the Wikipedia data shows that 47.8 % of the time slices have a p-value greater than 0.5, for a lognormal distribution with the empirical μ and σ^2. A



(a) Wikipedia



(b) Essembly

Figure 5. Evolution of the parameter μ, the mean of the logarithm of edit or vote counts, for Wikipedia articles and Essembly resolves. For Wikipedia edits, the evolution of the variance σ^2 is also included. The linear best fit line is included for comparison.

similar test for Essembly, shows that 45.6 % of the time slices have a p-value greater than 0.5. In Digg, statistical tests likewise confirmed the lognormal distribution [20]. The lognormal form of the distribution of contributions per topic is demonstrated in figure 4 for several time slices from Wikipedia and Essembly.

The time dependence of the distribution parameters μ and σ^2 with article or resolve age provides another confirmation of the accuracy of equation 4. The linear dependence of μ, which is the mean of the logarithm of participation counts, with topic age in Wikipedia and Essembly is demonstrated in figures 5. The dependence of the variance σ^2 is also included for Wikipedia. For Wikipedia, occasional large deviations from the pattern are noted and explained in the figure. For Essembly, the number of data are not large enough to demonstrate the trend as clearly; compare the sample variability with the first 50 or so weeks of Wikipedia data (the numbers of data points in these time slices are similar). In Digg, as previously mentioned, the time dependence was more complex because of the decay of novelty, but observed values of the parameters μ and σ^2 were found to have the correct time dependence [20].

4. Conclusion

This paper presented a set of observations from four different online social systems: Wikipedia, Digg, Bugzilla and Essembly. These systems each have a different focus and scope, ranging from 12,500 members of the political discussion community Essembly up to more than 5 million contributors to the huge online encyclopedia Wikipedia. I analyzed exhaustive data sets from these four systems to study the distribution of participation levels per person and per topic.

I first showed that user participation levels in all four systems was well described by a power law, in which a few very active users accounted for most of the contributions. The power law form implies that there is a momentum associated with participation such that the probability of quitting is inversely proportional to the number of previous contributions. The power law exponent was shown to correspond to the effort required to contribute, with higher exponents in systems where more effort is required. A striking similarity was observed in the exponent between Wikipedia edits and Bugzilla comments, and between Digg and Essembly votes. This suggests that the user participation distribution is primarily dependent only on the participation momentum rule and the system's barrier to contribution.

I then presented theory and observations of a heavy-tailed lognormal distribution in the number of contributions per topic. The mechanism explains the propensity of a few very visible popular topics to dominate the total activity in coactive systems, and is observed in Wikipedia, Digg and Essembly. It is rather remarkable that the many forms of variation at the individual level of these systems can be accounted for with such a simple stochastic model.

The observed regularities are of practical relevance to the understanding of large coactive systems, describing how the participation is concentrated among a small number of disproportionately dedicated users and popular topics. Because they are governed by simple mechanisms and are consistent across a variety of systems, the regularities provides a useful tool for estimation and comparison of metrics such as the barrier to participation or the rate at which topics accumulate popularity.

This paper also illustrates the importance of large data sets in the study of coactive phenomena. For example, the nearly 25,000 resolves and 12,500 users of Essembly were barely enough to detect the time-dependence in the distribution of topic popularities. Access to electronic records of online activity is thus essential to progress in this area, and it can only be assured if privacy continues to be respected completely as the scientific community has done to date.

Acknowledgments The author thanks Gabor Szabo, Mike Brzozowski, and Travis Kriplean for their help processing the data, Bernardo Huberman and Gabor Szabo for helpful conversations, Chris Chan and Jimmy Kittiyachavalit of Essembly for their help in preparing and providing the Essembly data, and the Digg development team for allowing API access to their data.

References

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1964.

[2] K. N. Berk. A central limit theorem for m-dependent random variables with unbounded m. *Ann. Prob.*, 1(2):352–354, 1973.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

[4] F. Brooks. *The Mythical Man-month*. Addison-Wesley, Reading, Mass., 1975.

[5] A. Clauset, C. Shalizi, and M. E. J. Newman. Power law distributions in empirical data. 2007. preprint, http://arxiv.org/abs/0706.1062.

[6] Wikimedia Foundation. http://meta.wikimedia.org/wiki/List_of_Wikipedias, accessed 11/23/2007.

[7] J. R. Galbraith. *Organizational Design*. Addison-Wesley, Reading, Mass., 1977.

[8] T. Hogg, D. M. Wilkinson, G. Szabo, and M. Brzozowski. Multiple relationship types in online communities and social networks. 2008. to appear in Proc. AAAI Conf. on Social Information Processing, 2008.

[9] B. A. Huberman and L. A. Adamic. Growth dynamics of the World Wide Web. *Nature*, 399:130, 1999.

[10] B. A. Huberman, P. Pirolli, J. E. Pitkow, and R. M. Lukose. Strong regularities in World Wide Web surfing. *Science*, 280(5360):95–97, 1998.

[11] Alexa Internet Inc. http://www.alexa.com/site/ds/top_500, accessed 11/23/2007.

[12] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[13] B. K. Øksendal. *Stochastic Differential Equations: an Introduction with Applications*. Springer, Berlin, 6th edition, 2003.

[14] J. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, CA, 2nd edition, 1995.

[15] A. Spoerri. What is popular on Wikipedia and why? *First Monday*, 12(4), 2007.

[16] G. Szabo and K. Bimpikis. personal communications.

[17] R. Toivonen, J.-P. Onnela, J. Saramäki, Jörkki Hyvönen, and K. Kaski. A model for social networks. *Physica A*, 371(2):851–860, 2006.

[18] S. Whittaker, L. Terveen, W. Hill, and L. Cherny. The dynamics of mass interaction. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 257–264, New York, NY, USA, 1998. ACM.

[19] D. Wilkinson and B. Huberman. Assessing the value of cooperation in Wikipedia. *First Monday*, 12, 2007.

[20] F. Wu and B. Huberman. Novelty and collective attention. *Proc. Natl. Acad. Sci. USA*, 105:17599, 2007.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-169

Diffusion and Cascading Behavior in Networks

Jure Leskovec [a]

[a] *Machine Learning Department,*
Carnegie Mellon University,
Pittsburgh, PA, USA

Abstract. Information cascades are phenomena in which individuals adopt a new action or idea due to influence by others. As such a process spreads through an underlying social network, it can result in widespread adoption overall. Here we consider information cascades in the context of recommendations and information propagation on the blogosphere. In particular, we study the patterns of cascading recommendations that arise in large social networks. We review recent studies of cascading behavior in product recommendation networks, and information diffusion on the blogosphere. Next, we examine theoretical models of information, virus and influence propagation. Last, we present developments on selecting and targeting nodes in networks to maximize the influence or detect cascades and disease/information outbreaks effectively.

Introduction

Diffusion is a process by which information, viruses, ideas and new behavior spread over the network. For example, adoption of a new technology begins on a small scale with a few "early adopters", then more and more people adopt it as they observe friends and neighbors using it. Eventually the adoption of the technology may spread through the social network as an epidemic "infecting" most of the network. As it spreads over the network it creates a cascade. Cascades have been studied for many years by sociologists concerned with the *diffusion of innovation* [33]; more recently, researchers have investigated cascades for selecting trendsetters for viral marketing, finding inoculation targets in epidemiology, and explaining trends in blogosphere.

There are three aspects of studies on diffusion and cascading behavior in networks: (a) mathematical models of information, virus and influence propagation, (b) empirical studies of diffusion in social and information networks, and (c) algorithms for detecting cascades and selecting influential nodes.

(a) Mathematical models

Most of the research on the flow of information and influence through the networks has been done in the context of epidemiology and the spread of diseases over the network [4]. Classical disease propagation models are based on the stages of a disease in a host. The disease is usually defined by two parameters: the infection probability defines how viral (easily spreading) is the disease, and the recover probability defines how likely is a node to get cured from the disease. Given a network a typically studied question is the *epidemic threshold* where we would like to know conditions (disease properties) under which the disease will either die out and how fast will it die out, or whether it will dominate the network, i.e., create an epidemic.

Related are the diffusion models that try to model the process of adoption of an idea or a product. They can generally be divided into two groups:

1. *Threshold model:* [12] A node adopts the behavior (*e.g.*, purchases a product) if a sum of the connection weights of its neighbors that already adopted the behavior is greater than the threshold.
2. *Independent cascade model* [16] where whenever a neighbor v of node u adopts, then node u also adopts with probability $p_{u,v}$, *i.e.*, every time a neighbor of u purchases a product, there is a chance that u will decide to purchase as well.

(b) Empirical studies of cascading behavior

While the above models address the question of how processes spread in a network, they are based on *assumed* rather than *measured* influence effects.

Most work on measuring cascading behavior has been done in the blog domain. Blog posts refer to each other using hyper-links. Since posts are time-stamped, we can trace their linking patterns all the way to the source, and so identify the flow of information from the source post to the followers and followers of the followers [24]. Similarly, viral marketing can be thought of as a diffusion of information about the product and its adoption over the network [22]. Here the cascades are formed by people recommending products to each other and so the product recommendations (and purchases) spread over the network.

In our work [22,24] we observed rich cascading behavior on the blogosphere and in the viral marketing and investigated several interesting questions: What kinds of cascades arise frequently in real life? Are they like trees, stars, or something else? And how do they reflect properties of their underlying network environment? Do certain nodes have specific propagation patterns?

(c) Detecting cascades and finding influential nodes

Exploiting cascades could lead to important insights. For example, in viral marketing where a company wants to use word-of-mouth effects to market a product, exploiting the fact that early adopters may convince their friends to buy the product is crucial. So, the company wants to identify the most important nodes to target to spread the information about the product over the network [16]? A similar problem is of detecting outbreaks in networks [23], where we are given a network and a dynamic process spreading over it, and we want to select a set of nodes to detect the process as effectively as possible. For example, consider a city water distribution network, delivering water to households via pipes and junctions. Contaminants may spread over the network, and so we want to select a few locations (pipe junctions) to install sensors to effectively detect the contaminations.

One can formulate above tasks as optimization over sets of nodes, which turns out to be hard computational problem. However, it turns out that influence functions exhibit a diminishing returns property called *submodularity*. Exploiting submodularity we design near-optimal algorithms [23,16] for finding influential nodes and effectively detecting outbreaks in networks.

The remainder of the chapter is organized as follows. First, we briefly review typical models and natural settings where cascades occur and are measurable in real life. Next, we present empirical observations and measurements of cascading behavior on the blogosphere and in the viral marketing domain where people make product recommendations, which then influence purchases. Last, we examine the problem of selecting most influential set of nodes.

1. Cascades in networks

Information cascades are phenomena in which an action or idea becomes widely adopted due to influence by others [5]. Cascades are also known as "fads" or "resonance." Cascades have been studied for many years by sociologists concerned with the *diffusion of innovation* [33]; more recently, researchers in several fields have investigated cascades for the purpose of selecting trendsetters for viral marketing [9], finding inoculation targets in epidemiology [28], and explaining trends in blogosphere [19]. Despite much empirical work in the social sciences on datasets of moderate size, the difficulty in obtaining data has limited the extent of analysis on very large-scale, complete datasets representing cascades. Later, we look at the patterns of influence in a large-scale, real recommendation network and examine the topological structure of cascades.

Most of the previous research on the flow of information and influence through the networks has been done in the context of epidemiology and the spread of diseases over the network [4,3]. Classical disease propagation models are based on the stages of a disease in a host: a person is first *susceptible* to a disease, then if she is exposed to an infectious contact she can become *infected* and thus *infectious*. After the disease ceases the person is *recovered* or *removed*. Person is then *immune* for some period. The immunity can also wear off and the person becomes again susceptible. Thus SIR (susceptible – infected – recovered) models diseases where a recovered person never again becomes susceptible, while SIRS (SIS, susceptible – infected – (recovered) – susceptible) models population in which recovered host can become susceptible again.

Typically studied problems is that we are given a network, a set of infected nodes and the disease (virus) parameters β and δ, where β is the virus birth rate (probability that infected node with transmit a disease to a neighbor) and the δ is the virus death rate (probability that infected node recovers). Then the typically studied question is the *epidemic threshold* τ, *i.e.*, conditions (values of β and δ) under which the disease will either dominate or die out from the network. Interestingly, the largest eigenvalue of a graph adjacency matrix plays a fundamental role in deciding whether the disease will take over the network [7]. One can prove that there will be no epidemic if $\beta/\delta < \tau = 1/\lambda_{1,A}$, where $\lambda_{1,A}$ is the largest eigenvalue of adjacency matrix A of the network [35].

A parallel line of work focuses on diffusion models that try to model the process of adoption of an idea or a product can generally be divided into two groups:

- *Threshold model* [12] where each node in the network has a threshold $t \in [0, 1]$, typically drawn from some probability distribution. We also assign *connection weights* $w_{u,v}$ on the edges of the network. A node adopts the behavior if a sum

of the connection weights of its neighbors that already adopted the behavior (purchased a product in our case) is greater than the threshold: $t \leq \sum_{\text{adopters}(u)} w_{u,v}$.
- *Independent cascade model* [11] where whenever a neighbor v of node u adopts, then node u also adopts with probability $p_{u,v}$. In other words, every time a neighbor of u purchases a product, there is a chance that u will decide to purchase as well.

While these models address the question of how influence spreads in a network, they are based on *assumed* rather than *measured* influence effects. In contrast, the study presented here tracks the actual diffusion of recommendations through email, allowing us to quantify the importance of factors such as the presence of highly connected individuals, or the effect of receiving recommendations from multiple contacts. Compared to previous empirical studies which tracked the adoption of a single innovation or product, our data encompasses over half a million different products, allowing us to model a product's suitability for viral marketing in terms of both the properties of the network and the product itself.

1.1. Information cascades in blogosphere

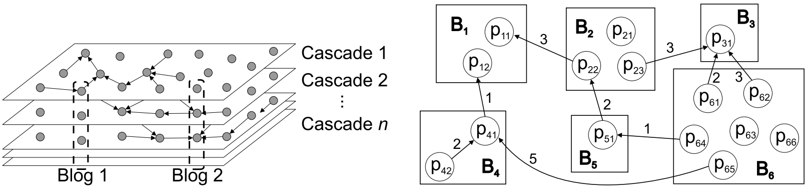
Most work on extracting cascades has been done in the blog domain [1,2,14]. The authors in this domain noted that, while information propagates between blogs, examples of genuine cascading behavior appeared relatively rarely. Studies of blogosphere then either spend a lot of effort mining topics from posts [2,14] or consider only the properties of blogosphere as a graph of unlabeled URLs [1].

There are several potential models to capture the structure of the blogosphere. Work on information diffusion based on topics [14] showed that for some topics, their popularity remains constant in time ("chatter") while for other topics the popularity is more volatile ("spikes"). [19] analyze community-level behavior as inferred from blog-rolls – permanent links between "friend" blogs. In their extension [20] performed analysis of several topological properties of link graphs in communities, finding that much behavior was characterized by "stars".

1.2. Cascades in viral marketing

Viral marketing can be thought of as a diffusion of information about the product and its adoption over the network. Primarily in social sciences there is a long history of research on the influence of social networks on innovation and product diffusion. However, such studies have been typically limited to small networks and typically a single product or service. For example, [6] interviewed the families of students being instructed by three piano teachers, in order to find out the network of referrals. They found that strong ties, those between family or friends, were more likely to be activated for information flow and were also more influential than weak ties [13] between acquaintances.

In the context of the internet, word-of-mouth advertising is not restricted to pairwise or small-group interactions between individuals. Rather, customers can share their experiences and opinions regarding a product with everyone. Quantitative marketing techniques have been proposed [26] to describe product information flow online, and the rating of products and merchants has been shown to effect the likelihood of an item being bought [31,8]. More sophisticated online recommendation systems allow users to



(a) Cascades as layers (b) Cascades as graphs

Figure 1. Two views on the formation of information cascades on the blogosphere. (a) Each layer presents a separate information cascade and posts at same vertical position belong to the same blog. (b) Blogs have posts, and there are time stamped links between the posts. We denote blogs with squares and each blog has multiple posts (denoted with circles). The links point to the sources of information and the cascades grow (information spreads) in the reverse direction of the edges.

rate others' reviews, or directly rate other reviewers to implicitly form a trusted reviewer network that may have very little overlap with a person's actual social circle. [32] used Epinions' trusted reviewer network to construct an algorithm to maximize viral marketing efficiency assuming that individuals' probability of purchasing a product depends on the opinions on the trusted peers in their network. [16] have followed up on the challenge of maximizing viral information spread by evaluating several algorithms given various models of adoption we discuss next.

2. Empirical observations of cascading behavior

We formally define a cascade as a graph where the nodes are agents and a directed edge (i, j, t) indicates that a node i influenced a node j at time t.

Consider three examples of cascade formation and propagation in networks:

- First, we present results on cascades in a large viral marketing network, where people recommend products to each other and we study the spread and success of recommendations over the network.
- Second, we consider the tracking of a large population of blogs over a long period of time and observe the propagation of information between the blogs.
- Third, we present the propagation of infectious water in large real water distribution networks, and ask the question of where to place a limited number of sensors so the disease outbreaks will be detected early.

Blogs (weblogs) are web sites that are updated on a regular basis. Often times individuals use them for online diaries and social networking; other times news sites have blogs for timely stories. Blogs are composed of time-stamped posts, and posts typically link each other, as well as other resources on the Web.

For example, figure 1 shows two alternative views of information cascades that may occur on the blogosphere. In figure 1(a) each circle represents a blog post, and all circles at the same vertical position belong to the same blog. Often blog posts refer to each other using hyper-links. Given that the posts are time-stamped and usually not updated, we can trace their linking patterns all the way to the source. It is easy to identify the flow if information from the source post to the followers and followers of the followers. So, each layer represents a different information cascade (information propagation graph).

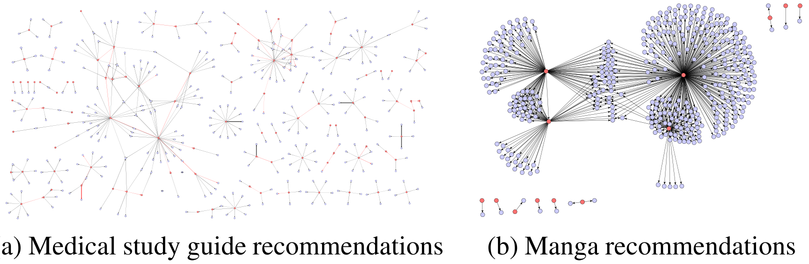

(a) Medical study guide recommendations (b) Manga recommendations

Figure 2. Examples of two product recommendation networks. (a) First aid study guide. Notice many small disconnected cascades. Right: Japanese graphic novel (manga). Notice a large, tight community.

Figure 1(b) gives an alternative view. Here posts (represented as circles) inside a rectangle belong to the same blog. Similarly, the information cascades correspond to connected components of the posts in the graph, *e.g.* posts p_{12}, p_{41}, p_{42} and p_{65} all form a cascade, where p_{12} is the *cascade initiator*.

Observing such behavior on the blogosphere or in the viral marketing poses several interesting questions: What kinds of cascades arise frequently in real life? Are they like trees, stars, or something else? And how do they reflect properties of their underlying network environment? How fast does the information spread? Do certain nodes have specific propagation patterns? What are the most important nodes to target if we want to spread the information over the network?

In addition to observing rich cascades and propagation [25] one can make a step further and analyze the effectiveness and dynamics of product recommendations in causing purchases [21,22]. To our knowledge this was the first study to directly observe the effectiveness of person to person word of mouth advertising for hundreds of thousands of products. Similarly, for blogs [24] is the first to perform a large study of cascading behavior in large blog networks.

2.1. Cascades in viral marketing

A recent study [21] examined a recommendation network consisting of 4 million people who made 16 million recommendations on half a million products from a large on-line retailer. Each time a person purchases a book, music, DVD, or video tape she is given the option to send an email recommending the item to her friends. The first recipient to purchase the item receives a discount and the sender of the recommendation receives a referral credit.

Figure 2 shows two typical product recommendation networks. Most product recommendation networks consist of a large number of small disconnected components where we do not observe cascades. Then there is usually a small number of relatively small components where we observe recommendations propagating. Also notice bursts of recommendations and collisions (figure 2(b)). Some individuals send recommendations to many friends which results in star-like patterns in the graph.

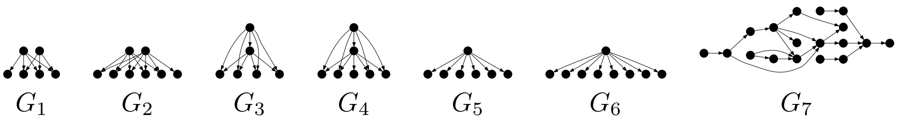
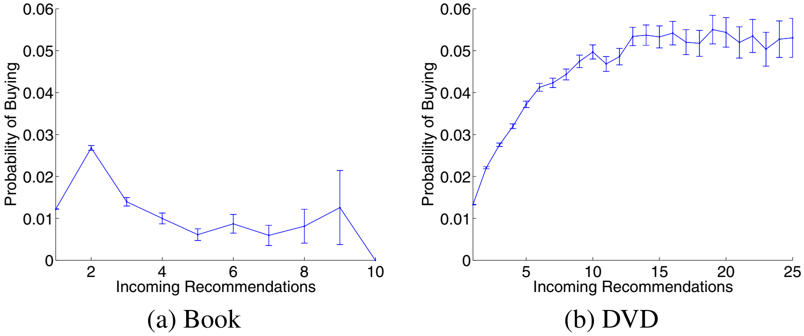


Figure 3. Typical classes of cascades. G_1, G_2: nodes recommending to the same set of people, but not each other. G_3, G_4: nodes recommending to same community. G_5, G_6: a flat cascade. G_7: a large propagation of recommendations.



(a) Book (b) DVD

Figure 4. Probability of purchasing a product given the number of received recommendations. Notice the decrease in purchasing probability for books and saturation for DVDs.

2.1.1. Cascading patterns

Consider the problem of finding patterns of recommendations in a large social network. One can ask the following questions: How does the influence propagate? What does it look like?

In order to analyze the data, new methods and algorithms had to be developed. First, to identify cascades, *i.e.* graphs where incoming recommendations influenced purchases and further recommendations. Next, to enumerate and count the cascade subgraphs. Graph isomorphism and enumeration are both computationally very expensive, so new algorithms for approximate graph isomorphism resolution were developed [25]. In a multi-level approach the computational complexity (and accuracy) of the graph isomorphism resolution depends on the size of the graph. This property makes the algorithm scale nicely to large datasets.

It has been found [24] that the distribution of sizes and depths of cascades follows a power law. Generally, cascades tend to be shallow, but occasional large bursts can occur. Cascades are mainly tree-like, but variability in connectivity and branching across different products groups was also observed. Figure 3 shows some typical examples of how the influence propagates over the recommendation network.

In addition to observing rich cascades and propagation one can make a step further and analyze the effectiveness and dynamics of product recommendations in causing purchases.

2.1.2. Implications for viral marketing

A study of Leskovec et al. [21] established how the recommendation network grows over time and how effective it is from the viewpoint of the sender and receiver of the recommendations. They examine what kind of product is more likely to be bought as a result of recommendation, and describe the size of the cascade that results from recommendations and purchases. While on average recommendations are not very effective at inducing purchases and do not spread very far, there are product and pricing categories for which viral marketing seems to be very effective.

Figure 4 presents an example of the findings. We plot the probability of purchasing a product given the number of received recommendations. Surprisingly, as more book recommendations are received their success *decreases*. Success of DVD recommendations saturates around 10 incoming recommendations. This means that after a person gets 10 recommendations they become immune to them – their probability of buying does not increase anymore. Traditional innovation diffusion models assume that an increasing number of infected contacts results in an increased likelihood of infection. Instead, it was shown that the probability of purchasing a product increases with the number of recommendations received, but then it quickly saturates. The result has important implications for viral marketing because providing too much incentive for people to recommend to one another can weaken the very social network links that the marketer is intending to exploit.

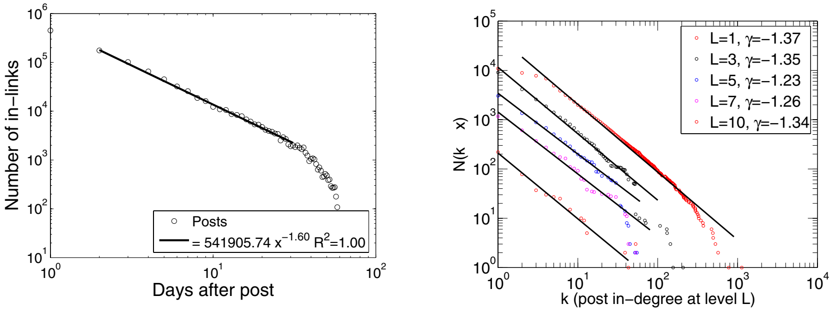
What determines the product's viral marketing success? A study [22] presents a model which characterizes product categories for which recommendations are more likely to be accepted, and find that the numbers of nodes and receivers have negative coefficients, showing that successfully recommended products are actually more likely to be not so widely popular. It shows that more expensive and more recommended products have a higher success rate. These recommendations should occur between a small number of senders and receivers, which suggests a very dense recommendation network where lots of recommendations are exchanged between a small community of people. These insights could be of use to marketers — personal recommendations are most effective in small, densely connected communities enjoying expensive products. Refer to [22] for more details.

2.2. Cascades on the blogosphere

Similarly to the viral marketing setting we also analyze cascades on the blogosphere. We address a set of related questions: What kinds of cascades arise frequently in real life? Are they like trees, stars, or something else? And how do they reflect properties of their underlying network environment?

2.2.1. Shape of information cascades

We extracted dataset presented here from a larger set of blogs and posts from August and September 2005 [10]. We were interested in blogs and posts that actively participate in discussions, so we biased our dataset towards the more active part of the blogosphere. We focused on the most-cited blogs and traced forward and backward conversation trees containing these blogs. This process produced a dataset of 2.5 million posts from $45,000$ blogs gathered over the three-month period. To analyze the data, we first create graphs of



(a) Power-law popularity decay

(b) In-degree at level L

Figure 5. (a) Number of in-links vs. the days after the post in log-linear scale, after removing the day-of-the week effects. The power law fit has the exponent -1.5. (b) Post in-degree distribution (number of in-links of a post) for posts at different depths of the cascade. Notice the power-law exponent practically remains constant which means that even posts that appear late (deep) in the cascade attract many in-links.

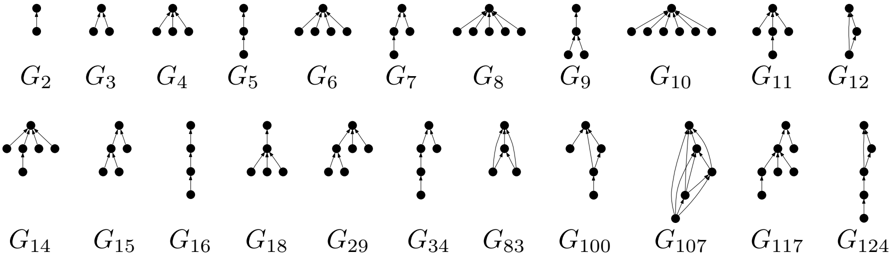


Figure 6. Common blog cascade shapes, ordered by the frequency of appearance.

time-obeying propagation of links. Then, we enumerate and count all possible cascade subgraphs.

We find novel patterns, and the analysis of the results gives us insight into the cascade formation process. Most surprisingly, the popularity of posts drops with a *power law*, instead of exponentially, that one may have expected. We collect all in-links to a post and plot the number of links occurring after each day following the post. This creates a curve that indicates the rise and fall of popularity. Figure 5(a) shows number of in-links for each day following a post for all posts in the dataset The exponent of the power law is -1.5, which is exactly the value predicted by the model where the bursty nature of human behavior is a consequence of a decision based queuing process [29,34] – when individuals execute tasks based on some perceived priority, the timing of the tasks is heavy tailed, with most tasks being rapidly executed, whereas a few experience very long waiting times.

We also find that probability of observing a cascade on n nodes follows a Zipf distribution: $p(n) \propto n^{-2}$. Figure 5(b) plots the in-degree distribution of nodes at level L of the cascade. A node is at level L if it is L hops away from the root (cascade initiator) node. Notice that the in-degree exponent is stable and does not change much given the level in the cascade. This means that posts still attract attention (get linked) even if they are somewhat late in the cascade and appear towards the bottom of it.

We also found rich cascade patterns. Generally cascades are shallow but occasional large bursts also occur. The cascade sub-patterns shown on figure 6 reveal mostly small tree-like subgraphs; however we observe differences in connectivity, density, and the shape of cascades. Indeed, the frequency of different cascade subgraphs is not a simple consequence of differences in size or density; rather, we find instances where denser subgraphs are more frequent than sparser ones, in a manner suggestive of properties in the underlying social network and propagation process.

For example, we found that BoingBoing, which is a very popular blog about "amusing things", is engaged in many cascades. Actually, 85% of all BoingBoing posts were cascade initiators. The cascades generally did not spread very far but were wide (*e.g.*, G_{10} and G_{14} in Figure 6). On the other hand 53% of the posts from an influential political blog MichelleMalkin were cascade initiators, but the cascades here were deeper and generally larger (*e.g.*, G_{117} in Figure 6) than those of BoingBoing.

3. Simple model of information cascades

Next we present a conceptual model for generating information cascades that produces cascade graphs matching several properties of real cascades. The model builds on independent cascade model [16]. The model is intuitive and requires only a single parameter that corresponds to how interesting (easy spreading) the conversations in general on the blogosphere are.

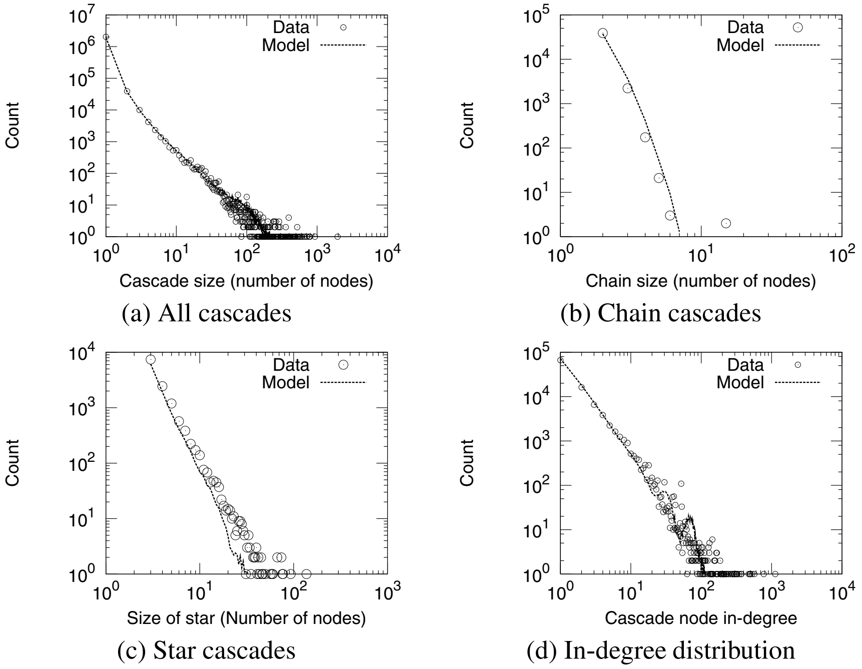
Intuitively, cascades are generated by the following principle. A post is posted at some blog, other bloggers read the post, some create new posts, and link the source post. This process continues and creates a cascade. One can think of cascades as graphs created by the spread of a virus over the Blog Network. This means that the initial post corresponds to infecting a blog. As the cascade unveils, the virus (information) spreads over the network and leaves a trail. To model this process we use a single parameter β that measures the infectiousness of the posts on the blogosphere. The idea is that a blog B writes a new post and then with probability β neighbors in the social network of blog B get infected as they write a post and link the B's post. The model is very similar to the SIS (susceptible – infected – susceptible) model from the epidemiology [15].

Figure 7 compares the cascades generated by the model with the ones found in the real blog network. Notice a very good agreement between the reality and simulated cascades in all plots. The distribution over cascade sizes is matched best. Chains and stars are slightly under-represented, especially in the tail of the distribution where the variance is high. The in-degree distribution is also matched nicely, with an exception for a spike that can be attributed to a set of outlier blogs all with in-degree 52.

4. Node selection for early cascade detection

Next, we explore the general problem of detecting outbreaks in networks, where we are given a network and a dynamic process spreading over this network, and we want to select a set of nodes to detect the process as effectively as possible.

Many real-world problems can be modeled under this setting. Consider a city water distribution network, delivering water to households via pipes and junctions. Acciden-



(a) All cascades

(b) Chain cascades

(c) Star cascades

(d) In-degree distribution

Figure 7. Comparison of the true data and the model. We plotted the distribution of the true cascades with circles and the estimate of the model with dashed line. Notice remarkable agreement between the data and the prediction of the simple model.

tal or malicious intrusions can cause contaminants to spread over the network, and we want to select a few locations (pipe junctions) to install sensors, in order to detect these contaminations as quickly as possible.

As we already saw in the domain of weblogs (blogs), bloggers publish posts and use hyper-links to refer to other bloggers' posts and content on the web. Each post is time stamped, so we can observe the spread of information on the "blogosphere". In this setting, we want to select a set of blogs to read (or retrieve) which are most up to date, *i.e.*, catch (link to) most of the stories that propagate over the blogosphere. Figure 1(a) illustrates this setting. Each layer plots the propagation graph of the information. Circles correspond to blog posts, and all posts at the same vertical column belong to the same blog. Edges indicate the temporal flow of information: the cascade starts at some post (*e.g.*, top-left circle of the top layer of Figure 1(a)) and then the information propagates recursively by other posts linking to it. Our goal is to select a small set of blogs (two in case of Figure 1(a)) which "catch" as many cascades (stories) as possible. (In real-life multiple cascades can be on the same or similar story, but we still aim to detect as many as possible.) A naive, intuitive solution would be to select the big, well-known blogs. However, these usually have a large number of posts, and are time-consuming to read. We show, that, perhaps counterintuitively, a more cost-effective solution can be obtained, by reading smaller, but higher quality, blogs, which our algorithm can find.

4.1. Problem definition

More formally, we want to select a subset $\mathcal{A}$ of nodes (sensor locations, blogs) in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which detect outbreaks (spreading of a information) quickly.

Figure 1(b) presents an example of such a graph for blog network. Each of the six blogs consists of a set of posts. Connections between posts represent hyper-links, and labels show the time difference between the source and destination post, *e.g.*, post p_{41} linked p_{12} one day after p_{12} was published).

These outbreaks (*e.g.*, information cascades) initiate from a single node of the network (*e.g.*, p_{11}, p_{12} and p_{31}), and spread over the graph, such that the traversal of every edge $(s, t) \in \mathcal{E}$ takes a certain amount of time (indicated by the edge labels). As soon as the event reaches a selected node, an alarm is triggered, e.g., selecting blog B_6, would detect the cascades originating from post p_{11}, p_{12} and p_{31}, after 6, 6 and 2 timesteps after the start of the respective cascades.

Depending on which nodes we select, we achieve a certain placement score. Figure 1(b) illustrates several criteria one may want to optimize. If we only want to detect as many stories as possible, then reading just blog B_6 is best. However, reading B_1 would only miss one cascade (p_{31}), but would detect the other cascades immediately. In general, this placement score (representing, *e.g.*, the fraction of detected cascades, or the population saved by placing a water quality sensor) is a set function R, mapping every placement $\mathcal{A}$ to a real number $R(\mathcal{A})$ (our reward), which we intend to maximize.

Since blogs are expensive, we also associate a *cost* $c(\mathcal{A})$ with every placement $\mathcal{A}$, and require, that this cost does not exceed a specified budget B which we can spend. For example, the cost of selecting a blog could be the number of posts in it (*i.e.*, B_1 has cost 2, while B_6 has cost 6). In the water distribution setting, accessing certain locations in the network might be more difficult (expensive) than other locations. Also, we could have several types of sensors to choose from, which vary in their quality (detection accuracy) and cost. We associate a nonnegative cost $c(s)$ with every blog s, and define the cost of placement $\mathcal{A}$: $c(\mathcal{A}) = \sum_{s \in \mathcal{A}} c(s)$.

Using this notion of reward and cost, our goal is to solve the optimization problem

$$\max_{\mathcal{A} \subseteq \mathcal{V}} R(\mathcal{A}) \text{ subject to } c(\mathcal{A}) \leq B, \tag{1}$$

where B is a budget we can spend for selecting the nodes, and $R(\mathcal{A}) = \sum_i R_i(\mathcal{A})$, where $R_i(\mathcal{A})$ is the reward we get for detecting cascade i by monitoring blogs in set $\mathcal{A}$.

4.2. Node selection criteria

There are several possible criteria one may want to optimize in outbreak detection. For example, one criterion seeks to minimize *detection time* (*i.e.*, to know about a cascade as soon as possible, or avoid spreading of contaminated water). Similarly, another criterion seeks to minimize the *population affected* by an undetected outbreak (*i.e.*, the number of blogs referring to the story we just missed, or the population consuming the contamination we cannot detect).

The detection time $T(i, s)$ in the blog setting is the time difference in days, until blog s participates in the cascade i, which we extract from the data. In the water network, $T(i, s)$ is the time it takes for contaminated water to reach node s in scenario i (depending on outbreak location and time). Given a set of monitored nodes (blogs, sensors) $\mathcal{A}$ let t_i denote the detection time, i.e., the time when cascade i first hit one of the blogs in $\mathcal{A}$, $t_i = \min_{s \in \mathcal{A}} T(i, s)$.

Now we can define the following objective functions:

1. *Detection likelihood (DL)*: fraction of information cascades detected by the selected nodes. Here, we do incur reward of 1 ($R_i(\mathcal{A}) = 1$) if we detect outbreak i in finite time ($t_i < \infty$), otherwise we incur reward of 0.
2. *Detection time (DT)* measures the time passed from outbreak till detection by one of the selected nodes: $R_i(\mathcal{A}) = t_i - T_i$, where T_i is the start time of cascade i.
3. *Population affected (PA)* by scenario (cascade, outbreak). The affected population measures the number of blogs involved in a cascade after the detection. The idea is that by reading blog s we would like to be the first to know, i.e., there will be many blogs joining the cascade after s. Here, $R_i(\mathcal{A})$ is the size of (number of blogs joining) cascade i after time t_i.

4.3. Exploiting submodularity

Optimizing the above objective functions is NP-hard [17], so for large, real-world problems, we cannot expect to find the optimal solution.

However, the functions defined above preserve problem structure. $R(\mathcal{A})$ has several important and intuitive properties: Firstly, $R(\emptyset) = 0$, *i.e.*, we do not get any reward if we do not read any blogs. Secondly, R is nondecreasing, *i.e.*, $R(\mathcal{A}) \leq R(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$. Hence, adding blogs can only increase the incurred reward (we detect more cascades). Thirdly, and most importantly, it satisfies the following intuitive diminishing returns property: If we add a blog to a small set $\mathcal{A}$, we improve our score at least as much, as if we add it to a larger set $\mathcal{B} \supseteq \mathcal{A}$. More formally, we can prove that for all sets $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and blogs $s \in \mathcal{V} \setminus \mathcal{B}$, it holds that

$$R(\mathcal{A} \cup \{s\}) - R(\mathcal{A}) \geq R(\mathcal{B} \cup \{s\}) - R(\mathcal{B}).$$

A set function R with this property is called *submodular*. We give the proof of the above fact in [23].

So, the outbreak detection objective functions are *submodular* [27], *i.e.*, they exhibit a diminishing returns property: Reading a blog (or placing a sensor) when we have only read a few blogs provides more new information, than reading it after we have read many blogs (placed many sensors). We find ways to exploit this submodularity property to *efficiently obtain* solutions which are *provably close* to the optimal solution. These guarantees are important in practice, since selecting nodes is expensive (reading blogs is time-consuming, sensors have high cost), and we desire solutions which are not too far from the optimal solution.

Now, we show how to exploit the submodularity of the objective (*e.g.*, detection time) to develop an efficient approximation algorithm, CELF, which achieves near-optimal placements (guaranteeing at least a constant fraction of the optimal solution).

First, consider two possible strategies to solve the problem. First, consider a simple greedily algorithm where sensors are sequentially added to the solution set $\mathcal{A}'$. At each step k sensor s_k is added to the solution set $\mathcal{A}'$ that maximizes the marginal reward,

$$s_k = \operatorname*{argmax}_{s \in \mathcal{V} \setminus \mathcal{A}'_{k-1}} R(\mathcal{A}'_{k-1} \cup \{s\}) - R(\mathcal{A}'_{k-1}).$$

The algorithm stops when the budget B is exhausted. However this algorithm does not work well as one can show that the solution $\mathcal{A}'$ can be arbitrarily bad (far away from optimal) [23].

Now, consider the modification of the greedy algorithm that produces a different solution set $\mathcal{A}''$. The idea here is to optimize the benefit/cost ratio, i.e., we prefer sensors that give us the most reward for their cost. So, at k-th step of the algorithm we choose node s_k that:

$$s_k = \operatorname*{argmax}_{s \in \mathcal{V} \setminus \mathcal{A}''_{k-1}} \frac{R(\mathcal{A}''_{k-1} \cup \{s\}) - R(\mathcal{A}''_{k-1})}{c(s)},$$

i.e., the greedy algorithm picks the element maximizing the benefit/cost ratio. The algorithm stops once no element can be added to the current set $\mathcal{A}''$ without exceeding the budget B. Similarly as before one can show that such benefit/cost optimization can give arbitrarily bad solutions [23].

However, the good news is that if one uses the best of both solutions, then we can prove

$$\max\{R(\mathcal{A}'), R(\mathcal{A}'')\} \geq \frac{1}{2}(1 - 1/e) \max_{\mathcal{A}, c(\mathcal{A}) \leq B} R(\mathcal{A}).$$

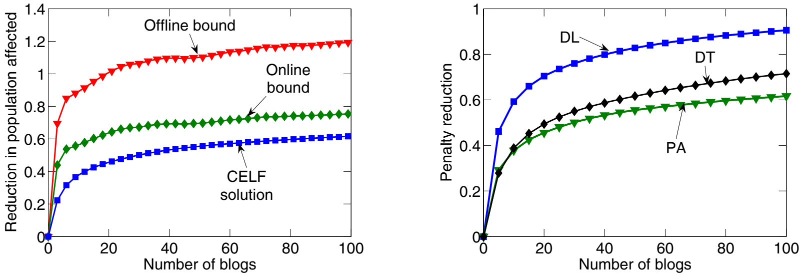
where $\mathcal{A}$ denotes the optimal solution to the problem that is NP-hard to compute in practice, and e is the base of a natural logarithm ($e = 2.71$). For further details on the formulation and the approximation guarantee see [23].

Now, we can define the CELF algorithm to be exactly the solution from the above theorem. We run two independent greedy algorithms and then pick the best of two solutions. First we compute a solution $\mathcal{A}'$ by running simple greedy algorithm that selects sensors based on marginal rewards as defined above. Then we also independently run a second greedy algorithm that this time optimizes the benefit/cost ratio which gives us solution set $\mathcal{A}''$. And the solution returned CELF by is by definition $\max\{R(\mathcal{A}'), R(\mathcal{A}'')\}$. In practice this means that solution found by CELF will be at most factor 3 ($\frac{1}{2}(1 - \frac{1}{e}) \approx 3$) away from the unknown hard to compute optimal solution. Moreover, we also derive novel data dependent bound that gives us even better information on how far away from unknown optimal is the solution returned by CELF. See [23] for the details on the online data dependent bound.

4.4. Evaluation on water distribution and blog networks

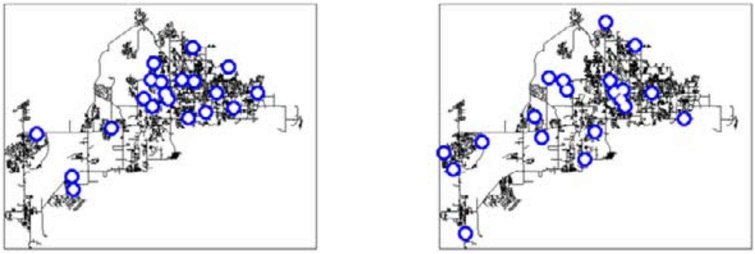
Next, we show the evaluation of the methodology on the applications introduced above – water quality and blogosphere monitoring. These are large real-world problems, involving a model of a water distribution network from the EPA with millions of contamination scenarios, and real blog data with millions of posts.

First, we evaluate the performance of CELF, and estimate how far from optimal the solution could be. Obtaining the optimal solution would require enumeration of $2^{45,000}$ subsets. Since this is impractical, we compare the algorithm to the bounds we develed.



(a) Solution quality (b) Various Objective functions

Figure 8. Both plots show the solution quality vs. the number of selected sensors (blogs). (a) Performance of CELF algorithm and off-line and on-line bounds. Notice on-line bound is much tighter. (b) Compares different objective functions: detection likelihood (DL), detection time (DT) and population affected (PA).



(a) Population Affected (b) Detection Likelihood

Figure 9. Water network sensor placements: (a) when optimizing Population Affected, sensors are concentrated in high population areas. (b) when optimizing Detection Likelihood, sensors are uniformly spread out.

Figure 8(a) shows scores for increasing budgets when optimized the Population affected criterion. As we select more blogs to read, the proportion of cascades we catch increases (bottom line). We also plot the two bounds. Notice the off-line bound (top line) is very loose. On the other hand, the on-line bound is much tighter than the traditional off-line bound.

In contrast to the off-line bound, our on-line bound is *algorithm independent*, and thus can be computed regardless of the algorithm used to obtain the solution. Since it is tighter, it gives a much better worst case estimate of the solution quality. For this particular experiment, we see that CELF works very well: after selecting 100 blogs, we are at most 13.8% away from the optimal solution. Similarly, figure 8(b) shows the performance using various objective functions. By using the on-line bound we also calculated that our results for all objective functions are at most 5% to 15% from optimal. See [23] for more details.

In August 2006, the Battle of Water Sensor Networks (BWSN) [30] was organized as an international challenge to find the best sensor placements for a real metropolitan area water distribution network. In Figure 9 we show two 20 sensor placements obtained by our algorithm after optimizing Detection Likelihood and Population Affected, respectively. When optimizing the population affected, the placed sensors are concentrated in the dense high-population areas, since the goal is to detect outbreaks which affect the population the most. When optimizing the detection likelihood, the sensors are uniformly spread out over the network. Intuitively this makes sense, since according to BWSN challenge, outbreaks happen with same probability at every node. So, for Detection Likelihood, the placed sensors should be as close to all nodes as possible. See [18] for more details.

References

[1] L. A. Adamic and N. Glance. The political blogosphere and the 2004 u.s. election: divided they blog. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

[2] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *Web Intelligence*, pages 207–214, 2005.

[3] R. M. Anderson and R. M. May. *Infectious diseases of humans: Dynamics and control.* Oxford Press, 2002.

[4] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications.* Hafner Press, 2nd edition, 1975.

[5] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change in informational cascades. *Journal of Political Economy*, 100(5):992–1026, October 1992.

[6] J. J. Brown and P. H. Reingen. Social ties and word-of-mouth referral behavior. *The Journal of Consumer Research*, 14(3):350–362, 1987.

[7] D. Chakrabarti, J. Leskovec, C. Faloutsos, S. Madden, C. Guestrin, and M. Faloutsos. Information survival threshold in sensor and p2p networks. In *INFOCOM '07: Proceedings of the 26th annual IEEE Conference on Computer Communications*, pages 1316–1324, 2007.

[8] J. Chevalier and D. Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345, 2006.

[9] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001.

[10] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Deriving marketing intelligence from online discussion. In *KDD '05: Proceeding of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 419–428, 2005.

[11] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 3(12):211–223, 2001.

[12] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[13] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[14] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, 2004.

[15] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Rev.*, 42(4):599–653, 2000.

[16] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[17] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[18] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning an Management*, 0:0, 2008.

[19] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 568–576, 2003.

[20] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, 2006.

[21] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 228–237, 2006.

[22] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):2, 2007.

[23] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD '07: Proceeding of the 13th ACM SIGKDD international conference on Knowledge discovery in data mining*, 2007.

[24] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM '07: Proceedings of the SIAM Conference on Data Mining*, 2007.

[25] J. Leskovec, A. Singh, and J. M. Kleinberg. Patterns of influence in a recommendation network. In *PAKDD '06: Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 380–389, 2006.

[26] A. L. Montgomery. Applying quantitative marketing techniques to the internet. *Interfaces*, 30:90–108, 2001.

[27] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[28] M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.

[29] J. G. Oliveira and A. L. Barabasi. Human dynamics: The correspondence patterns of darwin and einstein. *Nature*, 437:1251, 2005.

[30] A. Ostfeld, J. G. Uber, and E. Salomons. Battle of water sensor networks: A design challenge for engineers and algorithms. In *8th Symposium on Water Distribution Systems Analysis*, 2006.

[31] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In *The Economics of the Internet and E-Commerce*. Elsevier Science, 2002.

[32] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02: Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70, 2002.

[33] E. M. Rogers. *Diffusion of Innovations*. Free Press, New York, fourth edition, 1995.

[34] A. Vazquez, J. G. Oliveira, Z. Dezso, K.-I. Goh, I. Kondor, and A.-L. Barabasi. Modeling bursts and heavy tails in human dynamics. *Physical Review E*, 73(3):036127, 2006.

[35] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *SRDS*, pages 25–34, 2003.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-186

Link Analysis in Networks of Entities

Ronen Feldman[1]
School of Business Administration, Hebrew University, ISRAEL

Abstract. This chapter presents a link analysis approach for analyzing large networks of entities extracted from document collections. We assume that an initial NER and relationship extraction was performed and we are interested in analyzing the structure of the network of these entities. We start by presenting the basic definitions and then follow to focus on centrality analysis and computing equivalence classes on entities within the network. We use the network of the 9-11 hijackers as a running example.

Keywords. Text Mining, Link Analysis.

Introduction

When we get a large set of documents, we can perform entity extraction of the documents and find the relevant entities. Based on the outcome of the entity extraction stage we can establish links between entities either by using co-occurrence information (within some lexical unit such as document, paragraph, or sentence) or by using the semantic relationships between the entities as extracted by the information extraction module (such as family relations, employment relationship, mutual service in the army, etc). In this chapter we will describe the link analysis techniques that can be applied to results of the preprocessing stage (information extraction and term extraction).

A social network is a set of entities (e.g. people, companies, organizations, universities, countries) and a set of relationships between them (e.g. family relationships, various types of communication, business transactions, social interactions, hierarchy relationships, and shared memberships of people in organizations). Visualizing a social network as a graph enables the viewer to see patterns that are not evident before.

We start the chapter with a description of the running example of the 9/11 hijacker's network. After presenting the concepts of centrality and the various ways how to compute it we move to finding equivalence classes between nodes in the network.

1. Running Example: 9/11 Hijackers

We have collected information about the 19 9/11 hijackers from the following sources:

1. Names of the 19 hijackers, and the flights they boarded were taken from the FBI site http://www.fbi.gov/pressrel/pressrel01/091401hj.htm (see Table 1)

[1] Corresponding Author: Ronen Feldman, Information Systems Department, School of Business Administration, Mount Scopus, Jerusalem, ISRAEL 91905; E-mail: Ronen.Feldman@huji.ac.il.

2. Prior connections between the hijackers are based on information that was collected from the Washington Post site. If there was a connection between *n≥3* people, it was converted to *C(n,2)* (*n* choose 2) symmetric binary relations between each pair of people. http://www.washingtonpost.com/wp-srv/nation/graphics/attack/investigation_24.html

The undirected graph of binary relations between the hijackers is shown in Figure 1. The graph was drawn using Pajek, dedicated freeware link analysis software [1]. Force based graph drawing algorithms are described in [2, 3]. Algorithms for drawing large graphs are described in [4-6].

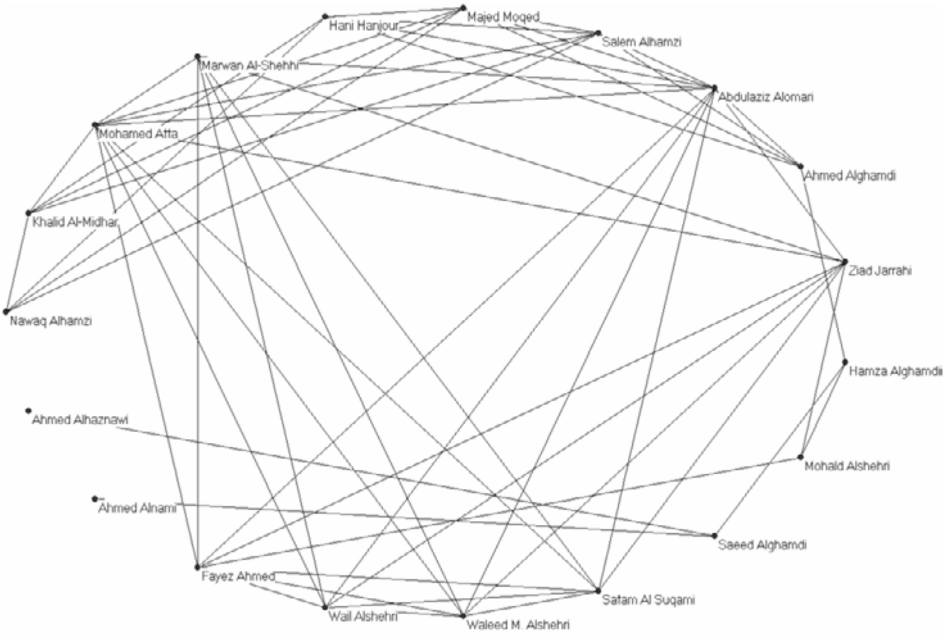


Figure 1. Connections between the 9/11 hijackers

The 19 hijackers boarded 4 flights, and in Table 1 we can see the names of the hijackers that boarded each flight.

Table 1. The 19 Hijackers ordered by flights

Flight 77 : Pentagon	*Flight 11 : WTC 1*	*Flight 175 : WTC 2*	*Flight 93: PA*
Khalid Al-Midhar	Satam Al Suqami	Marwan Al-Shehhi	Saeed Alghamdi
Majed Moqed	Waleed M. Alshehri	Fayez Ahmed	Ahmed Alhaznawi
Nawaq Alhamzi	Wail Alshehri	Ahmed Alghamdi	Ahmed Alnami
Salem Alhamzi	Mohamed Atta	Hamza Alghamdi	Ziad Jarrahi
Hani Hanjour	Abdulaziz Alomari	Mohald Alshehri	

We will use the flight information later when we discuss the various clustering schemes of the hijackers.

2. Centrality

The notion of centrality enables to identify the main and most powerful actors within a social network. Those actors should get special attention when monitoring the behavior of the network.

Centrality is a structural attribute of vertices in a network, it has nothing to do with the features of the actual objects represented by the vertices of the network (i.e., if it is a network of people, their nationality, title or any physical feature). When dealing with directed networks we use the term *prestige*. There are two types of prestige; the one defined on outgoing arcs is called *influence* while the one defined on incoming arcs is called *support*. Since most of our networks are based co-occurrence of entities in the same lexical unit, we will focus our attention to undirected networks and use the term centrality. The different measures of centrality that we will present can be adapted easily for directed networks and measure influence or support.

There are 5 major definitions used for centrality: Degree centrality, closeness centrality, betweeness centrality, eigenvector centrality and power centrality. We discuss these in the subsequent sections.

2.1. Degree Centrality

If the graph is undirected then the degree of a vertex $v \in V$ is the number of other vertices that are directly connected to it.

Definition: degree(v) = $|\{(v1, v2) \in E \mid v1 = v \text{ or } v2 = v\}|$

If the graph is directed then we can talk about in-degree or out-degree. An edge (v1,v2) $\in$ E in the directed graph is leading from vertex v1 to v2.
In-degree(v) = $|\{(v1, v) \in E\}|$
Out-degree(v) = $|\{(v, v2) \in E\}|$

If the graph represents a social network then clearly people who have more connections to other people can be more influential and they can utilize more of the resources of the network as a whole. Such people are often mediators and deal makers in exchanges among others, and are able to benefit from this brokerage.

When dealing with undirected connections, people differ from one another only in how many connections they have. In contrast, when the connections are directed, it is important to distinguish centrality based on in-degree from centrality based on out-degree. If a person has a high in-degree we will say that this person is *prominent*, and has high *prestige*. Many people seek direct connections to him indicating his importance. People who have high out-degree are people who are able to interact with many others and possibly spread their ideas. Such people are said to be *influential*. In

Table 2, we can see the hijackers sorted in a decreasing order of their (undirected) degree measures. We can see that Mohamed Atta and Abdulaziz Alomari have the highest degree.

Table 2. Degree Measure of the Top Hijackers

Name	Degree
Mohamed Atta	11
Abdulaziz Alomari	11
Ziad Jarrahi	9
Fayez Ahmed	8
Waleed M. Alshehri	7
Wail Alshehri	7
Satam Al Suqami	7
Salem Alhamzi	7
Marwan Al-Shehhi	7
Majed Moqed	7

2.2. Closeness Centrality

Degree centrality measures might be criticized because they only take into account the direct connections that an entity has, rather than indirect connections to all other entities. One entity might be directly connected to a large number of entities that might be pretty isolated from the network. Such an entity is central only in a local neighborhood of the network.

In order to solve the shortcomings of the degree measure we can utilize the closeness centrality. This measure is based on the calculation of the geodesic distance between the entity and all other entities in the network. We can either use directed or undirected geodesic distances between the entities. In our current example, we have decided to look at undirected connections. The sum of these geodesic distances for each entity is the "farness" of the entity from all other entities. We can convert this into a measure of closeness centrality by taking its reciprocal. We can normalize the closeness measure by dividing it by the closeness measure of the most central entity.

Formally, let $d(v_1,v_2)$ = the minimal distance between v_1 and v_2, i.e., the minimal number of vertices that we need to pass on the way from v_1 to v_2.

The closeness centrality of vertex v_i is defined as

$$C_i = \frac{|V|-1}{\sum_{j \neq i} d(v_i, v_j)},$$

this is the reciprocal of the average geodesic distance between v_i and any other vertex in the network.

Table 3. Closeness measures of the Top hijackers

Name	Closeness
Abdulaziz Alomari	0.6
Ahmed Alghamdi	0.5454545
Ziad Jarrahi	0.5294118
Fayez Ahmed	0.5294118
Mohamed Atta	0.5142857
Majed Moqed	0.5142857
Salem Alhamzi	0.5142857
Hani Hanjour	0.5
Marwan Al Shehhi	0.4615385
Satam Al Suqami	0.4615385

2.3. Betweeness Centrality

Betweeness centrality measures the effectiveness in which a vertex connects the various parts of the network. Entities that are on many geodesic paths between other pairs of entities are more powerful since they control the flow of information between the pairs. That is, the more other entities depend on a certain entity to make connections, the more power this entity has. If, however, two entities are connected by more than one geodesic path, and a given entity is not on all of them, it loses some power. If we add up, for each entity, the proportion of times this entity is "between" other entities for transmission of information we get the betweeness centrality of that entity. We can normalize this measure by dividing it by the maximum possible betweeness that an entity could have had (which is the number of possible pairs of entities for which the entity is on every geodesic between them $= \frac{(|V|-1)(|V|-2)}{2}$).

Formally:
g_{jk} = the number of geodetic paths that connect v_j with v_k
$g_{jk}(v_i)$ = the number of geodetic paths that connect v_j with v_k and pass via v_i.

$$B_i = \sum_{j<k} \frac{g_{jk}(v_i)}{g_{jk}}$$

$$NB_i = \frac{2B_i}{(|V|-1)(|V|-2)}$$

2.4. Eigenvector Centrality

The main idea behind eigenvector centrality is that entities receiving many communications from other well connected entities, will be better and more valuable sources of information, and hence be considered central. The Eigenvector centrality scores correspond to the values of the principal eigenvector of the adjacency matrix *M*.

Table 4. Betweeness measures of the Top hijackers

Name	Betweeness (B_i)
Hamza Alghamdi	0.3059446
Saeed Alghamdi	0.2156863
Ahmed Alghamdi	0.210084
Abdulaziz Alomari	0.1848669
Mohald Alshehri	0.1350763
Mohamed Atta	0.1224783
Ziad Jarrahi	0.0807656
Fayez Ahmed	0.0686275
Majed Moqed	0.0483901
Salem Alhamzi	0.0483901

Formally, the vector v satisfies the equation $\lambda v = Mv$, where λ is the corresponding eigenvalue and M is the adjacency matrix.

The score of each vertex is proportional to the sum of the centralities of neighboring vertices. Intuitively, vertices with high eigenvector centrality score are connected to many other vertices with high scores which are, in turn, connected to many other vertices and its continues recursively. Clearly, the highest score will be obtained by vertices that are members of large cliques or large p-cliques. In Table 5 we can see that the members of the big clique (with 8 members) are those that got the highest scores. Atta and Al-Shehhi got much higher scores than all the other hijackers, mainly since the connection between them is so strong. They were also the pilots of the planes going into WTC1 and WTC2 and they are believed to be the leaders of the hijackers.

Table 5. Eigenvector centrality scores of the Top hijackers

Name	E1
Mohamed Atta	0.518
Marwan Al-Shehhi	0.489
Abdulaziz Alomari	0.296
Ziad Jarrahi	0.246
Fayez Ahmed	0.246
Satam Al Suqami	0.241
Waleed M. Alshehri	0.241
Wail Alshehri	0.241
Salem Alhamzi	0.179
Majed Moqed	0.165

2.5. Power Centrality

Power centrality was introduced by Bonacich. Given an adjacency matrix *M*, the power centrality of vertex *i* (denoted c_i), is given by

$$c_i = \sum_{j \neq i} M_{ij}(\alpha + \beta \cdot c_j)$$

α is used to normalize the score; the normalization parameter is automatically selected so that the sum of squares of the vertices's centralities is equal to the number of vertices in the network.

β is an attenuation factor that controls the effect that the power centralities of the neighboring vertices should have on the power centrality of the vertex.

In a similar way to the eigenvector centrality, the power centrality of each vertex is determined by the centrality of the vertices it is connected to. By specifying positive or negative values to β the user can control if the fact that a vertex is connected to powerful vertices should have a positive effect on its score or a negative effect. The rational for specifying a positive β is that if you are connected to powerful colleagues it makes you more powerful. On the other hand, the rational for a negative β is that powerful colleagues have many connections and hence are not controlled by you, while isolated colleagues have no other sources of information and hence are pretty much controlled by you.

Table 6. - Power Centrality for the Top hijackers

	Power : β = 0.99	Power : β = -0.99
Mohamed Atta	2.254	2.214
Marwan Al-Shehhi	2.121	0.969
Abdulaziz Alomari	1.296	1.494
Ziad Jarrahi	1.07	1.087
Fayez Ahmed	1.07	1.087
Satam Al Suqami	1.047	0.861
Waleed M. Alshehri	1.047	0.861
Wail Alshehri	1.047	0.861
Salem Alhamzi	0.795	1.153
Majed Moqed	0.73	1.029

2.6. Network Centralization

In addition to the individual vertex centralization measures, we can assign a number between 0 and 1 that will signal the level of centralization of the whole network. The network centralization measures will be computed based on the centralization values of its vertices and hence we will have for each type of individual centralization measure an associated network centralization measure. A network that is structured like a circle will have a network centralization value of 0 (since all vertices have the same centralization value), while a network that structured like a star will have a network centralization value of 1. We will now provide some of the formulas for the different network centralization measures. (*n* is the number of vertices in the network)

Degree

$$Degree^*(V) = Max_{v \in V} Degree(v)$$

$$NET_{Degree} = \frac{\sum_{v \in V} Degree^*(V) - Degree(v)}{(n-1)*(n-2)}$$

Clearly, if we have a circle, all vertices have a degree of 2, and hence $NET_{Degree}= 0$, and if we have a star of n nodes (one node in the middle), then that node will have a degree of n-1 and all other nodes will have a degree of 1, hence

$$NET_{Degree} = \frac{\sum_{v \in V \backslash v^*} (n-1)-1}{(n-1)(n-2)} = \frac{(n-1)(n-2)}{(n-1)(n-2)} = 1$$

For the hijackers graph $NET_{Degree}=0.31$

Betweenness

$$NB^*(V) = Max_{v \in V} NB(v)$$

$$NET_{Bet} = \frac{\sum_{v \in V} NB^*(V) - NB(v)}{(n-1)}$$

For the hijackers network $NET_{Bet} = 0.24$

Summary Diagram

In Figure 2 we can see a summary diagram of the different centrality measures as they are applied to the hijacker’s network. We marked by solid arrows the hijackers that got the highest value for the various centrality measures, and by dashed arrows the runners-up. We can see for instance that Atta has the highest value for degree centrality, eigenvector centrality and power centrality while Alomari has the highest value for degree centrality (tied with Atta) and closeness centrality and is the runner-up for power centrality (with a negative beta). Based on our experience the most important centrality measures are power and eigenvector (which are typically in agreement). Closeness and even more so betweeness centrality signal the people that are crucial in securing fast communication between the different parts of the network.

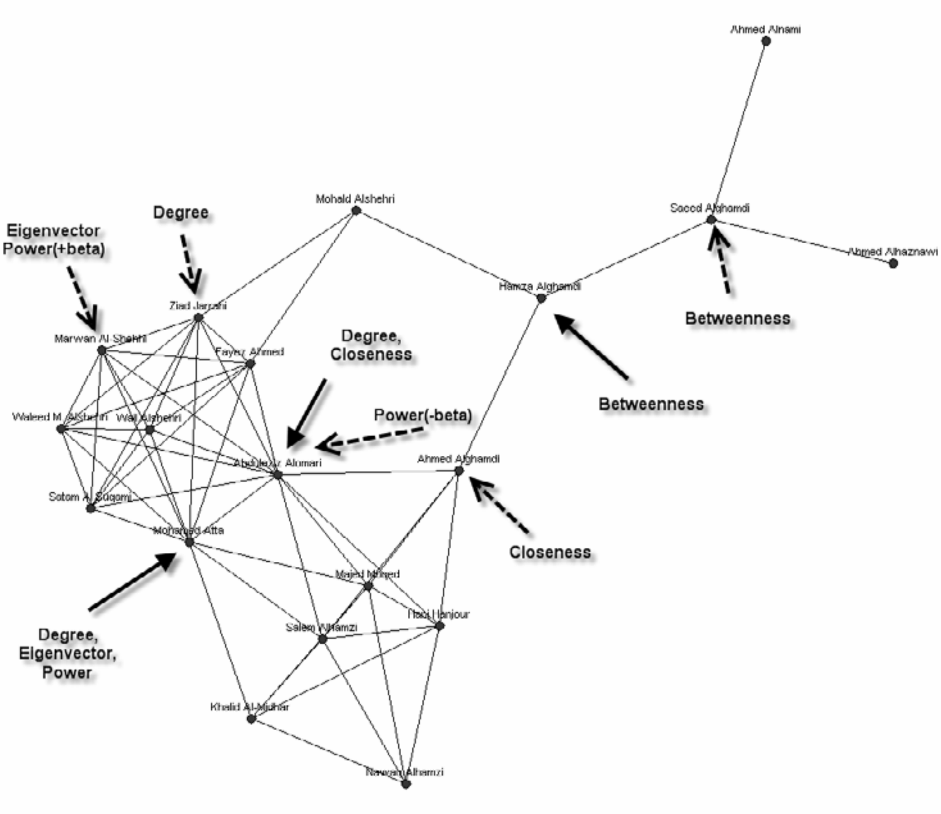


Figure 2. Summary Diagram of Centrality Measures (solid arrows point to highest value, dashed arrows point to second largest (done using netminer [7])

3. Equivalence between Entities

Given a network of entities, we are often interested in measuring the similarity between the entities based on their interaction with other entities in the network. In this section, we will formalize this notion of similarity between entities and provide examples of how to find similar entities and how we can utilize the similarity measure to cluster the entities.

3.1. Structural Equivalence

Two entities are said to be exactly structurally equivalent if they have the same relationships to all other entities. If A is "structurally equivalent" to B then it means these two entities are "substitutable". Typically we will not be able to find entities that are exactly structurally equivalent and hence we are interested in calculating the degree of structural equivalence between entities. Based on this measure of distance we will be able to perform hierarchical clustering of the entities in our network.

We will provide two formal definitions for structural equivalence. Both are based on the connection vectors of each of the entities. The first definition is based on the Euclidian distance between the connection vectors and other one is based on the number of exact matches between the elements of the vectors.

$$\text{EDis}(V_i, V_j) = \sqrt{\sum_k \left(M_{ik} - M_{jk}\right)^2}$$

$$\text{Match}(V_i, V_j) = \frac{\sum_{k=1}^{n} eq(M_{ik}, M_{jk})}{n}, \text{ where } eq(a,b) = \begin{cases} 1 & a = b \\ 0 & otherwise \end{cases}$$

3.2. Regular Equivalence

Two entities are said to be regularly equivalent if they have identical profile of connections with other entities that are also regularly equivalent. In order to establish regular equivalence we need to classify the entities into semantic sets such that each set contains entities with a common role. An example would be the sets of surgeons, nurses, and anesthesiologists. Lets assume that each surgeon is related to a set of 3 nurses and one anesthesiologist. We say that two such surgeons are regularly equivalent (and so are the nurses and the anesthesiologist), that is, they perform the same function in the network.

Entities that are "structurally equivalent" are also "regularly equivalent." However, entities who are "regularly equivalent" do not have to be "structurally equivalent." It is much easier to examine if two entities are structurally equivalent since there is a simple algorithm for finding EDis and Match. It is much harder to establish if two entities are regularly equivalent since we need to create a taxonomy of semantic categories on top of the entities. In Figure 3 we can see two pairs of people and one triplet that are structurally equivalent. In Table 7 we can see the EDis computed for each pair of entities. Entities that are structurally equivalent will have and EDis of 0. For instance Waleed M. Alshehri and Wail Alshehri are structurally equivalent and hence their EDis is 0. Based on this table we were able to use a hierarchical clustering algorithm (via the UCINET software package, see section 7.2) and generate the dendogram shown in Figure 4. People that are very close in the dendogram are similar structurally (i.e, they have low EDis), while people that are far away in the dendogram are different structurally.

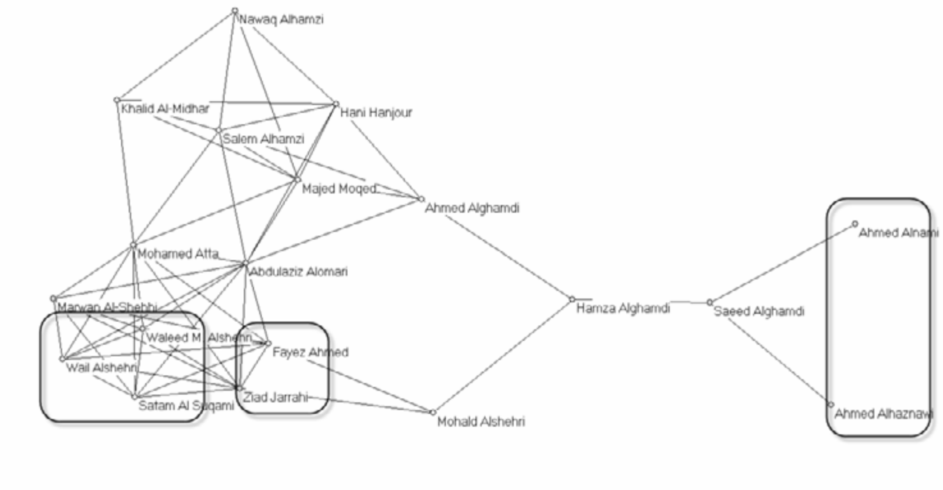


Figure 3. - Structural Equivalences in the Hijackers graph

Table 7. Euclidian distance (Edis) between each pair of entities

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	Nawaq Alhamzi	0.0	1.4	9.3	9.6	3.7	2.8	3.7	4.2	2.4	4.9	3.7	3.7	3.7	4.7	4.7	4.7	4.9	3.2	3.2
2	Khalid Al-Midhar	1.4	0.0	9.4	8.4	4.0	2.4	3.5	4.0	2.8	4.7	4.0	4.0	4.0	4.5	4.5	4.5	4.7	3.5	3.5
3	Mohamed Atta	9.3	9.4	0.0	2.4	9.8	9.7	10.2	7.5	9.4	7.6	9.8	9.4	9.8	7.5	7.5	7.5	7.6	9.6	9.6
4	Marwan Al-Shehhi	9.6	8.4	2.4	0.0	10.7	8.7	9.3	7.6	9.5	7.2	9.5	9.1	9.5	7.1	7.1	7.1	7.2	9.3	9.3
5	Hani Hanjour	3.7	4.0	9.8	10.7	0.0	3.2	2.0	5.3	4.0	6.8	6.0	6.3	6.3	6.6	6.6	6.6	6.8	6.0	6.0
6	Majed Moqed	2.8	2.4	9.7	8.7	3.2	0.0	1.4	4.2	3.2	5.3	4.7	5.1	5.1	5.1	5.1	5.1	5.3	4.7	4.7
7	Salem Alhamzi	3.7	3.5	10.2	9.3	2.0	1.4	0.0	4.9	4.0	6.2	5.7	6.0	6.0	6.0	6.0	6.0	6.2	5.7	5.7
8	Abdulaziz Alomari	4.2	4.0	7.5	7.6	5.3	4.2	4.9	0.0	4.0	3.2	4.9	4.5	5.3	2.8	2.8	2.8	3.2	4.9	4.9
9	Ahmed Alghamdi	2.4	2.8	9.4	9.5	4.0	3.2	4.0	4.0	0.0	4.7	3.5	3.5	3.5	4.5	4.5	4.5	4.7	3.5	3.5
10	Ziad Jarrahi	4.9	4.7	7.6	7.2	6.8	5.3	6.2	3.2	4.7	0.0	4.2	3.7	4.7	1.4	1.4	1.4	0.0	4.2	4.2
11	Hamza Alghamdi	3.7	4.0	9.8	9.5	6.0	4.7	5.7	4.9	3.5	4.2	0.0	2.8	2.8	4.5	4.5	4.5	4.2	2.0	2.0
12	Mohald Alshehri	3.7	4.0	9.4	9.1	6.3	5.1	6.0	4.5	3.5	3.7	2.8	0.0	2.8	3.5	3.5	3.5	3.7	2.8	2.8
13	Saeed Alghamdi	3.7	4.0	9.8	9.5	6.3	5.1	6.0	5.3	3.5	4.7	2.8	2.8	0.0	4.5	4.5	4.5	4.7	2.0	2.0
14	Satam Al Suqami	4.7	4.5	7.5	7.1	6.6	5.1	6.0	2.8	4.5	1.4	4.5	3.5	4.5	0.0	0.0	0.0	1.4	4.0	4.0
15	Waleed M. Alshehri	4.7	4.5	7.5	7.1	6.6	5.1	6.0	2.8	4.5	1.4	4.5	3.5	4.5	0.0	0.0	0.0	1.4	4.0	4.0
16	Wail Alshehri	4.7	4.5	7.5	7.1	6.6	5.1	6.0	2.8	4.5	1.4	4.5	3.5	4.5	0.0	0.0	0.0	1.4	4.0	4.0
17	Fayez Ahmed	4.9	4.7	7.6	7.2	6.8	5.3	6.2	3.2	4.7	0.0	4.2	3.7	4.7	1.4	1.4	1.4	0.0	4.2	4.2
18	Ahmed Alnami	3.2	3.5	9.6	9.3	6.0	4.7	5.7	4.9	3.5	4.2	2.0	2.8	2.0	4.0	4.0	4.0	4.2	0.0	0.0
19	Ahmed Alhaznawi	3.2	3.5	9.6	9.3	6.0	4.7	5.7	4.9	3.5	4.2	2.0	2.8	2.0	4.0	4.0	4.0	4.2	0.0	0.0

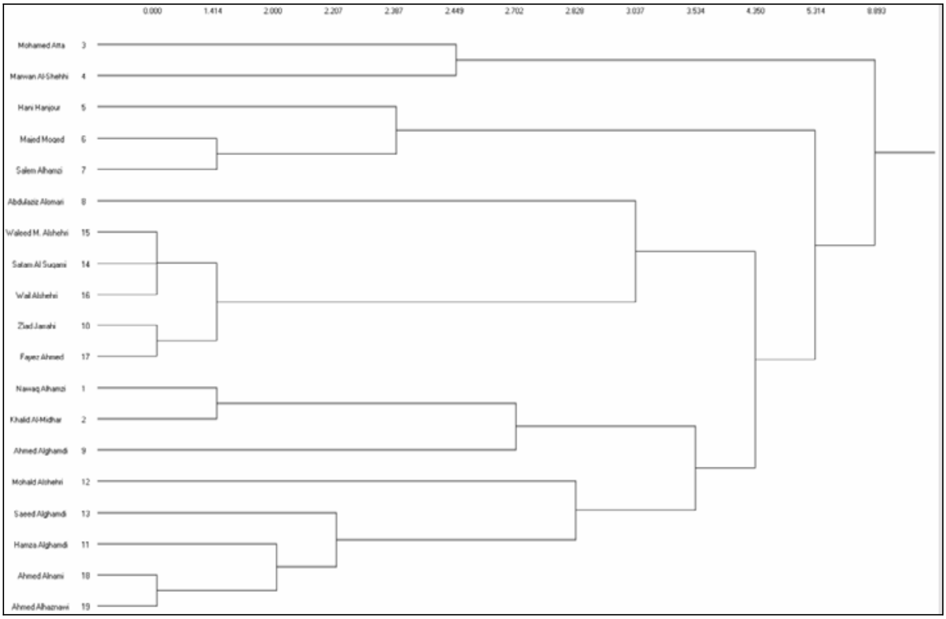


Figure 4. - Clustering based structural equivalence between the hijackers (we can see that {15,14,16} as well as {10,17} and {18,19} are structural equivalence classes)

Bibliography

[1] Batagelj, V. and A. Mrvar. *Pajek - Analysis and Visualization of Large Networks.* in *Graph Drawing Software*. 2003. Berlin: Springer.

[2] Kamada, T. and S. Kawai, *An Algorithm for Drawing General Undirected Graphs.* Information Processing Letters, 1989. **31**: p. 7-15.

[3] Fruchterman, T. and E. Reingold, *Graph Drawing by Force-directed Placement.* Software – Practice and Experience, 1991. **21**(11): p. 1129-1164.

[4] Davidson, R. and D. Harel, *Drawing Graphs Nicely Using Simulated Annealing.* ACM Transactions on Graphics, 1996. **15**(4): p. 301-331.

[5] Hadany, R. and D. Harel, *A Multi-Scale Method for Drawing Graphs Nicely.* Discrete Applied Mathematics, 2001. **113**: p. 3-21.

[6] Harel, D. and Y. Koren. *A Fast Multi-Scale Method for Drawing Large Graphs*. in *Graph Drawing*. 2000: Springer.

[7] Cyram. *NetMiner Webpage: (http://www.netminer.com/)*. 2004

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-198

Evolving networks

Pierre BORGNAT [a] Eric FLEURY [b] Jean-Loup GUILLAUME [c,1]
Clémence MAGNIEN [c] Céline ROBARDET [d] Antoine SCHERRER [a]
[a] *Université de Lyon, ENS Lyon, Laboratoire de Physique (UMR 5672 CNRS)*
[b] *Université de Lyon, ENS Lyon, INRIA/ARES*
[c] *Université Pierre & Marie Curie, LIP6 (UMR 7606 CNRS)*
[d] *Université de Lyon, INSA Lyon, LIRIS (UMR 5205 CNRS)*

Abstract. Most real networks often evolve through time: changes of topology can occur if some nodes and/or edges appear and/or disappear, and the types or weights of nodes and edges can also change even if the topology stays static. Mobile devices with wireless capabilities (mobile phones, laptops, etc.) are a typical example of evolving networks where nodes or users are spread in the environment and connections between users can only occur if they are near each other. This who-is-near-whom network evolves every time users move and communication services (such as the spread of any information) will deeply rely on the mobility and on the characteristics of the underlying network. This paper presents some recent results concerning the characterization of the dynamics of complex networks through three different angles: evolution of some parameters on snapshots of the network, parameters describing the evolution itself, and intermediate approaches consisting in the study of specific phenomena or users of interest through time.

Keywords. Complex networks, evolving networks, social networks.

Introduction

Complex networks play an important role in several scientific contexts: computer science, social and interaction networks or epidemiology. Typical examples of such networks are the Internet, web graphs, E-mail, phone calls, P2P networks, etc. In these networks, *links* between entities generally represent some kind of interaction. Studied as a whole, these networks share some non trivial properties and some problems span over a large variety of networks. For instance, the spreading of information is studied in computer science but also in epidemiology and the detection of dense subnetworks (communities) is also a problem having strong implications in many domains. Last decade, this domain has proposed a large set of tools which can be used on any complex network to get a deep insight on its properties and to compare it to other networks (see for instance [1] for a review of parameters).

However, one fundamental property has until recently be understudied. Complex networks evolve: new nodes and edges appear while some old ones disappear. These evolutions are often playing a key role in all the scientific domains cited above: people get new acquaintances, web pages are created or modified on a daily basis, machines are

[1] Corresponding Author E-mail: jean-loup.guillaume@lip6.fr

added or removed on the Internet, etc. If some studies are dedicated to the dynamics of complex networks [2,3,4] they are still too few. It appears crucial to better understand the evolution of these networks first to get knowledge but also to be able to generate random evolutive networks which can be used for simulation purposes.

In this paper, we detail three distinct approaches which are currently used to study complex networks and we explicit these approaches using typical complex networks. First, it is possible to describe the evolution of a network as a sequence of static networks and since there exist many parameters to describe accurately a static network, one can study the evolution of the network through the evolution of these parameters (Sec. 1). Second, one can study the evolution itself and define parameters to capture it, such as the rate of appearance or disappearance of nodes and edges (Sec. 2). Third, an intermediate approach can be used which consists in studying specific phenomena or users of interest through time (Sec. 3). For all these approaches methods from graph theory, statistical physics, data mining and random processes can be used and in many cases new tools and parameters have to be introduced.

1. Evolution of static properties

The most natural way to describe the dynamics of a complex network is to study the evolution of static properties through time. Static networks have been widely studied and a lot a simple parameters are available to describe a network as a whole (number of nodes and edges, number of triangles, specific subgraphs, length of paths, connected components, etc.) or to describe specific nodes (number of neighbors, number of edges between the neighbors, clustering coefficient [5], etc.).

Therefore it is possible to consider an evolving network as a time sequence G_t of networks (snapshots) and to study each of these independently. This yields for each parameter a time series which can be studied using signal processing notions (see Fig. 1). Properties such as the mean, standard deviation and other statistical properties can be computed on these time series.

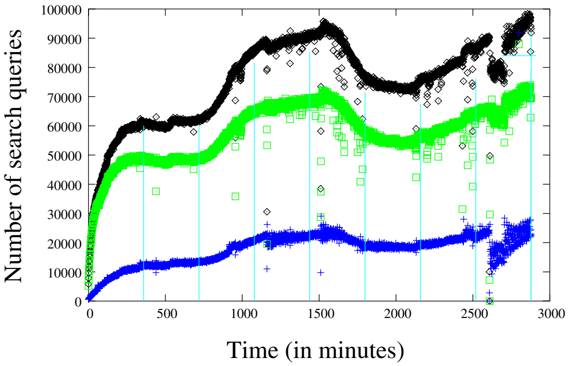


Figure 1. Number of queries per minute during 50 hours on a small size P2P Edonkey server. The three curves correspond to different types of queries. Day/night effects can be observed as well as the start of the measurement during which a lot of new peers connect which yields an increasing number of queries.

A more complex property is the autocorrelation function for a quantity X: $C_X(\tau) =< X(t+\tau)X(t) >_t - (< X(t) >_t)^2$, where $< \cdot >_t$ is the mean over time.

From this, we can extract a *correlation time* [6], defined as the first time were the function $C_X(\tau)$ equals zero (it always happens due to the summation rule of empirical C_X). The correlation time quantifies the "memory" of the property: the longer it is, the greater are the persistence of fluctuations in the data.

Note that very often data are not given as a sequence of snapshots but rather as a sequence of events: email networks for instance are defined by a set of triples (from,to,date), the date being the moment when the mail was sent. If one considers a snapshot every second, since it is likely that no two events (two emails sent exactly at the same time) happen simultaneously then the observed networks are very small. On the contrary, if the aggregation is done on a larger scale (every minute, hour or day) more events are to be observed on each snapshot but the temporal order of these events is going to be lost in each snapshot (mail replies or forwards for instance). In complex networks, different time scales can be used depending on the parameter or phenomenon observed. For instance when considering a typical P2P system, one can study the instantaneous throughput (one second or less), connection duration of a peer (minutes to days), download duration of a file (minutes to weeks) or even the duration during which a file is available on the network (up to years), etc.

2. Definition of dynamic properties

Considering the evolution as a sequence of snapshots is an efficient and simple approach in many cases but some properties cannot be directly observed in this framework. For instance it is natural to look at the duration of contacts or non-contacts between individuals in a network [7] or to study the evolution of communities. For these simple examples, one has to consider the evolution of the network from one time step to the next, or the whole evolution. Hereafter we detail the case of the evolution of communities in a network.

Communities are defined as dense subgraphs with few edges between them and can be found in many complex networks. The identification of such subgraphs is important in many contexts since such communities can correspond to groups of friends or people with similar interests, web pages with a similar content, etc. Moreover studies show that information (rumors for instance) spread more rapidly inside communities than between communities.

Many algorithms are available to find communities automatically on graphs, however theses methods are often time expensive and very sensitive to small modifications of the topology: the addition of one edge can have strong implications on the global community structure. Therefore it is likely that applying these methods will produce completely different decompositions for each snapshot. One approach has been used in [4] using a non classical definition of communities which allows to follow the evolution of community using a simple set of rules (birth, death, merge, split, growth and contraction). Similar ideas are presented in [8] by the identification of dense subgraphs in each snapshot, the subgraphs begin merged afterwards. In [9], the authors present an approach not specifically dedicated to the identification of communities but to the clustering problem in general which allows to cluster data in a timely fashion while keeping a good clustering and no strong variations from one snapshot to the next. Approaches using tools from data-mining are also available, which allow to compute dense sets of nodes with many interactions for a long period of time [10,11] (see Fig. 2).

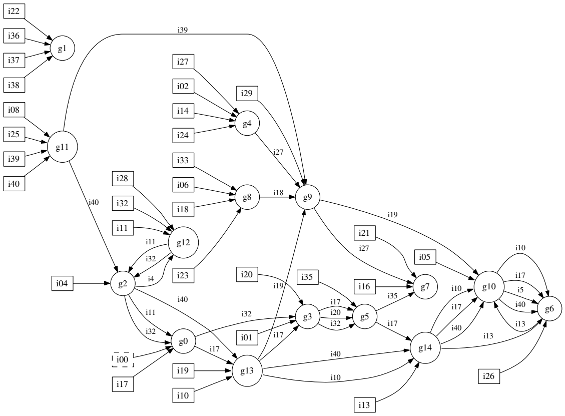


Figure 2. Time ordered trajectories of individual (square) in groups (circles) in a contact network. Groups are dense connected subgraphs which appear frequently in the evolving network.

The results obtained using any community detection algorithm for evolving network give some information on the communities (lifetime, rate of apparition and disappearance, probability of merging and splitting, etc.) [11]. The study of the evolution of the network can therefore be done at a different scale which is not local and not global.

3. Study of specific users and phenomenon

Another approach which can be used to study evolving networks is to study specific nodes or groups of nodes of particular interest (for instance communities). In many contexts algorithms and protocols are designed for average users and it is important to know the number of users who are significantly deviant from this average and how they behave in order to optimize protocols. For instance in most P2P systems, the load for a peer is somehow proportional to the number of files shared, users which share many files can therefore become bottlenecks if they are queried too often. In Fig. 3 we show results obtained on a typical P2P network when trying to identify the users who share many files. These users are more likely to be queried very often by other peers.

3.1. Transmission of information

A typical phenomenon on complex networks is the diffusion such as for viruses in epidemiology, routing in computer networks, innovation, etc. If recent studies have taken into account the real dynamics [12,13], for most of them the process of transmission is living on a static graph. In the static case the main issues are to find networks parameters which explain the persistence of viruses within a given graph. It has been shown for instance that there is a strong relation between the largest eigenvalue of the adjacency matrix of the network and the epidemic threshold [14].

Dynamics are also central in new communication services which are relying on mobile users spread in the environment. The routing of information in such a context depends on the connectivity between nodes and the mobility of these nodes. Understanding the characteristics of these networks (called Delay Tolerant Networks) is therefore cru-

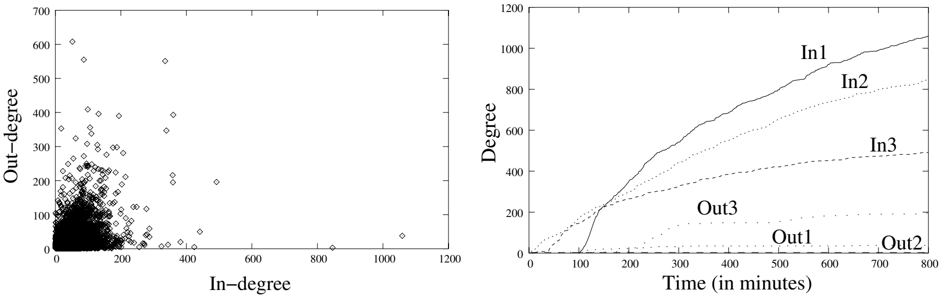


Figure 3. Left: joint distribution of the number of files offered by a peer (in-degree) versus the number of files he looks for (out-degree) in a P2P system. Peers offering many files and peers who do not offer any file but look for many (free-riders) can be easily identified and further studied. Right: evolution of the number of files requested for the three peers offering the more files (the three rightmost on the left figure). After 13 hours some of their files still have not been requested.

cial to propose protocols suitable for this context. The simplest ways to transmit data in this context are the opportunistic forwarding algorithms [7]: when a node needs to send some data to a destination, it uses its contacts to relay the data to the destination. Two naive algorithms belong to this class, the first one consists in waiting to be connected to the destination to send the data directly, the second consists in forwarding the data to all the neighbors which in turn are going to pass it to their neighbors. This is going to flood the network with the data which eventually should arrive at the destination (see Fig. 4).

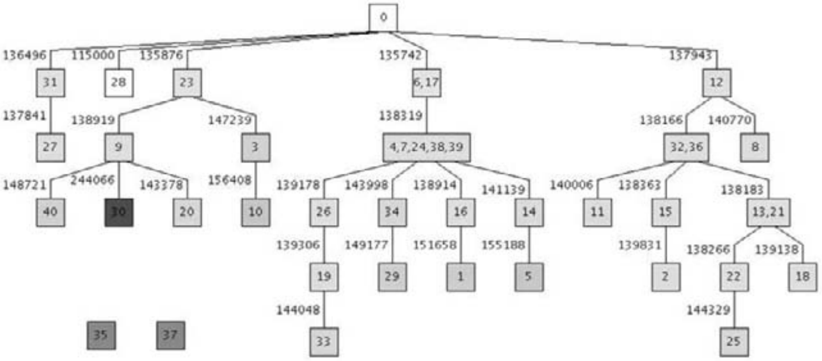


Figure 4. Naive instantaneous flooding in a mobile network, where node 0 is trying to send a message to all other nodes at time 115000. Time on the edges represents the earliest time for the message to be transmitted to the group of nodes below, *i*.e. the time when an edge is created between a node on top and a node at the bottom. Note that two users (35 and 37) cannot be reached. The data used for the flooding simulation are described in [11]

4. Conclusion

We presented here three different approaches which can be used to study an evolving network:

- the evolving network can be considered as a sequence of snapshots and each of these snapshots can be studied as a static network;

- properties can be defined on the evolution itself, for instance the duration of contacts in a network or the evolution of communities through time ;
- finally, specific users or phenomena can be studied, the more obvious being the diffusion of information in an evolving network.

Many studies have been focused on static networks, therefore the first approach is the more developed, however the definition of proper time scales is a unsolved problem and there is no warranty that such time scales can be defined in an automatic way given an evolving network. For both other approaches, much work has to be done in order to define new relevant parameters to describe the evolution as precisely as possible.

Finally, using all the parameters obtained with the previous approaches would allow the introduction of evolutionary models for dynamic complex networks. Such models could be used in formal contexts and for simulation purposes. Defining random models is not an easy task and even for static networks some simple parameters cannot be captured in a satisfactory way. A few models have already been introduced (see for instance [15]) which are modifying a given network by the addition of nodes and edges, however the aim is in general not to generate an evolving network but to eventually obtain a network with a given set of static properties. Much therefore remains to be done in this direction.

References

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics 74, 47*, 2002.

[2] S.N. Dorogovtsev and J.F.F. Mendes. *Handbook of Graphs and Networks: From the Genome to the Internet*, chapter Accelerated growth of networks. Wiley-VCH, 2002.

[3] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *ACM SIGKDD*, 2005.

[4] G. Palla, A. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, April 2007.

[5] J.D. Watts and S.H. Strogatz. Collective dynamics of'small-world'networks. *Nature*, 393(6684):409–10, 1998.

[6] H. Abarbanel. *Analysis of Observed Chaotic Data.* Springer, 1996.

[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *INFOCOM*, 2006.

[8] Y. Chi, S. Zhu, X. Song, J. Tatemura, and B.L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172. ACM Press, 2007.

[9] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560. ACM, 2006.

[10] J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome. Constraint-based concept mining and its application to microarray data analysis, 2005. Intelligent Data Analysis 9(1):59-82, 2005.

[11] E. Fleury, J.-L. Guillaume, C. Robardet, and A. Scherrer. Analysis of dynamic sensor networks: Power law then what? In *Second International Conference on COMmunication Systems softWAre and middlewaRE (COMSWARE 2007)*, Bangalore, India, 2007. IEEE.

[12] J. Leskovec, L.A. Adamic, and B.A. Huberman. The dynamics of viral marketing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 228–237. ACM Press, 2006.

[13] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs, 2007.

[14] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *22nd Symposium on Reliable Distributed Computing*, 2003.

[15] R. Albert, H. Jeong, and A.L. Barabasi. The diameter of the world wide web. *Nature*, 401:130–131, 1999.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-204

Mining Networks through Visual Analytics: Incremental Hypothesis Building and Validation

David AUBER [a], Romain BOURQUI [a] and Guy MELANÇON [a,1]
[a] *INRIA Bordeaux Sud-Ouest / CNRS UMR 5800 LaBRI*
France

Abstract. Interactive visualization supports the analytical process: interacting with abstract views, using the data as a malleable material, analysts build hypothesis that can be further validated on the whole dataset. We use graph clustering in order to group elements and show them as meta-nodes and reduce the number of visual items, further organizing data over into several layers, in an effort to provide a multilevel model of the studied phenomenon. In doing so, hierarchical clustering not only contributes to the study of data but also brings in ingredients for interaction, enabling the user to zoom in and out of clusters while exploring the data in quest for evidence.

Keywords. Visual Analytics, Graph Mining, Information Visualization, Interactive Exploration

Introduction

Networks are everywhere. Social networks: people know each other, people meet, people exchange information. People rely on third parties belonging to different organizations, people link organizations. Events first group into sequences and then into causal networks when people collaborate. Semantic networks: documents link because they share content or authors, or through citations. Concepts and ideas link through documents - people create, access, modify and share documents. Analysts are faced with massive collections gathering documents, events and actors from which they try to make sense. That is, they search data to locate patterns and discover evidence. Interactive exploration of data has now established as a fruitful strategy to tackle the problem posed by this abundance of information. The Visual Analytics initiative promotes the use of Information Visualization to support analytical reasoning through a sense-making loop based on which the analysis incrementally builds hypotheses (Figure 1).

"Information Visualization" supports the visual exploration and analysis of large datasets by developing techniques and tools exploiting human visual capabilities – ac-

[1] Corresponding Author:
INRIA Bordeaux Sud-Ouest / CNRS UMR 5800 LaBRI
Campus Université Bordeaux I
351 Cours de la Libération
33405 Talence Cedex – FRANCE; E-MAIL: GUY.MELANCON@LABRI.FR

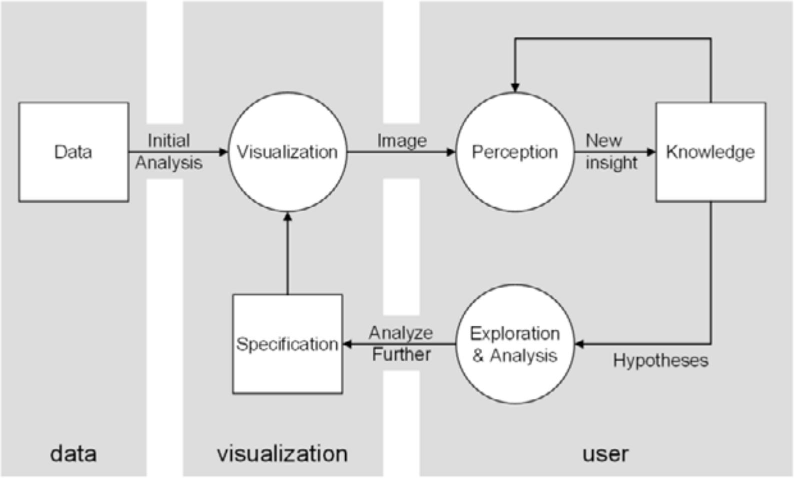


Figure 1. The “sense making loop” (adapted from [1] – see also [2]).

cording to [3], 40% of our cortical activities are dedicated to processing visual signals. The design of new visualization methods and tools becomes even more necessary with the continuously increasing volume of available data, which poses a problem that obviously cannot be solved by relying solely on the increase of CPU power.

Visually mining data requires combining data analysis with visual graphics and interaction. Mining itself draws not only on statistics but on a rather astute mixture of mathematical rigor and heuristic procedures. As David Hand puts it [4] [5]:

> “*To many, the essence of data mining is the possibility of serendipitous discovery of unsuspected but valuable information. This means the process is essentially exploratory.*”

From Hand’s perspective, we see that information visualization has much to share with data mining because visualization often comes as an aid to exploratory analysis. The perspective we adopt is a combination of (semi) automated data processing together with human analytical and perceptual capabilities. Although relying on technology, the analysis task remains in total control of the human user. The National Visual Analytics Center (NVAC) research agenda [2] clearly states:

> “[The] *analysis process requires human judgment to make the best possible evaluation of incomplete, inconsistent, and potentially deceptive information* [. . .]”,

later calling for the development of

> “[. . .] *visually based methods to support the entire analytic reasoning process . . . .*”

That is, in ideal cases the visualization should be designed in order not only to assist the analysis but to also actively contribute to its progress. Visual Analytics thus appears as a multi-disciplinary field embracing a large spectrum of competences. This partly comes from the need to cover all processes involved in the so-called "Visualization pipeline" as depicted by dos Santos and Brodlie [6] (Figure 2):

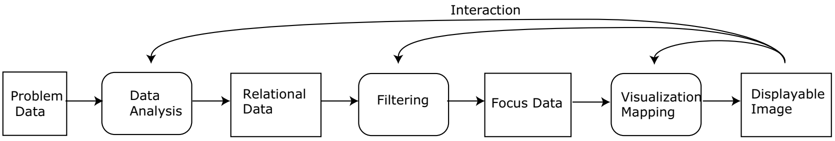


Figure 2. Visualization pipeline (adapted from [6]).

1. Interaction, Scalability, Graph Hierarchies

Visual Analytics obeys Daniel Keim's mantra "*Analyse first – Show the Important – Zoom, Filter and Analyse Further – Details on Demand*" . Showing the important can be understood in several different manners; what remains essential here is to enable the user to dynamically build views from the original dataset under study. For example, the graph shown in Figure 3 has been extracted from a NCTC dataset (consisting of 10 000 nodes and 20 000 edges) to help locate collaboration between terrorists groups based on territorial activity. Continents have been inserted into the graph to help organize the overall layout. Drawing the whole graph with tens of thousands of elements on the screen is pointless, as the resulting drawing obviously lacks readability. Once this smaller graph has been drawn and explored, a second iteration can be designed based on a different point of view, building on previous observations. The overall strategy is thus to filter out data elements and build sequences of "virtual" or "partial" graphs, in an attempt to "see" what can possibly be present in the data.

The visualization supports the analytical process: ultimately, the analyst will build hypothesis that can be further validated on the whole dataset. It is important to note that this incremental methodology uses the data as a malleable material. The analysis of "virtual" graphs offering partial views on the dataset can be useful in bringing structural properties upfront. Indeed, having established that the graph under study is small world, or scale-free, can trigger various scenarios to further analyze the data.

2. Hierarchical clustering

The number of data elements drawn on the screen must necessarily be kept small, for sake of readability. A layout of a graph containing a thousand nodes is already hard to read, depending on its structural properties (a tree is much easier to read than a totally random graph, of course). A common strategy is then to use graph clustering in order to group elements and show them as meta-nodes and reduce the number of visual items. Promising approaches develop techniques based on the intuitive notion of intracluster density versus intercluster sparsity. Methods based on the computations of node or edge centralities [7] [8] have proved successful [9] [10]; other approaches based on local indices (edge strength [11] [12] or Burt's constraint on edges and nodes [13], for instance) have also been suggested.

Typically, hierarchical clustering will organize data over several layers. In ideal cases, these layers will contribute to model the data under study as layers reveal structure. Graph hierarchies appear as an adequate formalization capturing the notion of multiscale communities in a network (network of networks). Current studies confirm the absolute presence of hierarchies either in nature itself or in abstract human construction such as

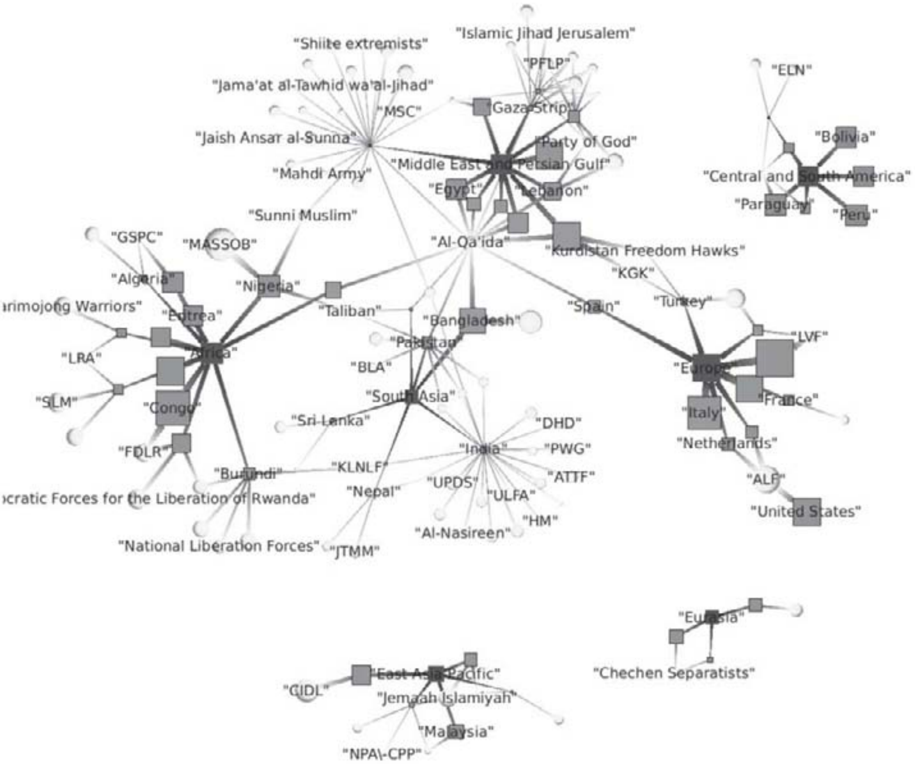


Figure 3. The drawing of a graph sketching the relationships among terrorist groups and countries. Continents are further inserted to organize the layout.

language [14]. Current evolutionary models in biology try to capture the multilevel nature of networks formed by various biological entities [15] [16], just as with cities and city systems in geography.

Also, hierarchical clustering not only contributes to the study of data but also brings in ingredients for interaction. Our methodology for studying large small world networks relies on graph hierarchies (nested subgraphs) as a central paradigm. For example, Figure 4 displays a hierarchy of subgraphs computed out of a network of keywords extracted from documents (keywords are linked based on a similarity measure; a graph is then induced by thresholding the similarity matrix). As can be seen on the figure, the top level graph consists of meta-nodes themselves containing a lower level hierarchy of graphs. This visual representation can then be zoomed in and out in order to explore more focused regions of the graph. The top level graph shows the overall organization of the network. Meta-nodes are connected to one another according to connections between the nodes they contain.

Once a graph hierarchy has been computed from the original dataset, it can be used to compute various statistics on data elements – either to assess the relevance of the hierarchy or simply to explore properties of low level nodes. Guimera et al. [17] define the z-score and participation coefficient for nodes $v \in V$ in a clustered graph $(G; \mathbf{C})$ where $G = (V, E)$ and $\mathbf{C} = \{C_1, \dots, C_k\}$ is a partition of V.

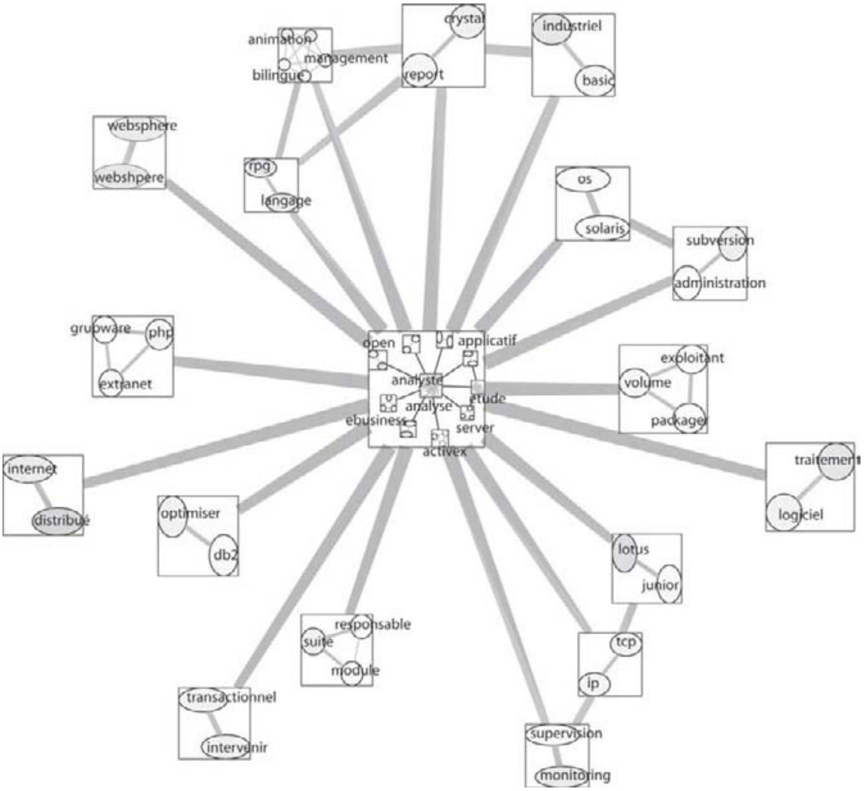


Figure 4. A hierarchy of subgraphs computed out of a network of keywords extracted from documents. The top level graph consists of meta-nodes themselves containing a lower level hierarchy of graphs.

$$
\begin{aligned}
z_{\mathbf{C}}(v) &= \frac{d_G(v) - \bar{d}_G}{\sigma_{d_G}} \\
p_{\mathbf{C}}(v) &= 1 - \sum_j \left(\frac{d_{C_j}(v)}{d_G(v)}\right)^2
\end{aligned}
\qquad (1)
$$

where $d_G(v)$ denotes the degree of v (in G), $\bar{d}_G$ and σ_{d_G} denote the mean degree and standard deviation (in G) and $d_{C_j}(v)$ denotes the number of neighbors of v belonging to cluster C_j. The z-score roughly translates the mean value to 0 and normalizes the standard deviation to 1; this is just the usual transformation performed on normal or gaussian distribution. The z-score corroborates an individuals' dynamic within its own community. It must be noted however that the score does not translate into a probability or exceptionality of a value, since we cannot assume that the degree distribution follows a gaussian distribution. In fact, the degree distribution most probably follows a power law, which is typical of all scale-free graphs.

The participation coefficient of a node indicates how much it covers all other communities. When dealing with weighted graphs, one must however take the weighted degree of nodes into account. More precisely, let $\omega(e)$ denote the weight associated with an edge $e = \{u, v\}$, and define the weighted degree of a node u as the sum $d_\omega(u) = \sum_{v \in N(u)} \omega(u, v)$ of weights of all edges incident to u. The weighted participation coefficient p_ω then extends to weighted graphs by taking the weighted degree instead of the usual plain degree of nodes. Now, observe that two nodes u, v may appear as participating equally $p_\omega(u) = p_\omega(u)$ while their weighted degree might greatly differ $p_\omega(u) >> p_\omega(u)$. This can however be disturbing when comparing the relative roles of nodes.

This problem can be solved by looking at the overall participation of a cluster with other clusters, and by assigning a node a part of its cluster participation according to its relative weight (with respect to the cluster it belongs to). This implies we can define a participation coefficient of nodes on a flat graph (no clusters). Going back to a weighted graph $H = (W, F, \omega)$, we define the *flat participation coefficient* of a node $w \in W$ as

$$p(w) = 1 - \Big(\sum_{x \in N(w)} \frac{\omega(w, x)}{d_\omega(w)} \Big)^2.$$

Let now $\mathbf{C} = (C_1, \ldots, C_k)$ be a clustering of a weighted graph $G = (V, E, \omega)$. The flat participation coefficient can be computed on the quotient graph $G/\mathbf{C}$ where each cluster appear as a node (and where weights are induced from ω on edges between clusters. The contribution c_ω of a node $u \in V$ relative to its cluster C_i can then be computed as a ratio of the flat participation coefficient $p(C_i)$:

$$c_\omega(u) = \frac{d_\omega(u|C_i)}{d_\omega(C_i)}$$

where $d_\omega(C_i)$ denote the degree of C_i seen as a node in the weighted quotient graph and $d_\omega(u|C_i)$ denote the degree of $u \in V$ restricted to edges in E connecting with C_i.

When dealing with multiscale networks, the computation of the z-score and participation coefficient of individuals and communities, at various levels, reveals how the network's dynamic build through scales. Appropriate visual cues help locate key actors, pointing at individuals or communities as hubs, bridges or satellite.

3. Tulip

Figures 3 and 4 show views of graphs computed with the help of the Graph Visualization Framework Tulip[2] [18] [19] developed by our team includes graph hierarchies as a central navigation mechanism and data structure. The Tulip architecture actually mimics the pipeline shown in Figure 2 as do most Information Visualization software systems. However, the internal data structure of Tulip as been optimized as to directly deal with graph hierarchies avoiding the duplication of nodes and subgraph properties. Variants of hierarchical clustering algorithm, graph statistics or colormap schema, for instance, can readily be implemented as plug-ins and used on the spot. The overall architecture ex-

[2] Tulip is Open Source and distributed under GPL. See the URL `tulip.labri.fr`

ploits all capabilities of graphics hardware and C++, making it one of the most powerful publicly available graph visualization framework.

4. Conclusion and future work

We plan to extend the use of graph statistics, clustering and graph hierarchies to develop strategies for the visual analysis of dynamically evolving networks, using graphs as a visual metaphor for supervising evolving information space. Tracking the evolution of outlier nodes or clusters should help analysts identify weak signals and confirm specific or general tendencies.

References

[1] Jarke J. van Wijk. The value of visualization. In C. Silva, E. Groeller, and H. Rushmeier, editors, *IEEE Visualization*, pages 79–86. IEEE Computer Society, 2005.

[2] James J. Thomas and Kristin A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, 2006.

[3] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.

[4] David J. Hand. Statistics and data mining: intersecting disciplines. *ACM SIGKDD Explorations Newsletter*, 1(1):16–19, 1999.

[5] David J. Hand. Strength in diversity: The advance of data analysis. In Jean-François Boulicaut, Floriana Esposito, Fosca Gianotti, and Dino Pedreschi, editors, *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD 2004*, volume 3202 of *Lecture Notes in Artificial Intelligence*, pages 18–26, Pisa, Italy, 2004. Springer-Verlag New York, Inc.

[6] Selan dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311–325, 2004.

[7] Linton C. Freeman. A set of measures of centrality based upon betweeness. *Sociometry*, 40:35–41, 1977.

[8] Ulrik Brandes and Thomas Erlebach, editors. *Network Analysis*, volume 3418 of *Lecture Notes in Computer Science*. Springer, 2005.

[9] Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physics Reviews E*, 69(026113), 2004.

[10] Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy Science USA*, 99:7821–7826, 2002.

[11] David Auber, Yves Chiricota, Fabien Jourdan, and Guy Melançon. Multiscale navigation of small world networks. In *IEEE Symposium on Information Visualisation*, pages 75–81, Seattle, GA, USA, 2003. IEEE Computer Science Press.

[12] Yves Chiricota and Guy Melançon. Visually mining relational data. *International Review on Computers and Software*, May 2007 issue, 2007.

[13] Ronald S. Burt. *Brokerage and Closure*. Oxford University Press, 2005.

[14] Bruno Gaume, Fabienne Venant, and Bernard Victorri. Hierarchy in lexical organization of natural language,. In D. Pumain, editor, *Hierarchy in natural and social sciences*, Methodos series, pages 121–143. Springer, 2005.

[15] Allesandro Vespignani. Evolution thinks modular. *Nature*, 35(2):118–119, 2003.

[16] Alain Pavé. Hierarchical organization of biological and ecological systems. In D. Pumain, editor, *Hierarchy in natural and social sciences*, Methodos series, pages 39–70. Springer, 2005.

[17] Roger Guimera‘, S. Mossa, A. Turtschi, and Luis A. Nunes Amaral. The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(22):7794–7799, 2005.

[18] David Auber. Tulip - a huge graph visualization framework. In Petra Mutzel and Mickael Jünger, editors, *Graph Drawing Software*, Mathematics and Visualization Series. Springer Verlag, 2003.

[19] Maylis Delest, Tamara Munzner, David Auber, and Jean-Philippe Domenger. Exploring infovis publication history with tulip (2nd place - infovis contest). In *IEEE Symposium on Information Visaulization*, page 110. IEEE Computer Society, 2004.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-212

A Review of Anomaly Detection on Graphs

Kiriaki PLATANIOTI [a,1], Nicholas A. HEARD [b] and David J. HAND [b]
[a] *The Institute for Mathematical Sciences, Imperial College London, U.K.*
[b] *Department of Mathematics, Imperial College London, U.K.*

Abstract. Anomaly detection on graphs of social or communication networks has important security applications. The definition of a graph anomaly typically depends on the data and application of interest. Usually central in anomaly detection are the connections amongst the graph's entries and various methods have been developed for their analysis. We review these techniques and also discuss challenges currently faced for high-dimensional dynamic graphs.

Keywords. Social networks, graphs, anomaly detection

1. Introduction

In mathematics, graphs are considered to be the most appropriate data form for representing a network, e.g. phone or computer networks, or the World Wide Web. A graph consists of a set of nodes $\{v_i\}_{i=1}^n$, which denote the network's members, and a set of edges, E, containing any relational information known about the members. Graphs can also incorporate information on node labels, e.g. malicious or not and the type of edges; directed/undirected or labelled. The adjacency matrix, W, defines the strength and direction of association between the nodes and characterises the graph. In dynamic networks the members and structure change over time and W is time-evolving.

Anomaly detection identifies rare/irregular substructures in the links and attributes of the nodes. Depending on the nature of data and application, anomalies can be: irregularly connected substructures, e.g. small cliques of nodes displaying excessive connectivity, abrupt changes in the evolution of the network's dynamics, e.g. sudden flurry of connections, seemingly hidden/concealed edges or changes in the attributes of nodes.

2. Existing Methods and Techniques

A variety of techniques have been proposed for the analysis of graphs, one of the oldest being spectral methods. The eigenvalues and eigenvectors of W, or of the graph Laplacian, $L = I - D^{-1}W$, where D is diagonal and $D_{ii} = \sum_{j=1}^n W_{ij}$, are used to extract information. The multiplicity of the zero eigenvalue of L equals the number of non-

[1] Corresponding Author: Kiriaki Platanioti, The Institute for Mathematical Sciences, 53 Prince's Gate, South Kensington, London, SW7 2PG, U.K.; E-mail: kiriaki.platanioti99@imperial.ac.uk.

interacting groups of nodes in the graph. The eigenvectors of the smallest eigenvalues of L may also be utilised to cluster the nodes; [1]. In dynamic settings, the maximal eigenvector u of W characterises the activity of the network and thus, monitoring u over time can lead to the detection of sudden/abnormal changes in the network; [2]. However, the application of the methods for large n is hard, unless W is sparse.

Stochastic approaches such as random walk methods have also been used in the analysis of graphs. A Markov transition matrix $P = D^{-1}W$ is introduced to traverse the graph from a given starting node v_i; [3]. The steady-state probability of the chain visiting node v_j from v_i defines the relevance score (RS) of v_j to v_i and possible anomalies occur as pairs of linked nodes with low RS. The method applies to high dimensional sparsely connected graphs [4], but is not well-suited in dynamic cases. For large sparse dynamic graphs, another randomised method which approximates W over time has been recently developed; [5]. This allows sudden connection outbreaks to be detected in dynamic setups. For dynamic graphs of moderate size, scan statistics have also been proposed; [6].

In social network analysis, early algorithms labelled the nodes as authorities or facilitators; [7]. Although such methods have obvious security applications, they are hard to extend to evolving graphs. For dynamic settings, with known labels for a given set of nodes, new entries can be labelled according to distance metrics based on their connections and degree of associativity; [8]. In the same spirit, recent Bayesian approaches probabilistically predict links between nodes using information from node attributes; [9]. The basis of Bayesian approaches shows more promise for applications in dynamic graphs.

3. Conclusions and Future Directions

Existing approaches form a foundation for the analysis of graph data. However, they have been mostly developed for specific data structures and are mainly applicable to non-dynamic networks of moderate size. Methods for rapidly scanning large, evolving graphs are required as well as statistical techniques which will allow genuine anomalies to be distinguished from random irregularities arising by chance.

References

[1] U. von Luxburg, A tutorial on spectral clustering, *Stat Comp* **17** (2007), 395–416.

[2] T. Idé and H. Kashima, Eigenspace-based anomaly detection in computer systems. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004).

[3] J. Sun, H. Qu, D. Chakrabarti and D. Faloutsos, Relevance search and anomaly detection in bipartite graphs, *SIGKDD Explorations* **7** (2005), 48–55.

[4] H. Tong, C. Faloutsos and J.-Y. Pan, Fast random walk with restart and its applications. In: Sixth IEEE International Conference on Data Mining (2006).

[5] J. Sun, Y. Xie, H. Zhang and C. Faloutsos, Less is more: compact matrix decomposition for large sparse graphs. In: Proceedings of the 2007 SIAM International Conference on Data Mining (2007).

[6] C. E. Priebe, J. M. Conroy, D. J. Marchette and Y. Park, Scan Statistics on Enron Graphs, *Comput Math Organ Theory* **11** (2005), 229–247.

[7] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, *Journal of the ACM* **46** (1999), 604–632.

[8] C. Cortes, D. Pregibon and C. Volinsky, Computational methods for dynamic graphs, *J Comput Graph Stat* **17** (2007), 395–416.

[9] C. J. Rhodes and E. M. J. Keefe, Social network topology: a Bayesian approach, *J Oper Res Soc* **58** (2007), 1605–1611.

This page intentionally left blank

Text Mining

This page intentionally left blank

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-217

Using language-independent rules to achieve high multilinguality in Text Mining

Ralf STEINBERGER[1], Bruno POULIQUEN and Camelia IGNAT
European Commission – Joint Research Centre
21020 Ispra (VA), Italy

Abstract. A lot of information is hidden in foreign language text, making multilingual information extraction tools – and applications that allow cross-lingual information access – particularly useful. Only a few system developers offer their products for more than two or three languages. Typically, they develop the tools for one language and then adapt them to the others. Information on guidelines how to produce highly multilingual applications with the least possible effort is scarce. Due to the multinational character of the European institutions, the Text Mining applications developed at the *Joint Research Centre* (JRC) had to be planned with the multilinguality requirements in mind. The purpose of this chapter is to present the design principles behind the in-house developments and to show a number of different applications that are built according to these guidelines. The results of the text analysis applications presented here are visible on the publicly accessible multilingual news gathering, analysis and exploration tool NewsExplorer.[2]

Keywords. Design principles; multilinguality; language-independence; geotagging; named entity recognition; morphological name variation; name variant mapping; quotation recognition; date recognition; cross-lingual document similarity calculation.

1. Introduction

Text Mining generates valuable meta-information for texts that can be used for a variety of applications. These include the highlighting of relevant information in documents and storing the extracted meta-information to provide users with powerful search functionality that goes much beyond that of full-text search. As a lot of information is available in some languages and not in others, there is a clear benefit to multilingual information extraction. However, the development of text mining tools for new languages is typically rather time-consuming, which explains why most systems only cover one or a small number of languages. The purpose of this chapter is to present some simple, successfully applied design principles that make it easier to extend text analysis tools to many languages. They can be roughly summarised as follows: By combining universal, i.e. language-independent rules with relatively simple language-specific resource files, it is possible to extend the language coverage of text analysis applications quickly. Secondly: when *cross-lingual* applications need to be developed for a large number of language pairs, it is important to represent monolingual contents in a language-neutral way so as to avoid the need for language pair-specific resources.

[1] The email address of the corresponding author is Ralf.Steinberger@jrc.it.
[2] NewsExplorer currently covers 19 languages. It is publicly accessible at http://press.jrc.it/NewsExplorer.

Before describing these design principles in a bit more detail (Section 1.3), we will first remind the reader of the various benefits of using text mining tools (1.1) and give some concrete examples of cases where multilingual information extraction and aggregation tools provide more information than monolingual ones (1.2). Section 2 on related work will give an overview of existing multilingual and cross-lingual applications and of the few design principles mentioned by other system developers regarding multilinguality. This is followed by short descriptions of some Text Mining applications developed according to the principles presented here: the recognition and disambiguation of references to geographical places (3) and of persons and organisations (4); a light-weight approach of dealing with inflection and other regular word variations (5); a method to identify name variants across many languages and writing scripts (6), a tool to recognise quotations (7) and one to recognise and disambiguate dates (8). Section 9 then shows how various monolingual text analysis tools can be combined to link related documents across languages without the need of language pair-specific resources. Avoiding language pair-specific components is the essential requirement when the objective is to link related documents in many different language pair combinations. Due to the wide variety of linguistic phenomena, identifying language-independent rules and applying them to many languages is not trivial. Section 10 lists a few linguistic phenomena that made it necessary to adapt the generic rules, or to allow exceptions. A short summary concludes the chapter (11).

1.1. Benefits of using text mining tools

The usage of software for named entity recognition and classification (NERC), which can identify references to persons, organisation, locations and other entities, has a number of benefits for the information-seeking user, including the following:

- Detection of relevant information in large document collections and – on demand – highlighting of the found words or phrases in the text;
- Display of all the extracted information for a larger, monolingual or multilingual, document collection, serving as a multi-document summary and allowing users to narrow down the search by selecting some extracted information features;
- Advanced search functionality operating on meta-information, allowing, for instance, to search for person names mentioned in documents that mention a given country and a given person in a certain time period;
- Automatic inference: a text mentioning, e.g., the city of *Izmir* can be automatically marked as being about Turkey, even if the country is not mentioned explicitly;
- Visual summary: for document collections, in which locations have been identified and disambiguated, maps can give an overview of the geographical coverage;
- Geographical search: find all documents mentioning a place in a certain country, or within a certain radius of another location;
- Name variants: If spelling variants for the same person have been automatically detected, users can search for mentions of a person independently of the spelling;
- Cross-lingual information access: for instance, display in one language information extracted from texts in other languages;
- Linking of similar documents across languages: e.g. by virtue of the fact that they talk about the same entities;
- Visualisation of information: e.g. to show social network graphs with relations between people, or allow to navigate document collections via a document map.

This non-exhaustive list gives an idea of the power of text analysis applications and of the benefits users can reap by using them.

1.2. Practical arguments for multilinguality

The functionalities listed here are mostly available for the nineteen languages currently covered by the NewsExplorer application, which has been developed according to the basic principles described in this chapter. While many of these functionalities are applicable to information extracted from texts in a single language, the usefulness rises significantly if the information is derived from documents in many different languages. In NewsExplorer, which analyses an average of 35,000 articles gathered from about 1,300 news portals world-wide in 19 languages, about a quarter of the news articles are written in English. However, there is ample evidence that much of the information would not be found if only English texts were analysed. Some examples are:

- Some areas of the world are better covered by different languages. For instance, North Africa is covered better by French, Brazil by Portuguese, and the rest of Latin America by Spanish. By analysing texts in more languages, more information will be found.
- Daily and long-term social network analysis has shown ([1], [2]) that the dominant and most frequently mentioned personalities in English language news are the US President and the British Prime Minister. When looking at French, German, Arabic or Russian news, the respective leaders have a much more central role. Adding analysis results from more languages reduces bias and increases transparency.
- NewsExplorer automatically collects name attributes (titles, professions, age, role, nationality, family relations, etc.) about the approximately 700,000 persons found in media reports in the course of several years from the news in different languages. The various attribute lists show clearly that the information found across languages is often complementary. To give a concrete example: For the murdered Russian spy Alexander Litvinenko, texts in most languages contained the information that Litvinenko was a Russian and that he was an agent. However, we found in French news that he was 43 years of age, in Italian news that he was killed, in German news that he was a *critic* of some sort, etc.[3]

In a national context with a dominant language, for instance English in the USA, a possible solution to this multilinguality requirement is to use machine translation (MT) to translate the documents written in the most important foreign languages into the national language, and to then apply the text analysis tools to the texts in that language. In the European context, where there is not one, but 23 official languages, using MT is not an option, as there are 253 language pairs (`N*(N-1)/2`, with N being the number of languages involved). Multilingual text processing is generally important, but it is considerably more relevant in the European context than in most other settings. Another important issue is the fact that MT often fails when being confronted with proper names or specialist terminology ([3]). Larkey et al.'s *native language hypothesis* ([4]) – the observation that text analysis results are better if performed on the source language text – is also a good argument for applying multilingual text analysis rather than operating on machine-translated texts.

[3] See http://press.jrc.it/NewsExplorer/entities/en/97338.html for more details regarding this example.

1.3. Design principles making it easier to achieve high multilinguality

There is more than one way to achieve multilinguality and Section 2 shows that there are other systems that cover several languages, but only few developers have explicitly mentioned the principles behind their development. The guidelines underlying the development of the NewsExplorer application are the following:

- Use language-independent rules wherever possible, i.e. use rules that can be applied to any new language that will be added to the system.
- Keep language-specific resources and tools to a minimum: Such tools are, for instance, linguistic software like part-of-speech taggers and parsers. Language-specific resources are morphological or subject domain-specific dictionaries, linguistic grammar rules, etc. As acquiring or developing such language-specific resources is difficult and expensive, it is better not to use them, or – where they are necessary – to use them as little as possible.[4]
- Keep the application modular by storing necessary language-specific resources outside the rules, in language-specific parameter files. That way, new languages can be *plugged in* more easily.
- For the language-specific information that cannot be done without, use bottom-up, data-driven bootstrapping methods to create the monolingual resources.
- Avoid language pair-specific resources and procedures because the almost exponential growth of language pairs would automatically limit the number of languages a system can deal with.

These are very simple and generic guidelines, but sticking to them is not always easy and sometimes comes at a cost. Sections 3 to 9 will describe several applications of different types that each try to follow these guidelines.

2. Related work

The intention of this chapter is to specifically address the issues surrounding the development of multilingual and cross-lingual applications involving larger numbers of languages. When discussing related work, we will distinguish multi-monolingual tools, i.e. monolingual applications applied to more than one language (Section 2.1), from cross-lingual tools, i.e. those that involve crossing the language barrier in one way or another (Section 2.2). As the applications described in Sections 3 to 9 are only there to exemplify the guidelines presented in Section 1.3, we will not give a full state-of-the-art overview for each of these tasks.

2.1. Multi-monolingual systems

Applications that mainly use statistical methods, such as search engines, document categorisation tools, Optical Character Recognition (OCR) software or spell checkers, are usually available for larger numbers of languages. The news aggregators *Google News*, *Yahoo News* and *Europe Media Monitor* EMM, for instance, cover 17, 34 and 43

[4] Part-of-speech tagging software is freely available and can, in principle, be trained for any language, but we feel that the effort involved for training 20 languages is too big for the expected benefit. Using readily trained taggers is not an option for us, as the overheads of dealing with different tag sets and levels of performance, and with differing input and output formats, etc. are too big.

languages, respectively,[5] as the crawling of news sites is language-independent and the clustering of related news items can be done with mostly statistical methods. ZyLab's ZyScan® software for OCR claims to cover over 200 languages.[6] For more linguistic applications such as information extraction tools or grammar checkers, the number of available languages is usually much more restricted. Very few companies have produced and sell text analysis tools for a wider range of languages. InXight's information extraction tool set ThingFinder® is an exception, offering the impressive number of thirty languages.[7]

The usage of Machine Learning methods is rather popular as they are promising methods to overcome the bottleneck faced by introspection-based symbolic (rule-based) methods where linguists need to describe the vocabulary and the grammar separately for each language. In Machine Learning approaches to named entity recognition, for example, the idea is to use texts in which named entities (NEs) have been marked-up (usually manually) and to let algorithms learn the features that help to identify new NEs in new texts. Features are, for example, surrounding words, their distance to the NE, their part-of-speech, case information (uppercase vs. lowercase), and more. In order to train tools for new languages, it is thus necessary to create training sets with examples. For improved accuracy, it is furthermore advised to manually confirm or reject the learned rules. Even for Machine Learning methods, the effort required per language is not negligible. Similarly, the development of statistical part-of-speech taggers relies on rather large manually tagged document sets.

For all approaches, the effort of developing monolingual applications for N languages can be up to N times the effort of developing the application for one language, although the complexity of languages differs, of course. Furthermore, there is usually a learning curve involved and the development environment (e.g. the formalism that allows to write the rules) has to be developed only once, so that the effort per language gets less, the more languages are being covered ([5]). The limited number of languages offered by commercial and academic text analysis tools is due to the large language-specific effort required, but by throwing more manpower at the application, more languages can, in principle, be covered.

Multiple authors have described work on developing resources and tools for a number of different languages, but this was typically done by reusing the resources from a first language and adapting them to new languages (e.g. [5], [6], [7], [8]). Tips from various system developers for achieving multilinguality include modularity ([7], [8], [9]), simplicity of rules and the lexicon ([8]), uniform input and output structures ([8], [10]), and the use of shared token classes that are ideally based on surface-oriented features such as case, hyphenation, and includes-number ([8]). SProUT grammar developers took the interesting approach of splitting the multilingual grammar rule files ([8]): some files contain rules that are applicable to several languages (e.g. to recognise dates of the format *20.10.2008*) while others contain language-specific rules (e.g. to cover *20th of October 2008*). The fact that this latter date format and others can also be captured by using language-independent rules with language-specific parameter files will be shown in Section 8. Many people have used the GATE architecture to write grammars and tools for a large variety of languages and there surely are good-practice rules to save effort by reusing existing resources, but we have not been able to find an

[5] See http://news.google.com, http://news.yahoo.com and http://press.jrc.it. (last visited 1.02.2008)
[6] http://www.zylab.com/products_technology/productsheets/ZySCAN.pdf
[7] See http://inxight.com/products/sdks/tf/. (last visited 1.02.2008)

explicit mention of them apart from architectural design issues, the promotion of Unicode and the usage of virtual keyboards to enter foreign language script ([9]).

2.2. Cross-lingual applications

For *cross-lingual* applications, the situation is different. Cross-lingual applications are those that help cross the language barrier and thus involve language *pairs*. Examples are machine translation (MT), cross-lingual question answering, cross-lingual document similarity calculation (CLDS), cross-lingual glossing, cross-lingual topic detection and tracking, name variant mapping across languages and scripts, etc. With state-of-the-art methods, the effort to produce any such software is more or less linear to the number of language *pairs*. As the number of language pairs for N languages is `N*(N-1)/2` (e.g. for 10, 20 or 30 languages, there are 45, 190 and 435 language pairs, respectively), the number of available language pairs for cross-lingual applications is always rather restricted. According to the web sites of some of the biggest MT providers, Language Weaver[8], Systran[9] and Google[10], the companies offer 28 language pairs involving 21 languages, 18 language pairs involving 13 languages and 29 language pairs involving 14 languages, respectively. All companies offer mainly language pairs involving English.[11] Admittedly, MT is particularly resource-intensive, but similar language pair restrictions apply to other application types. There are various news aggregation systems, but besides our own, NewsTin[12] is the only one that offers any kind of cross-lingual functionality: for 10 languages, NewsTin allows users to search for multilingual articles about a chosen entity (e.g. a person name) or for a given subject category. In comparison, the NewsExplorer application presented in this chapter is able to link news clusters and extracted information in all language pair combinations for 19 languages[13] and is not restricted by existing categories.

Current approaches to cross-lingual name variant matching are limited to individual language pairs (English/Arabic [11], English/Chinese [12], English/Korean [13] or English to Japanese [14], [15]). The same applies to cross-lingual topic detection and tracking systems, which require CLDS calculation. These systems typically either use MT (e.g. [16]) or bilingual dictionaries (e.g. [17] and [18]) to link related texts across languages, and this again limits the number of language pairs worked on. Statistical methods that establish cross-lingual word associations by feeding the vector space of Machine Learning systems with small pieces of text and their translation (parallel texts), such as bilingual Latent Semantic Indexing (LSI, [19]) and Kernel Canonical Correlation Analysis (KCCA, [20]), have to date only been successful with individual language pairs (involving thus a maximum of two languages). Whatever the application,

[8] See http://www.languageweaver.com/page.asp?intNodeID=930&intPageID=960

[9] See http://www.systran.co.uk/translation/translation-products/desktop/systran-premium-translator

[10] See http://www.google.co.uk/language_tools

[11] In the field of MT, effort can be saved by substituting the *direct approach* (i.e. translating words or phrases directly from one language to the other) by a *transfer-based approach*. In the latter, each source language is transformed into an abstract representation, which does not depend on the target language. The only language pair-specific component thus is the transfer module. The *interlingual approach* to MT goes even further as the aim is to transform the source language text into a language-independent representation and to generate the target language equivalence out of this interlingual representation. The trend of the last few years has been to use Machine Learning to produce direct approach systems.

[12] See http://www.newstin.com/ (last visited 1.2.2008).

[13] For purely computational reasons, NewsExplorer currently offers 'only' the 93 language pairs involving the six pivot languages Dutch, English, French, German, Italian and Spanish, out of the 171 possible combinations for 19 languages. The remaining language pairs could be covered by adding an extra machine.

existing systems – be they commercial or academic research systems – are rather strictly limited in the number of languages covered.

There is currently an effort to tackle more languages and more language pairs, for instance in the EU-funded projects *EuroMatrix*[14] for MT and SMART[15] (*Statistical Multilingual Analysis for Retrieval and Translation*) for MT and CLDS. The European CLEF[16] project (*Cross-Lingual Evaluation Forum*) aims at pushing multilinguality for information retrieval and other applications. The CONLL'2003 shared task aimed at achieving language-independent named entity recognition.[17] The projects Multext[18], Multext-East[19] and Global-WordNet[20] produced – or are producing – multilingual linguistic resources. Furthermore, the highly multilingual parallel text corpora JRC-Acquis ([21]) and the DGT-Translation Memory[21] have recently become available, showing the increased interest in accessing information hidden in foreign language text. All these resources will certainly help push multilinguality, but additionally to the resources, approaches need to be found that allow producing tools in more languages quickly and to cover more language pairs. The approach presented in this chapter aims at exactly that: offer a possible solution to a large-scale problem. The following seven sections will describe how the approach presented in Section 1.3 has been applied to a number of different text analysis applications. The first six of the described solutions have been implemented as part of the NewsExplorer system and the results can be seen on its public news portal.

3. Recognition and disambiguation of references to geographical locations

The text mining task of *geo-tagging* (also referred to as *geo-coding* or as *grounding of geographical references*) consists of recognising references to geographical locations in free text and to identify unambiguously their (ranges of) latitude and longitude. Geo-tagging goes beyond the more commonly pursued task of *geo-parsing*, which consists of recognising the words or phrases without attempting to put a dot on a map. The latter can theoretically be done simply by looking at linguistic patterns, even if there is evidence that geo-parsing is difficult without a gazetteer ([22]). For instance, when we find the piece of text " … is located near XYZ", we can infer that XYZ is a place name of some sort. Geo-tagging, on the other hand, is not possible by looking at the text only. Additionally, a gazetteer, i.e. a list of existing places and their latitude-longitude and probably more information, is needed. A number of gazetteers are freely or commercially available, such as *GeoNames*[22] *and* the *Global Discovery*[23] database. These gazetteers usually contain place names in one or more languages, latitude and longitude information, size categories for each place (distinguishing capitals from major or minor cities, villages, etc.), as well as hierarchical information indicating that a town belongs to a county, which is part of a region, which is itself part of a country, etc. A third gaz-

[14] See http://www.euromatrix.net/ (last visited on 1.2.2008)
[15] See http://www.smart-project.eu/ (last visited on 1.2.2008)
[16] See http://clef.isti.cnr.it/ (last visited on 1.2.2008)
[17] See http://www.cnts.ua.ac.be/conll2003/ner/ (last visited on 5.2.2008).
[18] See http://aune.lpl.univ-aix.fr/projects/multext/ (last visited on 1.2.2008)
[19] See http://nl.ijs.si/ME/ (last visited on 1.2.2008)
[20] See http://www.globalwordnet.org/ (last visited on 1.2.2008)
[21] See http://langtech.jrc.it/DGT-TM.html (last visited on 1.2.2008)
[22] See http://www.geonames.org/ (last visited 30.01.2008)
[23] See http://www.europa-tech.com/ (last visited 1.2.2008)

etteer, the KNAB[24] database, is particularly useful as it contains a large number of exonyms (foreign language equivalents like *Venice*, *Venise* and *Venedig* for the Italian city of *Venezia*), as well as historical variants (e.g. *Constantinople* for the Turkish city of *Istanbul*).

Geo-tagging thus consists of finding gazetteer entries (and probably other expressions) in text. This is basically a lookup task, but there are a number of challenges that make this task much harder than it seems at first sight. We will present these briefly in the next section and will then present a language-independent rule set to deal with these issues. For a definition of the tasks and an overview of existing commercial and academic approaches, see the recent Ph.D. thesis by Leidner ([23]). For a more detailed description of the challenges, the proposed knowledge-poor solution and issues concerning the compilation of a multilingual gazetteer, see [24].

3.1. Challenges for Geo-tagging

When using a gazetteer, the first step in recognising place names in text is a look-up task. In all European languages, it is possible to compare only uppercase words in the text with the entries of the gazetteer. In languages not distinguishing case, such as Arabic or Chinese, every single word must be compared to the gazetteer. However, even for European languages, geo-tagging is more than a simple lookup task, as a number of ambiguities need to be solved and language-specific issues need to be tackled. The challenges are the following:

(a) Homography between places and persons: For instance, there are both places and persons with the names of *George* (South Africa plus several other places worldwide with this name), *Washington* (capital of the USA and almost 50 other locations), *Paris* (French capital and almost 50 other locations) and *Hilton* (United Kingdom and many more places with this name).

(b) Homography between different places: An example is *Alexandria*, as there are 24 different cities with this name in ten different countries: Greece, Romania, South Africa, USA, etc.

(c) Homography between places and common words: For example, the English adjective *Nice* is also a city in France, the English verb *Split* is homographic with a major city in Croatia, etc. We have found a large number of places homographic with common words for all languages we worked with.

(d) The same place has different names: This is not only true across languages (e.g. the Italian city of *Milano* has the 'translations' – or exonyms – *Milan, Milán, Mailand, Μιλάνο, Милан, Милано,* ميلانو etc.). Even within the same country – and sometimes within the same language – places can have several names. Examples are *Bombay/Mumbai* and *Bruxelles/Brussell.*

(e) Place names are declined or morphologically altered by other types of inflection. This has the consequence that the canonical form found in the gazetteer will frequently not coincide with the declension found in the text. The US-American city of *New York* can for instance be referred to as *New Yorkului* in Romanian or as *New Yorgile* in Estonian.

[24] See http://www.eki.ee/knab/knab.htm (last visited 1.2.2008)

3.2. Language-independent rules for Geo-tagging

Challenges (d) and (e) necessarily involve the usage of language-specific resources: If name variants like *Mailand*, *Milàn* or *Милан* are not in the gazetteer (challenge (d)), they cannot be looked up. The only solution to this problem is to find ways of populating an existing gazetteer with multilingual place name variants in the least time-consuming way. At the JRC, we have merged various gazetteers and additionally exploit the online encyclopaedia Wikipedia for this purpose. Wikipedia can even be used to find inflected forms of city names (challenge (e)). For instance, for the same city, the following inflection forms were found on the Finnish Wikipedia page: *Milanon, Milanossa, Milanosta, Milanolainen, Milanoon, Milanolaiset, Milanoa*. The temptation is big to simply use generous wild cards such as *Milan**, but due to the hundreds of thousands, or even millions of entries in a gazetteer and the likely similarity with other words of the language, this will lead to many wrongly recognised names, lowering Precision. This specific wild card pattern would, for instance, wrongly recognise the Italian city of *Milano Marittima* and the Polish city *Milanówek*. The issue of inflection and the challenge to recognise other name variants will be discussed in Section 5, together with the variations of person names.

While challenges (d) and (e) cannot be overcome without language-specific resources and procedures, there are generic solutions for challenges (a), (b) and (c) that require no – or extremely little – language-specific effort. The proposed heuristics are the following:

(a) For known names, prefer person name readings over location name readings: The idea is to ignore all potential locations that are homographic with a name part of a person talked about in the same document. For instance, if *Javier Solana* has been mentioned in the text, we should assume that any further reference to either *Javier* or *Solana* refers to the person and not to the respective locations in Spain and the Philippines. The reason for this rule is that person name recognition is more reliable than geo-tagging.

(b) Make use of information about a place's importance or size: Many gazetteers like GeoNames or Global Discovery use size classes to indicate whether a place is a capital (size class 1), a major city, a town, etc. or a village (size class 6). If no further information is available from the context, we can assume that the text refers to the larger location. For instance, the Romanian city of *Roma* is of size class 4, while the Italian city with the same name is a capital (size class 1). The Italian capital should thus be chosen over the Romanian town.

(c) Make use of the country context: the idea is that – if we already know that a text talks about a certain country – then it is likely that a homographic place name should be resolved in favour of the place in that country. For instance, if we know that the text talks about Romania because either the news source is from Romania or because another *non-ambiguous* reference is made to the country or any of its cities, the likelihood that we talk about the Romanian town *Roma* is much bigger.

(d) Prefer locations that are physically closer to other, non-ambiguous locations mentioned in the text: In the case of ambiguity between two homographic places of the same size class, it is likely that the author meant to refer to the one nearby. For instance, there are two cities called *Brest*, one in France and one in Belarus. If both *Brest* and *Warsaw* are mentioned in a text, the latter reading will be preferred because it is at a distance of 200 km from Warsaw, while the French port is 2,000 km away.

(e) Ignore places that are too difficult to disambiguate: Locations that are homographic with common words of a language frequently lead to wrong hits. Such locations should be put on a language-specific geo-stop word list and ignored if found in a text. If an author really intends to refer to the places called *And* (Iran) or *Had* (India, and others), these references will be missed, but many errors will be avoided. Such geo-stop word lists are language-dependent because the same words are likely not to be ambiguous in another language. For instance, *And* and *Had* are not ambiguous in German as there are no such words. Geo-stop word lists can be produced semi-automatically by comparing the gazetteer entries with a list of the most frequent words of a language and by hand-picking those words that were found. In order to reduce the work load, this list can be narrowed down to those words that appear more often in lowercase than in uppercase. The uppercase word *This* (potentially referring to a location in France), for instance, will be found in such a word frequency list, but the lowercase variant *this* will be much more frequent, meaning that *This* is a good candidate for the geo-stop word list. It is also advisable to compare the lexicon of a language with an international list of frequent first names.

These heuristics were derived from multilingual test data and were found to produce good results in a majority of cases ([24]). The first four are completely independent of the text language as they refer to external parameters such as geographical location and location size. The fifth heuristic is language-dependent, but a good geo-stop word list for any given language can be produced within a few hours.

3.3. Combination of the rules

The rules mentioned in the previous section may contradict each other, so they have to be combined into a single rule that regulates their relative preference. When geo-tagging a new text, the binary rules will first be applied, i.e. geo-stop words will be ignored and potential locations that are homographic with part of a person name found in the text will not be considered. In a second instance, Formula (1) will be applied. Formula (2) explains the calculation of the parameter *kilometric weight*.

For computing the score, the currents settings are: (1)

$$\begin{aligned} &Score = classScore\,[80,30,20,10,5] \\ &+ 100 \text{ (if country in context)} \\ &+ 20 \cdot kilometricWeight() \end{aligned}$$

where: *classScore* is a given score depending on the importance of the place (this is coded by the *class* attribute: 80 for country name, capital or big city, 30 for province level city, 20 for small cities, 10 for villages, 5 for settlements); *kilometricWeight*, which has a value between 0 and 1, is the minimum distance between the place and all non-ambiguous places. This distance *d* is weighted using the arc-cotangent formula, as defined by Bronstein ([25]), with an inflexion point set to 300 kilometres[25], as shown in Equation 2.

[25] Empirical experiments showed that distances of less than 200 km are very significant, and distances more than 500 km do not make a big difference. Therefore, we have chosen the inflexion point between those two values: 300.

$$\text{kilometric Weight}(d) = \frac{1}{arcCot(-\frac{300}{100})} arcCot(\frac{d-300}{100}) \tag{2}$$

The formulae do not make reference to any language-specific features so that they can be applied to any new language without further consideration.

4. Recognition of person and organisation names

The names of *known* persons and organisations can be identified in new documents through a lookup procedure, just like the geographical place names from a gazetteer. For morphological and other variants, the same measures can be taken as for geographical names. These measures will be discussed in Section 5. However, as exhaustive lists of person and organisation names do not exist, *new names* need to be identified in one of two different ways: (1) When using dictionaries of a language, one could assume that any unknown word found in text is a proper name, but using dictionaries alone would be dangerous because any typo would then be identified as a name, as well. For languages that distinguish case and that begin proper names with an uppercase letter, the number of name candidates can of course be limited. (2) Additionally, or instead of using dictionaries, local patterns can be used, which can be either hand-written or acquired automatically with Machine Learning methods. Such local patterns are words, multi-word expressions or regular expressions that occur near to the names and that indicate that some (uppercase) words are likely to be a name. Such patterns can be titles (e.g. *Minister*), words indicating nationality (e.g. *German*), age (e.g. *32-year old*), occupation (e.g. *playboy*), a significant verbal phrase (e.g. *has declared*), and more. The words and expressions of different types can be referred to generically as *trigger words*, as their presence triggers the system to identify names. It goes without saying that these pattern recognition resources are necessarily language-specific. The challenge is thus to (a) keep the effort to produce these patterns to a minimum and (b) to formulate them in a generic way, which makes it easy to produce the patterns for a new language.

4.1. Patterns for name guessing

In JRC's named entity recognition tools, the generic patterns are basically language-independent, but they make use of language-specific resource files that contain the language-specific information such as lists of titles, nationality adjectives, etc. For full details on the used methods, see [3] and [26]. Here, we will focus on the aspect of language independence and on how to create language-specific resources quickly and with little effort. One intrinsic feature of JRC's tools is that they will only recognise names with at least two name parts. The reason for this is that the aim in NewsExplorer and other JRC tools is not only to recognise that *Angela* or *Bush* are names, but to identify the exact referent and its name variants so that users can search for all news items mentioning this person.

The following are the basic rules to identify person names in languages that write names with uppercase. For other languages, see Section 10:

(a) Any group of at least two uppercase words that is found to the left or to the right of one or more trigger words will be identified as a name candidate. Trigger words can also be regular expressions such as *[0-9]+-?year-old* to capture *43-year-old Alexander Litvinenko*.

(b) The pattern allows the presence of a number of name infixes which can also be written in lower case, such as *von*, *van der*, *bin*, *al*, *de la*, etc. to also capture names such as *Mark van der Horst, Oscar de la Renta, Khaled bin Ahmed al-Khalifa, etc.*

(c) Furthermore, a number of other frequent words are allowed between trigger words and the name. These can be determiners (e.g. *the*, *a*), adjectives and other modifiers (*former, wandering*), or compound expressions (*most gifted*), allowing name recognition in sentences like "… *Juliette Binoche, the most gifted French actress*".

(d) Patterns should also allow for the inclusion of a slot for other names (e.g. *United Nations*), in order to capture expressions such as *Envoy to the United Nations*.

(e) Names are also recognised if one of the potential name parts is a known first name. Large lists of first names from different languages and countries are thus a useful resource, that can be used for the name recognition in all languages. In the example *Angela UnknownWord*, the second element would thus be identified as the second name part. First names are thus different from the other trigger words mentioned earlier because they are part of the name, while titles and country adjectives are not.

(f) Organisation names are different in that they are often longer and they can contain several lowercase words that are normal words of the language, as in Federal *Ministry of the Interior*, etc. In order to capture such names, the patterns must allow various sequences of typical organisation trigger words (e.g. *Bank*, *Organi[sz]ation*, *Ministry*, *Committee*, etc.), lowercase filler words (e.g. *of the*) and other content words (e.g. *Interior, Finance, Olympic*, etc.).

It is useful to also allow long sequences of trigger words to capture expressions like *former Lebanese Minister of Agriculture*. While the combination *Minister of Agriculture* may be enough to recognise the proper name, storing trigger words and their combinations has the advantage that they provide useful information on a person. In NewsExplorer, the historically collected trigger words are displayed together with each person.

4.2. Bootstrapping the acquisition of language-specific pattern ingredients

Besides the list of common first names, which can be used for the recognition of new names in any language, the various trigger words mentioned in the previous section are clearly different from one language to the other. These lists can be rather long. Our English list, for instance, consists of about 3400 trigger words. In order to compile such a list for a new language (e.g. Romanian), it is convenient to search a large corpus of that language for known names and to produce a frequency list of left and right-hand-side contexts of various sizes (e.g. between one and five words). The most frequent trigger words can then be manually selected. Bootstrapping will make the process more efficient: instead of going manually through the typically very long name context lists, it is better to use the most frequent trigger words found and to search the collection again for new names in order to collect more name contexts, and so on. Wikipedia or similar sources often contain lists of common first names and titles, so that such lists can also be used as a starting point.[26] The effort to produce working lists of recognition

[26] See, for instance, the web site www.behindthename.com for first names, and the following page for lists of professions: http://en.wikipedia.org/wiki/Category:Occupations (available in various languages).

patterns for a new language is between half a day and 5 person days. Note that the manual selection of trigger expressions is indispensable and that an intelligent reformulation of expressions can improve the rules massively. For instance, for Romanian occupations like *ministrul de interne*, experts can expand the rule immediately to other occupations: *ministrul de externe, ministrul de finanțe, ministrul justitiei, ministrul transporturilor* and more (Minister for external affaires, finance, justice and transport, respectively). They can even write a more complex pattern to allow the recognition of combinations like *ministrul transporturilor, constructiilor si turismului* (Minister of transport, construction and tourism) or *Ministrul delegat pentru comerț* (Vice-minister for commerce). In the case of Romanian, the first list has been validated to a certain extent and the expert produced 231 trigger words in about 3 hours of time. After this new list has been compiled, we launched the candidate extractor again and another validation was done by the expert. We now have 467 trigger words and Romanian is used fully in NewsExplorer, where it recognises an average of one hundred new names every day.

Another bootstrapping method would be to use MT or bilingual dictionaries in a triangulation approach, i.e. translating from two or more different languages into the new target language and to use only the overlapping results.

5. A light-weight way of dealing with inflection and other regular variations

We have seen that – for the tasks of geo-tagging, recognition of known person names and even for the guessing of new names in text – it is necessary to compare word forms in the text with lists of known words in lookup lists. Even though the task looks simple, looking up known entities in a new language is not always so straightforward because the word forms found in the text often differ from the word forms in the lookup tables.

5.1. Reasons for the existence of name variants

The main reasons for these differences between the dictionary form in the lookup tables and the word forms found in real text – together with the adopted solutions – are the following:

(a) Hyphen/space alternations: For hyphenated names such as *Jean-Marie*, *Nawaf al-Ahmad al-Jaber al-Sabah* or the place name *Saint-Jean*, we automatically generate patterns to capture both the hyphenated and the non-hyphenated forms (e.g. `Jean[\-\ ]Marie`).

(b) Diacritic variations: Words that carry diacritics are often found without the diacritic. For example, *François Chérèque* is often written *Francois Chereque*, Schröder as Schroder, Lech Wałęsa as Lech Walesa, Raphaël Ibañez as Raphael Ibanez, etc.. For each name with diacritics, we therefore generate a pattern that allows both alternatives (e.g. `Fran(ç|c)ois Ch(é|e)r(è|e)que`).

(c) Further common variations: Especially for place names, there are a number of very common variations, including the abbreviation of name parts such as *Saint* to *St* (with or without the final dot) or the use of a slash instead of common name parts. For instance, *Nogent-sur-Seine* and *Frankfurt am Main* can be found as *Nogent/Seine* and *Frankfurt/Main* (also: *Frankfurt a. Main*), etc. For all such names, we thus pre-generate the various common variants.

(d) Name inversion: While news texts in most languages mention the given name before the surname, the opposite order can also be found in some languages. In Hungarian, for example, local names are usually written with the last name first, while foreign names have the opposite order. The lookup procedure must consider this variation, as well.

(e) Morphological declensions: In some languages (especially the Balto-Slavonic and Finno-Ugric languages), person names can be declined. In Polish, for example we can find the inflected form *Nicolasowi Sarkozy'emu* (or – less frequent – *Nicolasowi Sarkoziemu*) referring to the French president *Nicolas Sarkozy*. Similarly *Tony'ego Blaira* or *Toniego Blaira* are found for the former British prime minister. For these languages, we pre-generate morphological variants for all known names according to rules that will be discussed below. It must be high-lighted that – in some languages – variations can also affect the beginning of the name. For instance, for the Irish place name *Gaillimh* (Irish version of *Galway*), we can find *nGaillimh* (in Galway). For some languages, the number of different inflections can be rather high.

(f) Typos: Even in the printed press, typos are relatively frequent. This is especially the case for difficult names such as *Condoleezza Rice*. For this person's given name, we found the typos *Condoleza*, *Condaleezza*, *Condollezza* and *Condeleeza*, each more than once.

(g) Simplification: In order to avoid repetition, names such as *Condoleezza Rice* and *George W. Bush* are frequently simplified to *Ms. Rice* and *President Bush*.

(h) Transliteration: Names are normally transliterated into the target language writing system. In the case of NewsExplorer, we are mainly interested in *Romanisation*, i.e. in using the Latin script as a target language. Depending on the target language, transliteration rules often differ so that two different Romanised versions of the same name can co-exist. For example, the Russian name Владимир Устинов is typically transliterated to *Wladimir Ustinow* in German, to *Vladimir Ustinov* in English, to *Vladímir Ustinov* in Spanish, to *Vladimir Oestinov* in Dutch and to *Vladimir Oustinov* in French. Conversely the French city *Strasbourg* is sometimes written in Russian Страсбург (/stʀasbuʀg/) sometimes Стразбург (/stʀazbuʀg/), in Ukrainian Страсбур (/stʀasbuʀ/), in Serbian Стразбур (/stʀazbuʀ/), without the final 'g' mimicking the original French pronunciation.

(i) Vowel variations, especially from and into Arabic: In Arabic and some other languages, short vowels are not always written. The string محمد (Mohammed) contains only the four consonants Mhmd, which is the reason why so many variants exist for this name: *Mohammed*, *Mohamed*, *Mahmoud*, *Muhamad*, and more.

5.2. Generating variants for known names

The variation types (a) to (d) are rather generic so that it is not difficult to pre-generate the most common name variants, as shown in the previous section. Morphological variations such as those shown in (e) are much harder to predict and they differ from one language to the next. This section describes how this variation type can be dealt with. For the variation types (f) to (i), see Section 6 below.

Profound linguistic skills and native speaker competence will help to produce good suffix addition and suffix replacement rules. [27], for instance, have produced extensive inflection rules for Serbian. However, in order to achieve high multilinguality, it is important to find an efficient and quick method to generate at least the most common variants. We found that, even without native speaker competence, it is possible to identify a number of frequent inflection rules purely by observing common variations for known names. These can be found by searching text collections using several generous regular expressions such as `Nicol.*Sarko[[:alpha:]]+` (for the French president) allowing to capture name variants and by then looking for regularities. In Romanian, for example, one will discover that known names are frequently accompanied by the suffixes `-ul` and `-ului` (suffix addition), and that the endings are `-l` and `-lui` if the name already ends in `-u`. If the name ends with `-a` we frequently find the `-ie` ending (suffix replacement). By collecting a number of such observations, we can produce suffix addition and replacement rules to pre-generate – for all names in the lookup tables – the most frequent variants. For the names *Paris, Bacău* and *Venezia*, for example, we can then generate the variants *Parisul*, *Parisului*, *Bacăul*, *Bacăului*, *Venezia* and *Veneziei*. Slavic languages are more complex, but the same principle holds. For the Slavic language Slovene, the regular expression substitution rule for person names is:

```
s/[aeo]?/(e|a|o|u|om|em|m|ju|jem|ja)?/
```

meaning that – for every name ending in –a, –e or –o – we pre-generate ten different variants, ending in `-e, -a, -o, -u, -om, -em, -m, -ju, -jem` and `-ja`, respectively. For every frequent known name in our name database such as the previous Lebanese political leader *Pierre Gemayel*, for instance, we will thus generate the pattern:

```
Pierr(e|a|o|u|om|em|m|ju|jem|ja)? Gemayel(e|a|o|u|om|em|m|ju|jem|ja)?
```

That way, *Pierrom Gemayelom* and any other of the other possible combinations will be recognised in text. Note that over-generation, i.e. producing name variants that do not exist in the language, is not normally a problem because they will simply not be found. However, especially short names can lead to over-generous patterns and new patterns should always be tested on large document collections before being applied in a real-world scenario.

To give an indication of the effort required: it takes us usually between 1 hour and 2 days to produce basic working lists of inflection patterns for a new language.

An alternative to *generating* morphological variants for highly inflective languages would be to map different name variants found in text using a mixture of string distance metrics and automatically acquired suffix-based lemmatisation patterns, as it was proposed by [28] for Polish.

5.3. Transcription of names written in different writing systems

Names from a language using a different writing system are usually transliterated. Transliteration rules are simple, normally hand-written rules mapping characters or sequences of characters from one writing system to their counterparts in another writing system ([29]). Some Greek examples are:

- ψ => ps
- λ => l
- μπ => b

Frequently, more than one transliteration system exists for the same language pair (the Arabic first name سعي is mainly transliterated in French as Saïd but sometimes also as *Said*, *Sayyed* and *Saeed*), which explains why different target language versions may exist for the same name and the same source-target language pair. As pointed out in bullet (h) in Section 5.1, transliteration rules usually differ depending on the target language. It is less common knowledge that transliteration also exists for languages using the same writing system. For example, the name *Bush* will be found in Latvian language as *Bušs*. We deal with intra-script transliteration in the normalisation step described in Section 6. At the JRC, we use transliteration rules for the following scripts:

- Cyrillic (used for Russian, Bulgarian and Ukrainian):
  Симеон Маринов → Simeon Marinov;
- Greek: Κώστας Καραμανλής → Kostas Karamanlis;
- Arabic (used for Arabic, Farsi and Urdu; some additional transliteration rules were added for Farsi and Urdu):
  جلال طلباني → jlal tlbani ("Jalal Talabani");
- Devanagari (used for Hindi and Nepalese):
  सोनिया गांधी → soniya gandhi.

Transliteration sometimes produces name forms that are rather different from the usual spelling. For frequent names, it is thus appropriate to hard-code the transliteration of the full name. Here are some examples of source language strings, their letter-by-letter transliteration and the aimed for target language form:

- Russian Джордж → [Djordj] → *George;*
- Russian Джеймс → [Djaims] → *James;*
- Hindi डब्ल्यु → [dableyu] → W (as in *George W. Bush*);

All other spelling differences of transliterated versus non-transliterated names are dealt with in the normalisation step, described in the next section. Adding transliteration tables for new languages using letters (alphabetical scripts) or syllables is not difficult.[27] Adding rules for Hindi took us two hours. Dealing with ideographic languages such as Chinese is harder and needs different procedures.

6. Rules to identify name variants

In Section 5.1, we have seen a number of reasons why variants exist for the same name. After having applied named entity recognition in up to nineteen languages over a period of about five years, we have found up to 170 variants for the same name.[28] Identifying these variants as referring to the same person has many advantages, including improved search and retrieval, as well as more accurate results for tasks where person co-references are required such as social network generation based on quotations ([30]) or on co-occurrence ([1]; see also Feldman's chapter on *Link Analysis in Networks of Entities*, in this volume). There is thus a clear need for methods to identify whether similar, but different names found in text are variants belonging to the same person or not. There are a number of known approaches to identify name equivalences for specific language *pairs*. In this section, we present an alternative approach, which is language and language pair-independent. It consists of normalising names into an abstract *consonant signature*. This consonant signature can then be used as the basis for comparing all (normalised) names found. All name pairs that have a similarity above a certain threshold will be marked as referring to the same person. The threshold was set so that only good name pairs would be merged for a given test set. The idea behind this is

[27] Various transliteration tables can be found at the *Institute of the Estonian Language* and the *American National Geospatial Agency* (http://transliteration.eki.ee/; http://earth-info.nga.mil/gns/html/romanization.html).

[28] The terrorist Abu Musab al-Zarqawi: see http://press.jrc.it/NewsExplorer/entities/en/285.html.

that having two entries for the same person is less harmful than that of merging two different persons. For details on the name normalisation and name merging processes, see [26].

For every name found in the analysis of news articles in 19 languages carried out daily by the NewsExplorer application, we first check whether the name already has an entry in the NewsExplorer name database. Both main names (aliases) and known name variants are considered. All unknown names will be normalised (see Section 6.1) and compared to the consonant signature of any of the known names. If any of the consonant signatures coincide, name similarity measures will be applied (see Section 6.2). If successful, the new name variant will be added to the existing name. Otherwise, the name will be added as a new name into the database. Names found only ever once are otherwise ignored in order to avoid typos entering the database. If a name is found at least five times, this name gets the status of a frequent name so that it will be found by lookup in any future news articles (see Section 5).

6.1. Language-independent name normalisation rules

The idea behind name normalisation is to create a language-independent representation of the name. In principle, a representation of the (approximate) pronunciation of the name would be a good normalised form, as suggested by the *Soundex* algorithm for English ([31]). However, in order to infer the pronunciation of a name, it is necessary to know the original language of the name. In news articles, this information is not normally known as known persons from around the world are being talked about in the newspapers of any other country. To give an example: the name of the previous French president *Chirac* (pronounced in French as /ʃiʀak/) would be pronounced as /kiɾɑk/ if it were an Italian name, as /çiʀɑk/ in German, etc., while the name *Chiamparino* (Mayor of the city of Turin) should be pronounced as /kiɑmpɑɾino/. Automatic identification of the origin of a name ([32]) produces moderate results.

In order to overcome this problem, empirical observations on name spelling differences for the same name in many different languages have been used to produce normalisation rules that will be applied to all names, independently of their origin. The following are some examples:

- Name-initial 'Wl–' and the name-final '–ow' for Russian names will get replaced by 'Vl–' and '–ov'. This is to accommodate the typical German transliteration for names like *Vladimir Ustinov* as *Wladimir Ustinow.*
- The Slovene strings 'š', the Turkish 'ş', the German 'sch', the French 'ch' will get replaced by 'sh' in order to neutralize frequent spelling variants such as *Bašar al Asad*, *Baschar al Assad*, *Bachar al Assad* and *Başar Esad.*
- The French letter combination 'ou' will get replaced by 'u' to capture the regular transliteration differences for the sound /u/ in names like *Ustinov*, which is spelled in French as *Oustinov*.
- The letter 'x' will get replaced by 'ks', etc.

These normalisation rules are exclusively driven by practical requirements and have no claim to represent any underlying linguistic concept. They are written as a set of regular expression substitutions. For example, the normalisation rule for the Slavic ending –*ski* is written by the substitution rule below and will normalise the following names to a common suffix: *Stravinski*, *Stravinsky* (Finnish), *Stravinskij* (Slovak), *Sztravinszkij* (Hungarian), *Stravinskíj* (Icelandic):

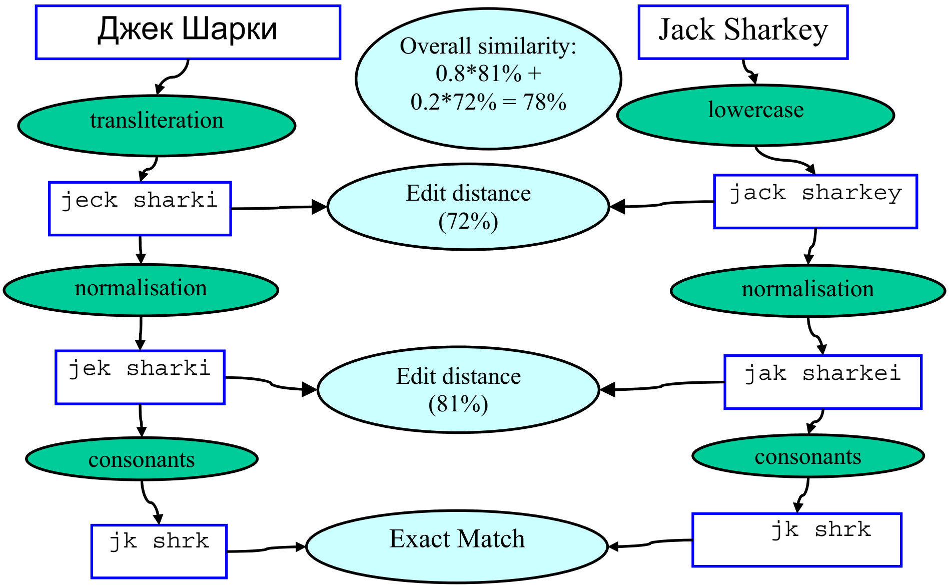


Figure 1. The Figure shows the transliteration and normalisation steps, as well as the similarity measurement applied to a name pair. The two names will not be associated to the same person because the overall similarity is 0.78 and thus lower than the requested threshold of 0.94.

`s/sz?k[yií]j?/ski/` (At the end of a word)

The final normalisation step consists of removing all vowels. Vowels frequently get distorted during transliteration and their removal is compulsory for languages using the Arabic script as short vowels are not normally written in Arabic. An original Russian name such as Джек Шарки will thus go through the stages transliteration (*jeck sharki*), normalisation (*jek sharki*) and vowel removal (*jk shrk*).

6.2. Similarity measure used to compare names with the same normalised form

All names having the same normalised form are considered name variant *candidates*. For each candidate pair, the edit distance similarity measure is applied twice, one time each on two different representations of the name: once between the normalised forms with vowels and once between the lowercased transliterated forms (see Figure 1). The two similarities have relative weights of 0.8 and 0.2. By applying the similarity measure only to name pairs with an exact signature match, we miss some good candidates, but the pre-filtering saves a lot of precious computing time. For further details, see [26].

Both the name normalisation and the similarity calculation steps are applied across the board to all names found in the currently 19 NewsExplorer languages and do not depend on the input language. There is thus no need for language pair-specific training or mapping rules. However, when adding a language with a new script, a new set of transliteration rules needs to be added and – depending on the language – it may be useful to add additional normalisation patterns (which will then be applicable to all languages).

7. Quotation recognition

Direct speech quotations can typically be identified in text due to the presence of quotation markers (e.g. single or double, lower or upper quotes (“„ ‘), single or double angled brackets (<<, >>, “, ”, etc.)), a reporting verb (*says*, *quotes*, *criticises*, *objects*, etc.) and a reference to the person who makes the quote. In [30], we present a number of language-independent patterns that allow to recognise quotations in running text. The most frequent quotation pattern is covered by the following rule:

> *name* [, *up to 60 chars* ,] *reporting-verb* [:|that] *quote-mark* QUOTE *quote-mark*
> e.g. *John Smith, supporting AFG, said: "Here we are!".*

By keeping the language-independent quotation recognition patterns separate from the language-specific lists of reporting verbs, the quotation recognition software can very easily be extended to new languages. For language-specific issues, see Section 10.

8. Date recognition

Date recognition is a known named entity recognition task (e.g. MUC-7)[29], but recognition patterns for date formats other than numerical dates (such as *26.05.2009*) are usually developed for individual languages (e.g. [10]). Our approach, described in [33], completely separates the language-independent rules from language-specific parameter files, for numerical and alpha-numeric date formats. The tool detects dates in full form format (such as *26th of May of the year two thousand and nine*), but also partial dates such as *26 May* or *May 2009*. Rules make reference to slot fillers, for which the language-specific expressions can be found in parameter files. The following rule captures complete dates with the day mentioned before the month (simplified pseudo-code).

```
(0..31|CARDINAL|ORDINAL)(SEPARATOR|MONTH-INTRO)
(0..12|MONTH)(SEPARATOR|YEAR-INTRO)
(1800..2099|00..99|YEAR-IN-LETTERS)
```

CARDINAL stands for a cardinal number in letters (e.g. *twenty-one*, *vingt-et-un*), *ORDINAL* for an ordinal number in letters or numbers (*1st*, *third*, French: *1er*, *troisième*, etc.), *SEPARATOR* for a common character used to separate numbers (e.g. /), *MONTH-INTRO* for a common expression to introduce the month (twenty-sixth *of* May; Spanish: catorce *de* febrero), *MONTH* for a month name or an equivalent abbreviation (e.g. *July*, *Jan.*), *YEAR-INTRO* for a common expression to introduce the year (26th of May *in the year* 2009; Romanian: întîi martie *al anului* 2009), and *YEAR-IN-LETTERS* for the full or partial year in letters (*sixty-four*, *nineteen eighty*; French: *deux mille un*). Variants of these words (e.g. words with and without diacritics, with or without case endings) should also be listed. Language-specific parameter files for date recognition are simple and can be created within half a day’s work per language (including variants).

Procedures to resolve relative date expressions (e.g. *last May*), to disambiguate ambiguous date formats (e.g. the European vs. the US-American reading of a date like *10-05-2009*: 10 May vs. 5 October) and to convert dates from other calendars (e.g. Arabic *01/06/1430*, which is equivalent to *26/5/2009*) are obviously language-

[29] See http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/ne_task.html for the MUC-7 task specification (last visited 6 May 2008).

independent. Date expressions found can be normalised for storage and improved retrieval.

9. Cross-lingual document similarity calculation

Document similarity calculation across languages can be useful for news navigation (e.g. in NewsExplorer), for multilingual topic detection and tracking, for cross-lingual plagiarism detection, in query-by-example scenarios, and possibly more. We have shown in Section 2 that existing methods for cross-lingual document similarity (CLDS) calculation are bilingual in nature and are thus limited with respect to the number of language pairs. NewsExplorer can calculate CLDS for currently 19 languages (171 language pairs), although one of the ingredients of the formula (Eurovoc indexing) is not available for the six non-EU languages covered, so that the software is fully functional for language pairs involving *only* 13 languages.

The NewsExplorer approach (described in detail in [33]) is, in principle, almost language-independent. The main idea behind it is to find a language-independent representation for each of the texts written in different languages, i.e. to find a collection of anchors that serve to establish the link between texts in different languages. The CLDS in NewsExplorer is based on four ingredients (represented each as a vector). The first three of these are represented in a language-neutral manner:

1. A weighted list of subject domain classes using the Eurovoc thesaurus ([35]). The Eurovoc classes are represented by their numerical identifier.
2. A frequency list of countries to which each text makes a reference, either direct (country name) or indirect (city name, country adjective, citizen name, etc.; see Section 3). Countries are represented by their country ISO 3166 code.
3. A frequency list of persons made reference to in each of the texts. The persons are represented by their numerical identifier in the multilingual name database, which subsumes various name variants under the same identifier (see Section 4).
4. A weighted list of monolingual keywords. While most words of texts in different languages will not match, we found that there is frequently a number of cognates, numbers, acronyms, names or name parts that match between the languages.

Each document or group of documents (NewsExplorer works with clusters of related documents) is thus represented by four language-neutral vectors. For each pair of documents in different languages, the CLDS formula calculates the cosine similarities between each of the four vectors and combines them with the relative weight of 0.4, 0.3, 0.2 and 0.1 in order to come up with an overall CLDS. The higher the overall similarity is, the more similar the two (clusters of) documents are. In NewsExplorer, document pairs below the similarity threshold of 0.3 are ignored, i.e. the documents are treated as if they were not related. The similarity formula itself is language-independent, but language-specific tools obviously need to be developed to produce the language-neutral representation. A big advantage is, however, that no *language pair-specific* procedures or resources are needed. New languages can be plugged in without any additional effort.

10. Language-specific issues – the limitations of language-independence

In previous sections, we have already seen that some language-specific resources are needed in addition to the language-independent rules, including:

- Gazetteers of place names (for geo-tagging);
- Geo-stop word lists (for geo-tagging);
- Lists of trigger pattern components (for person and organisation name recognition);
- Lists of inflection patterns for all lookup procedures (regular expressions to add or substitute suffixes);
- Lists of days of the week, months, numbers in words, etc. (for date recognition).

Apart from the gazetteers, these resources can be produced relatively quickly by exploiting bootstrapping methods, as described earlier. Additionally, there are a number of issues where language-specific rules or procedures are required or will at least help improve the results. These are:

(a) Diacritic normalisation: In some Italian sources, diacritics are replaced by an apostrophe following the word (e.g. *Libertà* – freedom – can be found as *Liberta'*).

(b) Case information: Some languages (e.g. Arabic, Farsi, Hebrew, Chinese, Japanese, Hindi, etc.) do not distinguish upper and lower case so that every word in the text needs to be considered in a lookup procedure instead of considering only uppercase words. This also has the consequence that it is harder to guess where a person name ends. The opposite also holds for languages like German, which write nouns or other non-names in uppercase: name boundaries are less easily detected because uppercased nouns may follow a person name.

(c) Missing vowels: In Arabic, Hebrew and – to some extent – Hindi, short vowels are not normally written (the Arabic spelling of the name *Mohammed* consists of the four consonants *mhmd* only). The result of transliterating names written with the Arabic script is thus often different from their Latin counterpart.

(d) Tokenisation (no word separators): In Chinese, Thai and some other languages, words are not separated by spaces. Words could thus start at any character in a string of characters. This needs to be considered in the lookup procedures.

(e) Tokenisation (apostrophe): Different, language-specific tokenisation rules are required to deal with the following strings (which all contain proper names), because the apostrophe sometimes marks the word border and sometimes it is part of the name: *Le Floc'h*, *Tony Blair's*, *Jan Figel'*, *Stéphane M'Bia*, etc.

(f) Tokenisation (agglutinative languages): In languages such as Turkish, Hungarian and Finnish, various particles (e.g. prepositions or determiners) may be attached to the end of any word, including names. This increases the number of possible suffixes to be considered in lookup procedures enormously. It is possible that morphological analysis software will eventually be needed for such languages to achieve good results. At the same time, our analysis has shown that the number of proper noun endings – even for Finnish – is limited.

(g) Compounding: In languages like German, nouns can be combined with (mostly) other nouns to produce compound nouns. For instance, the words *Bundesverkehrsminister* consists of *Bund* (Federal), *Verkehr* (transport) and *Minister*. As this combination is productive, it is not possible to list all combinations. This is obviously a problem for producing trigger word lists for name recognition. Regular expressions with generous wild cards must be used word-initially (*.*minister*), which could lead to problems regarding the computational performance of the system. The Finnish word *Lontoolaishotelliin* (to a hotel in London) shows that compounding and agglutination can also occur combined.

(h) Affix types: while most Western-European languages almost exclusively use suffixes for name inflection, Irish also uses prefixes (*le hAngela Carter* "with Angela Carter"). We are not aware of any language using infixes for names.

(i) Order of first name and family name: As discussed in Section 5.1, given and last names of local names are inverted in Hungarian, but not in foreign names. This must be considered in the lookup procedure.

(j) Quotation markers (see Section 7): In most NewsExplorer languages, the information on the speaker and the reporting verb are found *outside* the quotation, but Italian and Swedish allow to move them to a place inside the quotation markers (see [30]) (e.g. “Sacco e Vanzetti – ha ricordato Nichi Vendola – erano due emigranti” / “Sacco and Vanzetti were two emigrants” reminded Nichi Vendola).

11. Conclusion

In this chapter, we proposed a number of guidelines that are useful when aiming at building multi-monolingual and cross-lingual applications for larger numbers of languages (Section 1.3). These guidelines are: (a) To the largest extent possible, use language-independent rules instead of language-specific ones. (b) Reduce the usage of language-specific and especially *language pair*-specific resources to a minimum. (c) Where they are necessary, keep them in language-specific parameter files to keep the applications modular so that new languages can be *plugged in*. (d) For those resources that cannot be avoided, use bottom-up bootstrapping methods to create the resources. (e) For *cross-lingual* applications, attempt to use a language-neutral document representation to reduce the effect of the near-exponential complexity increase when dealing with many languages.

To our knowledge, other system developers do not work according to these seemingly obvious and simple principles. Instead, the most common approach is to develop tools for one language and to then adapt these tools to further languages (See Section 2 on related work). We believe that tools for new languages will be developed quicker and more easily if the proposed guidelines are followed.

In order to show that putting these principles into practice is feasible for a range of different applications, we described such rule sets – and, where necessary, the simple acquisition procedures for the related resources – for seven applications: geo-tagging (Section 3); person and organisation name recognition (4); processing of name inflection (morphological variation) and other surface form variations (5); name variant mapping within the same language or across different languages and scripts (6); the recognition of quotations (7), date recognition (8), and cross-lingual document similarity calculation (9). These application examples are followed by a list of specific issues where the applicability of language-neutral rules ends and where language-specific adaptations are needed.

All the applications described in this chapter have been implemented to an operational level. The first six of them are used daily in the public large-scale news aggregation, analysis and exploration system NewsExplorer, which analyses 35,000 news articles per day in the 19 languages Arabic, Bulgarian, Danish, Dutch, English, Estonian, Farsi, French, German, Italian, Norwegian, Polish, Portuguese, Romanian, Russian, Slovene, Spanish, Swedish and Turkish.

12. Acknowledgment

We would like to thank the many people who – in the course of several years – have contributed with the knowledge of their languages to produce language-specific resources and to test the results. We would like to thank the whole team of the *Web Mining and Intelligence* group at the JRC – and especially the team leaders Clive Best and Erik van der Goot – for having provided the reliable and high-quality news data in a standardised and clean format. We are grateful to our colleagues Ken Blackler, Flavio Fuart and Martin Atkinson for making the results of our work accessible to the wider public using robust web portals, and to Jenya Belyaeva for her quality control.

13. References

[1] Pouliquen Bruno, Ralf Steinberger & Jenya Belyaeva (2007). Multilingual multi-document continuously updated social networks. Proceedings of the RANLP Workshop Multi-source Multilingual Information Extraction and Summarization (MMIES'2007). Borovets, Bulgaria.

[2] Tanev Hristo (2007). Unsupervised Learning of Social Networks from a Multiple-Source News Corpus. Proceedings of the RANLP Workshop *Multi-source Multilingual Information Extraction and Summarization* (MMIES'2007). Borovets, Bulgaria.

[3] Pouliquen Bruno & Ralf Steinberger (in print). Automatic Construction of a Multilingual Name Dictionary. In: Cyril Goutte, Nicola Cancedda, Marc Dymetman & George Foster (eds.): Learning Machine Translation. MIT Press.

[4] Larkey Leah, Fangfang Feng, Margaret Connell & Victor Lavrenko (2004). Language-specific Models in Multilingual Topic Tracking. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 402-409.

[5] Gamon Michael, Carmen Lozano, Jessie Pinkham & Tom Reutter (1997). Practical Experience with Grammar Sharing in Multilingual NLP. In Proceedings of ACL/EACL, Madrid, Spain.

[6] Rayner Manny & Pierrette Bouillon (1996). Adapting the Core Language Engine to French and Spanish. Proceedings of the International Conference NLP+IA, pp. 224-232, Mouncton, Canada.

[7] Pastra Katerina, Diana Maynard, Oana Hamza, Hamish Cunningham & Yorick Wilks (2002). How feasible is the reuse of grammars for Named Entity Recognition? Proceedings of LREC, Las Palmas, Spain.

[8] Carenini Michele, Angus Whyte, Lorenzo Bertorello & Massimo Vanocchi (2007). Improving Communication in E-democracy Using Natural Language Processing. In IEEE Intelligent Systems 22:1, pp 20-27.

[9] Maynard Diana, Valentin Tablan, Hamish Cunningham, Christian Ursu, Horacio Saggion, Kalina Bontcheva & Yorick Wilks (2002). Architectural Elements of Language Engineering Robustness. Journal of Natural Language Engineering 8:3, pp 257-274. Special Issue on Robust Methods in Analysis of Natural Language Data.

[10] Bering Christian, Witold Drożdżyński, Gregor Erbach, Lara Guasch, Peter Homola, Sabine Lehmann, Hong Li, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, Atsuko Shimada, Melanie Siegel Feiyu Xu & Dorothee Ziegler-Eisele (2003). Corpora and evaluation tools for multilingual named entity grammar development. Proceedings of the Multilingual Corpora Workshop at Corpus Linguistics, pp. 42-52, Lancaster, UK.

[11] Glover Stalls Bonnie & Kevin Knight (1998). Translating Names and Technical Terms in Arabic Text. In Proceedings of the ACL-CoLing Workshop 'Computational Approaches to Semitic Languages'.

[12] Lee C.-J., J. S. Chand, and J.-S.R. Jang (2006). Extraction of transliteration pairs from parallel corpora using a statistical transliteration model. Information Science 17, 6, pp. 67-90.

[13] Oh Jong-Hoon and Key-Sun Choi (2002). An English-Korean transliteration model using pronunciation and contextual rules. Proceedings of ACL.

[14] Knight Kevin & Jonathan Graehl (1998). Machine Transliteration. Computational Linguistics 24:4, pp. 599-612.

[15] Qu Yan & Gregory Grefenstette (2004). Finding ideographic representations of Japanese names written in Latin script via language identification and corpus validation. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain.

[16] Leek Tim, Hubert Jin, Sreenivasa Sista & Richard Schwartz (1999). *The BBN Crosslingual Topic Detection and Tracking System*. In 1999 TDT Evaluation System Summary Papers.

[17] Wactlar H.D. (1999). New Directions in Video Information Extraction and Summarization. In Proceedings of the 10th DELOS Workshop, Sanorini, Greece.

[18] Saralegi Urizar Xabier & Iñaki Alegria Loinaz (2007). Similitud entre documentos multilingües de carácter científico-técnico en un entorno Web. Proceedings of SEPLN, Seville, Spain.

[19] Landauer Thomas & Michael Littman (1991). A Statistical Method for Language-Independent Representation of the Topical Content of Text Segments. Proceedings of the 11th International Conference 'Expert Systems and Their Applications', vol. 8: 77-85. Avignon, France.

[20] Vinokourov Alexei, John Shawe-Taylor & Nello Cristianini (2002). Inferring a semantic representation of text via cross-language correlation analysis. Proceedings of Advances of Neural Information Processing Systems 15.

[21] Steinberger Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş & Dániel Varga (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. Proceedings of the 5th International Conference on Language Resources and Evaluation LREC, pp. 2142-2147. Genoa, Italy.

[22] Mikheev A., M. Moens & C. Gover (1999) Named Entity Recognition without Gazetteers. In Proceedings of EACL, Bergen, Norway.

[23] Leidner Jochen (2007). Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names. Ph.D. thesis, School of Informatics, University of Edinburgh, Edinburgh, Scotland, UK.

[24] Pouliquen Bruno, Marco Kimler, Ralf Steinberger, Camelia Ignat, Tamara Oellinger, Ken Blackler, Flavio Fuart, Wajdi Zaghouani, Anna Widiger, Ann-Charlotte Forslund, Clive Best (2006). Geocoding multilingual texts: Recognition, Disambiguation and Visualisation. Proceedings of the 5th International Conference on Language Resources and Evaluation LREC, pp. 53-58. Genoa, Italy.

[25] Bronstein I. N., K. A. Semendjajew, G. Musiol & H Muhlig (1999). Taschenbuch der Mathematik (4. ed.). Frankfurt am Main, Thun: Verlag Harri Deutsch.

[26] Steinberger Ralf & Bruno Pouliquen (2007). Cross-lingual Named Entity Recognition. In: Satoshi Sekine & Elisabete Ranchhod (eds.). Lingvisticæ Investigationes LI 30:1, pp. 135-162. Special Issue Named Entities: Recognition, Classification and Use.

[27] Vitas Duško, Cvetana Krstev & Denis Maurel (2007). A note on the Semantic and Morphological Properties of Proper Names in the Prolex Project. In: Satoshi Sekine & Elisabete Ranchhod (eds.). Lingvisticæ Investigationes LI 30:1, pp. 115-134. Special Issue Named Entities: Recognition, Classification and Use.

[28] Piskorski Jakub, Karol Wieloch, Mariusz Pikula & Marcin Sydow (2008). Towards Person Name Matching for Inflective Languages. In: Proceedings of the WWW'2008 workshop 'Natural Language Processing Challenges in the Information Explosion Era'. Beijing, China.

[29] Daniels Peter T. & William Bright (eds.) (1996). The World's Writing Systems. Oxford University Press, Oxford, UK:.

[30] Pouliquen Bruno, Ralf Steinberger & Clive Best (2007). Automatic Detection of Quotations in Multilingual News. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP. Borovets, Bulgaria.

[31] Hall P. and G. Dowling (1980). Approximate string matching. Computing Surveys, 12:4, pp. 381-402.

[32] Konstantopoulos Stasinos (2007). What's in a name? quite a lot. In Proceedings of the RANLP workshop 'Workshop on Computational Phonology'. Borovets, Bulgaria.

[33] Ignat Camelia, Bruno Pouliquen, António Ribeiro & Ralf Steinberger (2003). Extending an Information Extraction Tool Set to Central and Eastern European Languages. Proceedings of the RANLP Workshop Information Extraction for Slavonic and other Central and Eastern European Languages (IESL'2003). Borovets, Bulgaria.

[34] Steinberger Ralf, Bruno Pouliquen & Camelia Ignat (2005). Navigating multilingual news collections using automatically extracted information. Journal of Computing and Information Technology - CIT 13, 2005, 4, 257-264..

[35] Pouliquen Bruno, Ralf Steinberger & Camelia Ignat (2003). Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus. In: Proceedings of the EUROLAN Workshop Ontologies and Information Extraction at the Summer School The Semantic Web and Language Technology - Its Potential and Practicalities. Bucharest, Romania.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-241

Mining the Web to Build a Complete, Large-Scale Language Model

Gregory GREFENSTETTE
CEA LIST, Commissariat à l'Energie Atomique
BP 6, 92265 Fontenay-aux-Roses, France

Abstract. In order to detect the extraordinary, a computer must have a model of ordinary interactions. There is enough text on the web in the 420 languages that are represented there for language models to be built. All the text available for most languages can be crawled in a few months. These language models can show how words are normally associated. Unusual associations can then be detected automatically. This article presents an overview our work in building a very large and complete language model from the web for the French language.

Keywords. language model, WWW, corpus, text mining

Introduction

Over 6000 languages are thought to be spoken throughout the world [1], but only a little more than 400 have a web presence. For those languages, it should be possible to extract all the words present on the web and to produce a language model for each language.

1. Building a Model

We are currently working on building of a complete language model for the French language. By language model, we mean, that for each word in the French lexicon, we will know the relative frequency of this word on the web, the words with which it enters into syntactic relations with their relative frequency (for example, how many times in 1 million words we found the subject-verb relation *dog-barks*), the noun phrases that contain the word (with their relative frequencies) and the words that are most often found around the word (within windows of 5 or 10 words).

The steps involved in building the language model are the following. (Step 1) produce a wordlist for your language. We start with a list of surface forms for the language. We already possessed a full form lexicon for French. For other languages see http://borel.slu.edu/crubadan/stadas.html, Kevin Scannel's which gives statistics of lexicons generated from his web crawler An Crúbadán. Decompiling ISPELL lexicons http://lasr.cs.ucla.edu/geoff/ispell-dictionaries.html can be another source of word forms. (Step 2) Gather URLs that cover your language. For each word in the lexicon, send off a web query to a search engine. To avoid interlingual homographs, anchor the query in your language using common words. For French, we sent off queries for pages that contained each word we were looking for, plus the common French words "et" "le" "la" "que" "pour". In this manner, we collected 4 million URLs that covered the French language. This set of URLs is the seed for a web crawl. This step also produces the relative frequency of each mot in the language (in terms of web pages, which is a good approximation of real frequency for non stopwords) (Step 3) Fetch pages. Using this list, we fetch pages using the Unix tool *wget*, verify the encoding and recode the text in UTF8 using the Unix tools *file* and *recode*, and then use *lynx* to extract the textual part of the page. This text is sent to a language identifier [2], and French pages are retained. We found as a general rule that 75% of our URLs fetched in step (2) above produce text that we can exploit. The others have either disappeared, timed out (2 sec), contain non French text, or are empty. (Step 4) For each downloaded text, we then tokenize, morphologically analyze, lemmatize, and perform dependency extraction [3]. All noun phrases containing a word are linked to that word. We also extract all non stopwords found in windows of 5 and 10 words on each word, as well as typed named entities (places, people, etc.) For example, from the sentence "*verify once again the spigot angle and tighten the nut*" we extract the following information *noun-modifier(spigot,angle)*, *verb_object(verify angle)*, *verb_object(tighten nut)* and the phrase *(spigot angle)* as well as other information detailed in [4].

These pieces of information about each word (the other words with which it is in dependency relations, the longer phrases that it appears in, and the words most often found near it) give a picture of the word, similar to the Word Sketches produced in [5], but derived from the web for the entire vocabulary of the language.

The tools we are developing are generic and can be applied to any language for which low-level parsers exist. See http://www.alias-i.com/lingpipe/web/competition.html for list of such parsers for other languages.

The purpose of creating a large scale model will be both as a tool for natural language processing tasks (choosing between alternatives in machine translation or speech transcriptions), as well as determining how different parts of the world fit together

References

[1]J. Paolillo, D. Pimienta, D. Prado Measuring linguistic diversity on the internet. Retrieved Oct 07, at http://unesdoc.unesco.org/images/0014/001421/142186e.pdf 2005

[2] G. Grefenstette. Comparing two Language Identification Schemes. In Proceedings of the 3rd Int'l Conference on the Statistical Analysis of Textual Data JADT'95. Rome, Dec. 1995

[3] N. Semmar, M. Laib, C. Fluhr, A Deep Linguistic Analysis for Cross-language Information Retrieval. In Proceedings of LREC 2006, Genoa, Italy 2006

[4] G. Grefenstette. Conquering Language: Using NLP on a Massive Scale to Build High Dimensional Language Models from the Web. In Proc CICLing 2007, Mexico City, Mexico, 2007

[5] A. Kilgarriff, P. Rychly, P Smrz, and D. Tugwell, D.. The sketch engine. In Proc. of EURALEX 2004, 105–116. 2004

[6] G. Grefenstette, The Color of Things:Towards the automatic acquisition of information for a descriptive dictionary. In Revue Française de Linguistique Appliquée, vol X-2, 2005

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-243

Integrating Text Mining and Link Analysis

Frizo JANSSENS [a,b,1], Wolfgang GLÄNZEL [b] and Bart DE MOOR [a]
[a] *Dept. of Electrical Engineering ESAT, Katholieke Universiteit Leuven, Belgium*
[b] *Steunpunt O&O Indicatoren, Dekenstraat 2, B-3000 Leuven, Belgium*

Abstract. The performance of unsupervised clustering and classification of knowledge fields is significantly improved by deeply merging textual contents with network structure.

We present a hybrid clustering method based on Fisher's inverse chi-square, which exploits information from both the text and graph worlds. It is complemented with a dynamic strategy of tracking clusters through time in order to unravel the structure and evolution of knowledge fields.

Keywords. Dynamic hybrid clustering, Fisher's inverse chi-square method

Introduction

The increased dissemination and consultation of huge amounts of information via the Internet and other communication channels, as well as the availability of personal information in several large-scale databases, lead to tremendous opportunities to improve intelligence gathering, profiling, and knowledge discovery processes. Paramount challenges still remain, however.

1. Overview

We show that accuracy of clustering and classification of knowledge fields is enhanced by incorporation of text mining and link analysis [1]. Both textual and link-based approaches have advantages and intricacies, and both provide different views on the same interlinked data set. In addition to textual information, links constitute huge networks that yield additional information. We integrate both points of view and show how to improve on existing text-based and link-based methods.

Firstly, we present the use of large-scale text mining techniques for information retrieval and for mapping of knowledge embedded in text. We demonstrate our text mining framework and a semi-automatic strategy to determine the number of clusters in an interlinked document set.

[1] Corresponding Author: Frizo Janssens, Katholieke Universiteit Leuven, Department of Electrical Engineering ESAT/SCD-SISTA, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium; E-mail: frizo.janssens@esat.kuleuven.be.

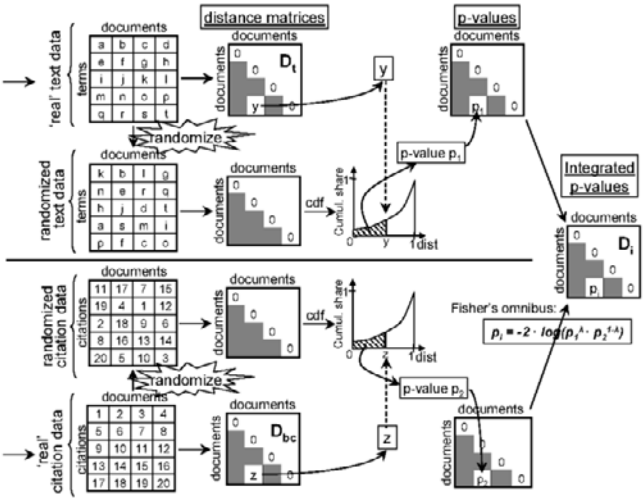


Figure 1. Example of distance integration by using *Fisher*'s inverse chi-square method. All text-based and link-based document distances in D_t and D_{bc} are transformed to p-values with respect to the cumulative distribution function of distances for randomized data. For randomization, term occurrences (and citations) are randomly shuffled between documents, while maintaining the average characteristic document frequency of each term. This randomization is a necessary condition for having valid p-values. In our setting, a p-value means the probability that the similarity of two documents could be at least as high just by chance. An integrated statistic p_i can be computed from the p-values for the textual data (p_1) and for the link data (p_2) by application of *Fisher*'s omnibus. The ultimate matrix with integrated p-values is the new integrated document distance matrix that can be used in clustering or classification algorithms.

Secondly, we focus on analysis of large networks that emerge from many relationships between individuals or between facts. These networks are analyzed with techniques from bibliometrics and graph theory in order to rank important and central ideas or social leaders, for clustering or partitioning, and for extraction of communities.

Thirdly, we substantiate the complementarity of text mining and graph analytic methods and propose schemes for the sound integration of both worlds (see Figure 1 for an example). The performance of unsupervised clustering and classification significantly improves by deeply merging textual content with network structure. We develop a clustering method based on statistical meta-analysis, which significantly outperforms text- and link-based solutions.

Finally, we devise a methodology for dynamic hybrid clustering of evolving data sets by matching and tracking clusters through time [2]. The integrated dynamic stance allows for a better interpretation of the structure and evolution of knowledge fields.

References

[1] F. Janssens. *Clustering of scientific fields by integrating text mining and bibliometrics*. Ph.D. thesis, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium, http://hdl.handle.net/1979/847, 2007.

[2] F. Janssens, W. Glänzel and B. De Moor. *Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis*. In *proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD'07), San Jose, 2007, pp. 360–369.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-245

Using linguistic information as features for text categorization

Arturo MONTEJO-RÁEZ [1], Luis Alfonso UREÑA-LÓPEZ,
Miguel Ángel GARCÍA-CUMBRERAS and José Manuel PEREA-ORTEGA
University of Jaén, Spain

Abstract. We report on some experiments using linguistic information as additional features as part of document representation. The use of linguistic features on several information retrieval and text mining tasks is a hot topic, due to the polarity of conclusions encountered by several researchers. In this work, extracted information of every word like the *Part Of Speech*, *stem* and *morphological root* have been combined in different ways for experimenting on a possible improvement in the classification performance and on several algorithms. Our results show that certain gain can be obtained when these varied features are combined in a certain manner, and that these results are independent from the set of classification algorithms applied or the evaluation paradigm chosen, providing certain consistency to our conclusions in text categorization on the Reuters-21578 collection.

Keywords. Automatic text categorization, linguistic features, document representation

Introduction

We report on some experiments using linguistic information as additional features in a classical Vector Space Model [1]. Extracted information of every word like the *Part Of Speech* and *stem*, *morphological root* have been combined in different ways for experimenting on a possible improvement in the classification performance and on several algorithms, like SVM[2], BBR[3] and PLAUM.

The inclusion of certain linguistic features as additional data within the document model is being a subject of debate due to the variety of conclusions reached. This work exposes the behavior of a text categorization system when some of these features are integrated. Our results raise several open issues that should be further studied in order to get more consistent conclusions on the subject. Linguistic features may be useful or not depending on the task, the language domain, or the size of the collection. Nevertheless, we focus here on a very specific aspect: the way we combine features is also crucial for testing its effectiveness.

Automatic Text Classification (TC), or Automatic Text Categorization as it is also known, tries to relate documents to predefined set of classes. Extensive research has been carried out on this subject [4] and a wide range of techniques are applicable to solve this task: feature extraction [5], feature weighting, dimensionality reduction [6], machine

[1] Corresponding Author: Universidad de Jaén, Jaén 23071, Spain; E-mail: amontejo@ujaen.es.

learning algorithms and more. Besides, the classification task can be either binary (one out of two possible classes to select), multi-class (one out of a set of possible classes) or multi-label (a set of classes from a larger set of potential candidates). In most cases, the latter two can be reduced to binary decisions [7], as the algorithm used does in our experiments [8]. This is the reason why machine learning algorithms have been playing a central role in TC.

In order to do machine learning when dealing with documents, a proper representation of the document has to be built. So far, the most common strategy is to follow the *bag of words* approach, where words from the document are extracted, transformed in some way and then weighted according to their frequency of use within the document. In this manner, documents are represented as vectors, where each dimension corresponds to the weight of a given term (i.e. a lemmatized word or a multi-word mainly) in the document.

Due to the large amount of terms within any vocabulary, reduction strategies must be applied in order to reduce the dimensionality of these document vectors. For dimension reduction there are also several solutions, which we can broadly classify into two main approaches: feature selection and feature transformation. The former relies upon mechanisms that discard non relevant features in some way [5], [6], [9], while the second one is related to methods using representation in reduced dimension feature spaces, such as term clustering approaches [10] or Latent Semantic Indexing [11].

This work focuses on the early phase of document representation, deciding which information from the document is extracted as features. In a step forward to the bag of words, we study how some of the output data that we can obtain from Natural Language Processing (NLP) methods can enrich document representation by evaluating a text categorization problem as a proof of concept.

1. Considering linguistic features

In Natural Language Processing, the document is a source of valuable information related to the different levels of analysis that can be performed on a given text. Nowadays, several linguistic tools are available for analyzing our documents content and extracting lexical and syntactic information, along with emerging and more abstract information at semantic level. Some of the information that can be considered as available from text by applying NLP could be the morphological root of word (e.g. *construct* as replace for *constructed*; more examples in table1), a multi-word term (e.g. noun phrases like *tropical plant*), the resolution of anaphora (e.g. *Sara was playing cards with John and she asked him to leave* could be replaced by *Sara was playing cards with John and Sara asked John to leave*), part-of-speech (POS) analysis (e.g. *I (Pronoun) told (Verb) you (Pronoun)*), semantic roles, dependency trees as result of shallow parsing, and named entities (e.g. *United Nations* as a unique term).

Our hypothesis is that adding data from a higher level of abstraction will enrich our feature space with additional information whenever this data is related in some way. We believe this is due to the fact that information derived from base data by more abstracted reasoning incorporates new information, as that reasoning is performed on heuristics and knowledge beyond the scope of the problem domain (i.e. the explicit content of the document). That is, the knowledge behind NLP tools is aggregated to new features and should, therefore, be exploited by the system.

original word	morphological root	stem
communications	communication	commun
decided	decide	decid
becoming	become	becom
bought	buy	bought

Table 1. Examples of obtained stems and morphological roots

Now the question is: *how to incorporate this abstract information, to the Salton's Vector Space Model in a blind way?* We can find previous research on applying NLP to text categorization successfully in the work by Sable, McKeown and Church [12], but their method is based on a careful consideration and combination of linguistic features. Our concern is on adding some linguistic features as additional information into a traditional bag-of-words representation with no further processing. Of course, every possible combination of linguistic features is not considered here. Our goal is rather to prove that some of them could lead to certain enhanced versions of document representation. This assertion argues against some previous related work, like the one by Moschitti and Basili [13], but is consistent with the conclusions given by Bigert and Knutsson [14] and Pouliquen et al [15]. In this last work, the authors explored the possible benefits of incorporating stop-word removal, multi-word detection and lemmatisation, concluding that these were very limited in the case of multi-word treatment and lemmatization, but a remarkable one when eliminating stop-words.

Moschitti and Basili's research [13] incorporates POS tags, noun senses and complex nouns (multi-words) as features for text categorization. These enriched document representations have been generated and tested on Reuters-21578[2], Ohsumed[3] and 20-NewsGroups[4] benchmark collections. They found worthless improvements. We think that some possible combinations were missing, while in our research such combinations are studied.

2. Experiments

In this section, the algorithm applied for multi-label classification is introduced along with the description of the data preparation phase and the results obtained in the designed experiments.

2.1. Multi-label classifier system

In the *Adaptive Selection of Base Classifiers* (ASBC) approach [16] we basically train a system using the battery strategy (many classifiers working together independently), but *(a)*, we allow tuning the binary classifier for a given class by a balance factor, and *(b)* we provide the possibility of choosing the best of a given set of binary classifiers. To this end, the algorithm introduces a hyper-parameter α parameter resulting in the algorithm given in figure 1. This value is a threshold for the minimum performance allowed to a binary classifier during the validation phase in the learning process, although the class

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3] http://trec.nist.gov/data/filtering/

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/

still enters into the evaluation computation. If the performance of a certain classifier (e.g. F1 measure, described in next section) is below the value α, meaning that the classifier performs badly, we discard the classifier and the class completely. By doing this, we may decrease the recall slightly (since less classes get trained and assigned), but we potentially may decrease computational cost, and increase precision. The effect is similar to that of the *SCutFBR* [17]. We never attempt to return a positive answer for rare classes. In [16], it is shown how this filtering saves us considering many classes without important loss in performance.

```
Input:
    a set of training documents $D_t$
    a set of validation documents $D_v$
    a threshold $\alpha$ on the evaluation measure
    a set of possible label (classes) $L$,
    a set of candidate binary classifiers $C$
Output :
    a set $C' = \{c_1, ..., c_k, ..., c_{|L|}\}$ of trained
    binary classifiers
Pseudo code:
    $C' \leftarrow \emptyset$
    for-each $l_i$ in $L$ do
        $T \leftarrow \emptyset$
        for-each $c_j$ in $C$ do
            $\textit{train-classifier}(c_j, l_i, D_t)$
            $T \leftarrow T \cup \{c_j\}$
        end-for-each
        $c_{best} \leftarrow \textit{best-classifier}(T, D_v)$
        if $\textit{evaluate-classifier}(c_{best}) > \alpha$
            $C' \leftarrow C' \cup \{c_{best}\}$
        end-if
    end-for-each
```

Figure 1. Adaptive Selection of Base Classifiers algorithm

The binary base classifiers selected within our experimental framework have been: Support Vector Machines (SVM) [2] under its implementation in the SVM-Light package[5], Logistic Bayesian Regression [3] using the BBR software[6] and the Perceptron Learning Algorithm with Uneven Margins [18] implemented natively in the TECAT package (which itself implements the whole ASBC multi-label strategy)[7]. All base classifiers have been configured with default values.

[5] Available at `http://svmlight.joachims.org/`

[6] Available at `http://www.stat.rutgers.edu/ madigan/BBR/`

[7] Available at `http://sinai.ujaen.es/wiki/index.php/TeCat`

Table 2. Contingency Table for *i* Category

	YES is correct	NO is correct
YES is assigned	a_i	b_i
NO is assigned	c_i	d_i

2.2. Evaluation Measures

The effectiveness of a classifier can be evaluated with several known measures [22]. The classical "Precision" and "Recall" for Information Retrieval are adapted to the case of Automatic Text Categorization. From categorizing test documents using a trained system, a contingency table is completed (Table 2), and then the precision and recall are calculated following equations 1 and 2.

$$P_i = \frac{a_i}{a_i + b_i} \tag{1}$$

$$R_i = \frac{a_i}{a_i + c_i} \tag{2}$$

On the other hand, the precision and recall can be combined using the F_1 measure:

$$F_1(R, P) = \frac{2PR}{P + R} \tag{3}$$

In order to measure the average performance of a system, three measures can be used: micro-averaged precision P_μ, macro-averaged precision in a document basis $P_{macro-d}$ and macro-averaged precision in a category basis $P_{macro-c}$.

$$P_\mu = \frac{\sum_{i=1}^{K} a_i}{\sum_{i=1}^{K} (a_i + c_i)} \tag{4}$$

$$P_{macro} = \frac{\sum_{i=1}^{K} P_i}{K} \tag{5}$$

where K is the number of categories or the number of documents depending on the basis used.

Recall and F1 measures are computed in a similar way. In our experiments we have used these measures in order to prove the effectiveness of the studied system.

2.3. Data preparation

The data used was the "ModApte" split of the Reuters-21578[8] collection, a dataset well known to the research community devoted to text categorization problems [19]. This collection contains 9,603 documents in the training set, while the test set is composed

[8] Prepared by David D. Lewis. The collection is freely available from the web page `http://www.research.att.com/~lewis/reuters21578.html`

of 3,299 documents. Each document is assigned to an average of slightly more than 2 classes. Documents contain little more than one hundred words per document.

In order to verify the contribution of the new features, we have combined them to be included into the vector space model by preprocessing the mentioned collection through some of the analysis tools available in the GATE architecture[9] [20]. Thus, we have generated enriched collections in the following ways:

1. `word` (w): a corpus with just plain text without any additional parsing has been used as base case
2. `stem` (s): each word has been transformed by applying the classical Porter's Stemmer algorithm [21]
3. `root` (r): instead of words, we consider their lexical roots
4. `stem+POS` (s+p): stems are, in this corpus, attached to their identified Part-Of-Speech, thus, each feature is a pair `stem-POS` (represented in our naming convention by a "+" sign)
5. `word+POS` (w+p): every word is attached to the associated POS tag
6. `root+POS` (r+p): every lexical root is attached to the associated POS tag
7. `word-root-stem-pos` (w-r-s-p): finally, a corpus every all previous features are in the document as independent features

2.4. Results

When evaluating text categorization, micro-averaged measures have been traditionally chosen as indicators of system quality. In multi-label text categorization we could also consider the possibility of using two additional indicators: macro-averaged measures by document and macro-averaged measures by class. These two are totally different and depending on how we want to apply our system, this choice may be crucial to really understand the performance of a proposed solution. In this way, macro-averaged precision by document, for instance, will tell us about how precise the labels are that we assign to every single document. On the other hand, macro-averaged precision by class will tell us how precise we are in assigning classes to documents in general. Certain differences arise since most of the classes are normally seldom assigned to most of the documents (there are many rare classes in real classification systems). Therefore, macro-averaging by document is an interesting indicator when the system is intended for individual document labeling. Of course, the counterpoint here is that if we are good with most frequent classes, then macro-averaged measurements by document will report good results, hiding bad behavior on rare classes, even when rare classes may be of higher relevance, since they are better discriminators when labels are used for practical matters. In our study, these three evaluation paradigms have been included.

In tables 3, 4 and 5, F1, precision and recall measurements on all the experiments run are shown. The best results obtained according to the algorithm used have been highlighted in cursive. The results in bold represent the feature combination that reported best performance on each algorithm and each of the three evaluation paradigms considered.

We can draw some conclusions from these evaluation measurements. The main one, that the winning feature combination turned out to be *w-r-s-p*. The use of the morphological root performs better than using stemming in general, although without noticeable

[9] Available at `http://gate.ac.uk`

F1	*w*	*r*	*r+p*	*s*	*s+p*	*w+p*	*w-r-s-p*
SVM avg	0.8211	0.8302	0.8224	0.8283	0.8234	0.8233	**0.8358**
SVM dAVG	0.8040	0.8212	0.8065	0.8194	0.8060	0.8086	**0.8268**
SVM cAVG	0.4345	0.4984	0.4673	0.4979	0.4979	0.4637	**0.5208**
BBR avg	0.8323	0.8367	0.8358	0.8305	0.8345	0.8323	**0.8384**
BBR dAVG	*0.8323*	0.8367	0.8358	*0.8305*	0.8345	0.8323	**0.8384**
BBR cAVG	0.4972	0.5696	0.5201	0.5648	0.5134	0.5046	**0.5759**
PLAUM avg	*0.8337*	*0.8392*	*0.8388*	*0.8323*	*0.8384*	*0.8392*	***0.8412***
PLAUM dAVG	0.8238	*0.8375*	*0.8362*	0.8253	*0.8376*	*0.8376*	***0.8392***
PLAUM cAVG	*0.5323*	*0.6015*	*0.5531*	*0.5842*	*0.5528*	*0.5460*	***0.6126***

Table 3. Combined F1 measurements on different algorithms and feature sets

Precision	*w*	*r*	*r+p*	*s*	*s+p*	*w+p*	*w-r-s-p*
SVM avg	***0.9277***	*0.9150*	*0.9226*	*0.9147*	*0.9269*	*0.9263*	*0.9212*
SVM dAVG	0.8195	0.8364	0.8220	0.8354	0.8219	0.8253	**0.8420**
SVM cAVG	0.6933	0.7302	0.6997	0.7302	0.7176	0.7034	**0.7614**
BBR avg	**0.9204**	0.8956	0.9068	0.8873	0.9065	0.9107	0.9022
BBR dAVG	0.8348	**0.8450**	0.8421	*0.8393*	0.8400	0.8380	0.8441
BBR cAVG	0.7583	0.7948	0.7594	0.8005	0.7585	0.7420	***0.8170***
PLAUM avg	**0.9142**	0.8935	0.8992	0.9014	0.8959	0.9016	0.8937
PLAUM dAVG	***0.8368***	*0.8469*	*0.8472*	0.8366	*0.8474*	*0.8477*	*0.8476*
PLAUM cAVG	0.7532	*0.7997*	*0.7804*	*0.8008*	*0.7718*	*0.7679*	**0.8139**

Table 4. Combined precision measurements on different algorithms and feature sets

Recall	*w*	*r*	*r+p*	*s*	*s+p*	*w+p*	*w-r-s-p*
SVM avg	0.7364	0.7598	0.7418	0.7569	0.7407	0.7410	**0.7650**
SVM dAVG	0.8033	0.8223	0.8064	0.8199	0.8050	0.8075	**0.8277**
SVM cAVG	0.3448	0.4113	0.3777	0.4097	0.3783	0.3707	**0.4280**
BBR avg	0.7596	**0.7851**	0.7752	*0.7806*	0.7730	0.7663	0.7830
BBR dAVG	0.8228	**0.8444**	0.8362	*0.8396*	0.8337	0.8302	0.8413
BBR cAVG	0.3996	0.4800	0.4301	0.4766	0.4234	0.4122	**0.4848**
PLAUM avg	*0.7663*	*0.7911*	*0.7860*	0.7730	*0.7878*	*0.7849*	***0.7946***
PLAUM dAVG	*0.8279*	*0.8458*	*0.8440*	0.8307	*0.8466*	*0.8447*	***0.8477***
PLAUM cAVG	*0.4412*	*0.5220*	*0.4691*	*0.4999*	*0.4676*	*0.4598*	***0.5359***

Table 5. Combined recall measurements on different algorithms and feature sets

performance differences. This can explain why people still apply stemming algorithms, which are easier to implement. Categorization results do not seem to improve when using stems and roots as replacement for words without morphological normalization, although they are useful to reduce the feature space. On the other side, when combined, categorization performance improves. This makes us think that there exist synergistic dependencies among them.

In order to validate these observations, statistical significance has been computed by applying a two-tailored Wilcoxon test on the obtained results. This test is the non-parametric equivalent of the paired samples *t-test*. This implies the assumption that both distributions are symmetrical, in which case the mean and medians are identical. Thus, the null hypothesis (usually represented by H_0) considers that for the two distributions the median difference is zero.

Distributions have been generated for each feature combination and for each evaluation measure. Thus, at each evaluation measure we have 60 values (3 algorithms multiplied by 30, the measurements obtained for the 30 most frequent categories). In tables 6, 7, 8 we have the p-values obtained using the two-tailored signed rank test (Wilcoxon test) comparing each possible pair of feature combinations. Values related to statistically significant differences are shown in bold (i.e. those p-values below 0.05).

Precision	**w**	**r**	**s**	**w+p**	**r+p**	**s+p**	**w-r-s-p**
w	0.50000000	0.99975792	0.99999722	0.99795972	0.99996494	0.99871684	0.99879193
r	**0.00024208**	0.50000000	0.99120325	**0.00191301**	0.21569861	0.08772633	0.28198043
s	**0.00000278**	**0.00879675**	0.50000000	**0.00025781**	**0.02299721**	**0.01308712**	**0.01383293**
w+p	**0.00204028**	0.99808699	0.99974219	0.50000000	0.94710365	0.78149820	0.97230034
r+p	**0.00003506**	0.78430139	0.97700279	0.05289635	0.50000000	**0.01874444**	0.58880332
s+p	**0.00128316**	0.91227367	0.98691288	0.21850180	0.98125556	0.50000000	0.83800353
w-r-s-p	**0.00120807**	0.71801957	0.98616707	**0.02769966**	0.41119668	0.16199647	0.50000000

Table 6. Two-tailored Wilcoxon test over Precision

Recall	**w**	**r**	**s**	**w+p**	**r+p**	**s+p**	**w-r-s-p**
w	0.50000000	**0.00000004**	**0.00000041**	**0.03389618**	**0.00003132**	**0.00013093**	**0.0000000001**
r	0.99999996	0.50000000	0.69496983	0.99999625	0.99945972	0.99992046	0.21925151
s	0.99999959	0.30503017	0.50000000	0.99998501	0.99785479	0.99985558	0.07531379
w+p	0.96610382	**0.00000375**	**0.00001499**	0.50000000	**0.00019374**	**0.00856058**	**0.00000009**
r+p	0.99996868	**0.00054028**	**0.00214521**	0.99980626	0.50000000	0.59073375	**0.00002613**
s+p	0.99986907	**0.00007954**	**0.00014442**	0.99143942	0.40926625	0.50000000	**0.00000146**
w-r-s-p	1.00000000	0.78074849	0.92468621	0.99999991	0.99997387	0.99999854	0.50000000

Table 7. Two-tailored Wilcoxon test over Recall

F1	**w**	**r**	**s**	**w+p**	**r+p**	**s+p**	**w-r-s-p**
w	0.50000000	**0.00005713**	**0.00071825**	0.30361848	**0.00924403**	**0.00698185**	**0.00000046**
r	0.99994287	0.50000000	0.95565518	0.99969308	0.99808699	0.99932684	0.16276175
s	0.99928175	**0.04434482**	0.50000000	0.99728397	0.98571432	0.99709336	**0.01274590**
w+p	0.69638152	**0.00030692**	**0.00271603**	0.50000000	**0.00760854**	**0.01334894**	**0.00000191**
r+p	0.99075597	**0.00191301**	**0.01428568**	0.99239146	0.50000000	0.29375643	**0.00016408**
s+p	0.99301815	**0.00067316**	**0.00290664**	0.98665106	0.70624357	0.50000000	**0.00002037**
w-r-s-p	0.99999954	0.83723825	0.98725410	0.99999809	0.99983592	0.99997963	0.50000000

Table 8. Two-tailored Wilcoxon test over F1

Regarding precision, the use of the original text without processing is the best option. But in terms of recall and F1, root and stem features may be preferred. Although root and *w-r-s-p* combination show similar results, from the p-value of the second one over the first one, we can observe that *w-r-s-p* is close to overperform root with statisticall significance.

3. Conclusions and future work

Our results show that certain linguistic features improve the categorizer's performance, at least on Reuters-21578. A text classification system shows many degrees of freedom (different tuning parameters), and small variations can produce big deviations, but from the results above, it is clear that for any of the algorithms selected and on any of the evaluation paradigms, the feature combination *word-root-stem-pos* produces better results, but with small improvements compared to the other feature combinations, like morphological root, according to the F1 measure.

Though the gain in precision and recall is not impressive, we believe that further research has to be carried out in this direction, and we plan to study different integration strategies, also considering additional features like *named entities*, term lists and additional combinations of all these features in the aim of finding more synergy. Also, the impact of such information may be higher for full texts than short fragments of Reuters-21578 texts. Collections like the HEP [23] or the JRC-Acquis [24] corpora will be used to analyze this possibility.

At this final point, we would like to underline relevant issues regarding the usage of linguistic features that should also be studied. Some languages (Slavonic languages and Finno-Ugric) are more highly inflected, i.e. there are more variations for the same lemma than, for example, in English. Another important issue is the trade-off between possible errors in the generation of these features by the linguistic tools used and the benefit that their inclusion can produce on the final document representation. Word sense disambiguation may introduce more noise into our data. Also, the stemming algorithm, may perform badly in texts of specialized domains and may harm the final categorization results. Finally, the size of the collection, the length of the document and other characteristics of the data can determine whether the inclusion of certain features is useful or not. Therefore, many questions remain open and the research community still has work to do on this topic.

Acknowledgements

This work has been partially financed by the TIMOM project (TIN2006-15265-C06-03) granted by the Spanish Ministry of Science and Technology and the RFC/PP2006/Id_514 granted by the University of Jaén.

References

[1] Gerard Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. Technical Report TR74-218, Cornell University, Computer Science Department, July 1974.

[2] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[3] D. Madigan, A. Genkin, D. D. Lewis, and D. Fradkin. Bayesian multinomial logistic regression for author identification. In *25th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 803 of *AIP Conference Proceedings*. American Institute of Physics, August 2005.

[4] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[5] D. D. Lewis. Feature Selection and Feature Extraction for Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217, San Mateo, California, 1992. Morgan Kaufmann.

[6] Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

[7] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Proc. 17th International Conf. on Machine Learning*, pages 9–16. Morgan Kaufmann, San Francisco, CA, 2000.

[8] A. Montejo-Ráez and L.A. Ureña López. Binary classifiers versus adaboost for labeling of digital documents. *Sociedad Española para el Procesamiento del Lenguaje Natural*, (37):319–326, 2006.

[9] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers Inc., 1997.

[10] L. Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM Press, 1998.

[11] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[12] C. Sable, K. McKeown, and K. Church. Nlp found helpful (at least for one text categorization task). In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, PA., 2002.

[13] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: A comprehensive study. In *ECIR*, pages 181–196, 2004.

[14] Johnny Bigert and Ola Knutsson. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *Proceedings of Romand 2002 (Robust Methods in Analysis of Natural language Data)*, Frascati, Italy, July 2002.

[15] Bruno Pouliquen, Ralf Steinberger, and Camelia Ignat. Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus. In Amalia Todirascu, editor, *Proceedings of the workshop 'Ontologies and Information Extraction' at the EuroLan Summer School 'The Semantic Web and Language Technology'(EUROLAN'2003)*, page 8 pages, Bucharest (Romania), 2003.

[16] A. Montejo-Ráez, R. Steinberger, and L. A. Ureña López. Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In Vicedo J. L. et al., editor, *Advances in Natural Language Processing: 4th International Conference, EsTAL 2004*, number 3230 in Lectures notes in artifial intelligence, pages 1–12. Springer, 2004.

[17] Yiming Yang. A study on thresholding strategies for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 137–145, New Orleans, US, 2001. ACM Press, New York, US. Describes RCut, Scut, etc.

[18] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. Kandola. The perceptron algorithm with uneven margins. In *Proceedings of the International Conference of Machine Learning (ICML'2002)*, 2002.

[19] F. Debole and F. Sebastiani. An analysis of the relative hardness of reuters-21578 subsets. 2004.

[20] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and Y. Wilks. Experience of using GATE for NLP R&D. In *Proceedings of the Workshop on Using Toolsets and Architectures To Build NLP Systems at COLING-2000*, Luxembourg, 2000. `http://gate.ac.uk/`.

[21] M. F. Porter. *An algorithm for suffix stripping*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[22] D. D. Lewis. Evaluating Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Morgan Kaufmann, 1991.

[23] A. Montejo-Ráez. *Automatic Text Categorization of Documents in the High Energy Physics Domain*. PhD thesis, University of Granada, March 2006.

[24] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. *The 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 2142–2147, May 2006.

Security Applications

This page intentionally left blank

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-257

Statistical Techniques for Fraud Detection, Prevention and Assessment

David J. Hand[a,b] and David J. Weston[b,1]
[a] *Department of Mathematics, Imperial College, London SW7 2AZ*
[b] *Institute for Mathematical Sciences, Imperial College, London SW7 2PG*

Abstract. Statistical, data mining, machine learning and other data analytic tools are important weapons in the battle against fraud. The fraud environment is reviewed and the magnitude of fraud described. Banking fraud, and especially plastic card fraud, is examined in detail. Tools for evaluating fraud detection systems are described and different classes of detection methods are outlined. To illustrate, the novel method of peer group analysis is described in detail, and illustrated in real applications.

Keywords. fraud detection, outlier detection, supervised classification, crime, banking fraud, peer group analysis

Introduction

The *Concise Oxford Dictionary* defines fraud as ‘Criminal deception; the use of false representations to gain an unjust advantage.’ Such ‘false representations to gain an unjust advantage’ are old as humanity itself - older if one thinks of innumerable examples in which animals try to trick each other. Animal camouflage provides just one example. As far as humans are concerned, fraud occurs in all areas of life, although the motivations may differ. Perhaps the most common motivation is financial gain. This applies to banking fraud, telecommunications fraud, insurance fraud, health care fraud, internet fraud, etc. In other areas, however, the objective might be increased power (e.g. electoral fraud) or influence and reputation (e.g. scientific fraud). Sometimes motivations are mixed together: low level credit card fraud is used to fund terrorist organisations, but the financial objective is secondary to the ideological objective.

There are various social aspects to limiting and managing fraud. For example, a bank, which relies on the confidence of its investors in order to do business, might be unwilling to admit to being a target of fraud. Of course, since all banks are so targeted, one might be suspicious of any bank which denied suffering from fraud, to the extent of wondering what they had to hide. Complementary to that, if a bank is known to have highly effective fraud detection systems, and is known to rigorously pursue

[1] Corresponding Author: David Weston, The Institute for Mathematical Sciences, 53 Prince’s Gate, South Kensington, London, SW7 2PG, U.K.; E-mail: david.weston@imperial.ac.uk.

fraudsters, then perhaps it is less likely to be a fraud target. And this applies even if a bank is not *known* to have such systems, but is merely *believed* to have them. Reputation is all.

Related to these points are aspects of the *economic imperative*. It is not worth a bank spending $200m to stop $20m of fraud (provided potential losses due to damage to reputation, etc, are factored in). On a smaller scale, this principle manifested itself in a letter to the London *Times* on 13th August 2007, which said: ‘I was recently the victim of an internet fraud. The sum involved was several hundred pounds. My local police refused to investigate, stating that their policy was to investigate only for sums over £5000.’ This has disturbing implications, not only for the amount of fraud perpetrated, but also for the message it gives and for society as a whole. It suggests that a certain background level of fraud is almost being officially sanctioned.

Another, related, aspect of the economic imperative is the fact that (figuratively speaking) the first 50% of fraud is easy to stop, the next 25% takes the same amount of effort, the next 12.5% the same again, and so on, so that one can never stamp out all fraud. It is, again, a matter of deciding where to draw the line.

Finally, as far as economics goes, the resources available for fraud detection are always limited. In the UK about 3% of police resources are spent on fraud, and this will not increase. The police have other problems to deal with - although, as indicated by the comment on terrorism above, they are often linked.

The upshot of all this is that, if we cannot outspend the fraudsters, we must outthink them. We need to bring sophisticated modern technologies to bear, such as the detection technologies discussed in this paper.

Turning to these technologies, there are several problems with which they must contend. Firstly, fraud data sets are typically very large, certainly in terms of number of cases (transactions, accounts, etc) and often in terms of number of variables (characteristics of accounts and account holders, descriptors of transactions, etc.). Most of the variables will be irrelevant to classifying a case as fraudulent or not, and most of the cases will not be fraudulent. Indeed, the balance between fraudulent and non-fraudulent cases is often so dramatic (e.g. of the order of 1:1000 in credit card transactions) that it is a classic needle in a haystack data mining problem.

Things are further complicated by the dynamic nature of fraud. In general, the underlying domain evolves over time. Our forefathers did not have to worry about credit card fraud, phishing, pharming, 419 fraud, or telecommunications fraud because the technologies did not exist. The economic climate changes, encouraging some people to take the risks of fraud, or to take advantages of opportunities for fraud. The political climate also changes. Worse still, fraud evolves in a reactive way. This is sometimes called the ‘waterbed’ effect, and refers to the fact that, when certain kinds of fraud are prevented (perhaps by improved detection strategies) the fraudsters typically switch to another mode: push it down in one place and up it pops in another. Most financial fraud is perpetrated by gangs, and they are not likely to abandon their whole enterprise merely because one mode of fraud has been stopped. We shall have more to say about these issues, in the context of credit card fraud, below.

The temporal aspect of fraud also manifests itself in the need for quick detection in many applications. Ideally, an attempt to use a credit card fraudulently would be detected immediately, in time for the transaction to be prevented. Certainly, a detector which took months to determine that a fraud had been committed would be of limited value. Timeliness is very important.

The effectiveness of fraud detection systems can be enhanced by integrating data from a variety of sources. However, this carries its own dangers, such as that of disclosure risk leading to identity theft. Moreover, data integration aggravates the high dimensionality, many variables problem, as well as increasing the risk of introducing errors. Errors in fraud detection systems can be serious. While it is crucial to minimise the number of frauds misclassified as legitimate, the opposite error of misclassifying a legitimate case (account, account holder, transaction, etc) as fraudulent also carries a cost in terms of the bank's reputation. We also discuss such matters in more detail below.

The sources of data for fraud detection depend very much on the domain, and so does the nature of the data. Numerical data are common, but image data (face recognition, numberplate recognition), text data (health insurance data), and other types are increasingly important.

Needless to say, the use of multiple sources of data, and the storage of large databases describing individuals and their behaviour, raises issues of civil liberties. Unfortunately, we do not have space to go into these in this article.

Reviews of statistical and other approaches to fraud detection are given in [1,2,3].

Section 1 of the paper examines how large is the fraud problem - a far from straightforward question. Section 2 focuses down on banking fraud. Before effective fraud detection systems can be developed, it is necessary to know what one means by 'effective' - to develop criteria by which methods may be assessed - and Section 3 examines such criteria. Section 4 then gives a lightning review of the strengths and weaknesses of the main classes of methods. One particular method, a novel approach called *peer group analysis* is described and illustrated in detail in Section 5.

1. How Big is Fraud

Determining the extent of fraud is extremely difficult, for a variety of reasons. Often the definition of what constitutes fraud is ambiguous. This may be because of intrinsic uncertainty about the underlying motive or because of administrative approaches to classification. For example, someone who declares bankruptcy or claims their credit card has been stolen may be doing so legitimately, or may be perpetrating a fraud, and we are unlike ever to know which was the motive. In administrative terms, theft from an ATM may be classified as theft, but could equally well be classed as fraud. The news media, which seem to prefer bad news over good, have a tendency to preferentially report larger figures. For example, [4] give a range of £6.8 billion to £13.8 billion for fraud in the UK, but the news reports focuses almost without exception on the larger figure. Furthermore, as we have already noted, the fraud environment changes rapidly, and in a reactive way, so that the figures of [4] are now very out of date. They are likely to be underestimates, if only because the economy has grown. Complicating things is the fact that fraud, by definition, may be slow in being discovered, or may not be discovered at all. Estimating the size of fraud is very much a case of estimating the size of an iceberg, which is largely hidden from sight. Finally, there are different kinds of loss. There is the immediate loss due to the fraud, the cost of detection and prevention strategies, the cost of lost business while systems are rebuilt (or a card is replaced), the opportunity cost, and the deterrent effect on the spread of e-commerce. [5] have explored these various issues in the context of fraud in the UK.

Table 1. Fraud and money laundering figures from the U. S. Department of Justice, Federal Bureau of Investigation, Financial Crimes Report to the Public, Fiscal Year 2006, [7].

	Cases	Convictions	Recoveries $m	Fines $m
Corporate Fraud	490	124	42	14
Securities and Commodities Fraud	1,165	164	21	81
Health Care Fraud	2,423	534	1,600	173
Mortgage Fraud	818	204	1	231
Identity Theft	1,255	405	4	1
Insurance Fraud	233	54	3	-
Mass Marketing Fraud	147	44	-	87
Asset Forfeiture/Money Laundering	473	95	3	-
Total	7,004	1,624	1,674	587

Despite all these difficulties, various values have been given for fraud losses. In the UK, estimates range from the £7 billion lower estimate of [4] to a huge £72 billion estimate from [6]. In the US, the Association of Certified Fraud Examiners has estimated that US organisations lose 5% of their annual revenues to fraud, so that, applying this figure to the estimated 2006 US GDP yields ‘approximately £652 billion in fraud losses’. In the US, the FBI is responsible for investigating fraud. [7] says ‘These crimes are characterized by deceit, concealment, or violation of trust, and are not dependent upon the application or threat of physical force or violence. Such acts are committed by individuals and organizations to obtain personal or business advantage. The FBI focuses its financial crimes investigations on such criminal activities as corporate fraud, health care fraud, mortgage fraud, identity theft, insurance fraud, mass marketing fraud, and money laundering.’ [7] gives figures for their investigations, reproduced in Table 1. If the gross estimates above are anything to go by, a huge amount of fraud goes unpunished.

Taking this down to a personal level, the specific problem of identity theft is one which has received substantial media coverage in recent years. In identity theft fraudsters acquire sufficient information about you to pass themselves off as you - in obtaining credit cards, telecoms services, bank loans, and even mortgages and apartment rentals. Even worse, they will give your name and details if charged with a crime. This leaves you with the debts and problems to sort out. Figures suggest that there were around 10 million victims of identity theft in the US in 2003, with an average individual loss of around $5,000, and that it can take up to two years to sort out all the problems (clearing credit records, etc) and reinstate one’s reputation after the theft has been detected.

2. Fraud in Banking

Banking fraud has many aspects, ranging from money laundering to credit card fraud. In this paper our main focus will be personal banking fraud, involving such things as credit cards, mortgages, car finance, personal loans, current accounts, savings accounts, etc, and we will be especially concerned with plastic card fraud.

Plastic card fraud detection problems have a number of characteristic features in addition to the data characteristics of large dimensionality, many cases, lack of balance, and reactive population drift mentioned above. In particular, the costs of the different kinds of misclassification are different, there are many different ways of committing card fraud, there is a delay in learning the true class labels, the transaction arrival times are random, and objects may have incorrect labels.

The lack of balance means that one has to be very careful about how detection methods are evaluated. In particular it is possible to have very high sensitivity and specificity levels (say, each 0.99) with a very low true fraud rate amongst those classified as fraud (0.09, if the prevalence of fraud is 0.001). This matters because operational decisions must be made on the basis of the detection classification, and one does not want to irritate perfectly good customers by blocking their account. (Though, to some extent this is compensated for by the fact that, up to a point, customers are pleased to see that the bank is monitoring potential fraud.)

If the detector indicates a potential fraud, then the true class is learned essentially immediately (a phone call to the account holder). However, if the detector does not raise the alarm, then the true class is not determined until later - for example, until the account holder studies their monthly statement. This leads to selectivity bias: more suspicious transactions are more likely to be fraudulent, and they are also more likely to have their true class known. Related to this is the fact that a bank cannot always say for certain when a sequence of fraudulent transactions commenced.

Fraudulent transactions may be mislabelled as legitimate because the account holder fails to check their statement sufficiently rigorously. There are also further subtleties which make this problem rather different from the classic supervised classification paradigm. For example, in the latter, each object has a fixed unknown true class. But consider the following fraud scenario. Someone uses their card to make a series of high value legitimate transactions. But then, shocked by the total amount spent, they report their card as having been stolen. At the time of the transactions they were legitimate, and it is only by virtue of the card holder changing their mind that they have become fraudulent. In particular, the descriptive characteristics of the account and transactions have not changed; only the labels have changed.

Returning to reactive population drift, a graphic example of this occurred in the UK last year. On February 14th 2006, the UK card industry rolled out a chip and PIN system to replace the magnetic stripe and signature system. As a consequence certain kinds of card fraud decreased and others increased. In particular, the use of counterfeit cards declined, but cardholder-not-present fraud (e.g. phone or internet purchases) increased. Furthermore, the use of stolen UK credit card details abroad increased, where cloned magnetic stripe cards could be used in those countries which had not installed a chip and PIN system.

3. What is a Good System?

In principle a good fraud detection system would be one which classified fraudulent transactions as fraudulent, and legitimate ones as legitimate. But no method is perfect, so there will be some misclassifications. A measure of performance needs to measure the extent of these misclassifications in an appropriate way. Familiar performance criteria, including the Gini coefficient (equivalent to the area under an ROC curve), the Kolmogorov-Smirnov statistic, divergence, and misclassification rate are unsuitable for this purpose (see [8,9]). In general, since different performance criteria may lead to different models, it is important to choose a criterion which matches the objectives.

Table 2 shows the notation we will use to describe detection performance. Our aim is to combine the four counts shown in this table to yield a criterion which can be used for assessing performance and choosing between detectors. Ideally that means we should reduce them to a single number.

In order to completely define measures based on the counts in Table 2, we need to say what they are counts *of*. For example, they could be entire accounts, individual transactions, or groups of transactions. One common organisation uses accounts, with an account being flagged as possibly fraudulent if at least one transaction is so flagged. This has obvious weaknesses since it means that the probability that an account will be flagged as possibly fraudulent can be increased arbitrarily by increasing the length of time for which it is observed. Transactions are at least well-defined. The problem with using individual transactions is that they are highly variable: one might normally use a credit card for only small supermarket purchases, but then suddenly buy a £5000 air ticket with it. For this reason, *activity records* are sometimes used, being statistical summaries of small numbers of consecutive transactions. For example, one might take all the transactions in a day or week, or all of the last three or five transactions.

We recommend basing measures on the four counts in Table 2, where the counts are either of individual transactions or of activity records, and are total counts over the accounts in the data set that is being used to evaluate the detector. For example, $m_{n/f}$ is the total number of legitimate transactions incorrectly classified as fraudulent by the detector over all accounts in the database. Given that the raw elements being counted are individual transactions or activity records, and not accounts, then timeliness implicitly occurs through the count $m_{f/n}$. This tells us the number of fraudulent transactions which have been missed by the detector before either a fraud alarm is raised on a true fraud, or the observation period stops. This is an appropriate measure of timeliness since these are the transactions which cost money due to undetected fraud.

Table 2. Counts of true and predicted classes for evaluating fraud detection systems.

		True class	
		Fraud	Legitimate
Predicted class	Fraud	$m_{f/f}$	$m_{n/f}$
	Legitimate	$m_{f/n}$	$m_{n/n}$

The counts in Table 2 can be combined in various ways to reveal different aspects of performance. In [9] we proposed two measures.

$$T_1 = \left(m_{f/f} + m_{n/f} + km_{f/n}\right) \Big/ \left(km_f + m_n\right) \qquad (1)$$

where k is the relative cost of misclassifying a fraudulent case as legitimate, compared to misclassifying a legitimate case as fraudulent. The argument behind this measure is that investigating fraud alarms incurs some costs (and applies to all elements of counts in the top two cells) but failing to investigate a true fraud incurs a much larger cost.

$$T_2 : \text{minimise } m_{f/n} \text{ subject to } \left(m_{f/f} + m_{n/f}\right) = C \qquad (2)$$

Here the argument is that a bank will be able to afford to investigate C cases, and better detectors will minimise the number of frauds not detected for this cost.

A familiar approach to displaying classification performance is via the ROC curve. The ROC curve is produced by plotting $m_{n/n}\big/m_n$ on the vertical axis against $m_{f/n}\big/m_f$, where the positive class consists of the non-fraud cases, m_n is the total number of non-fraud cases and m_f the total number of fraud cases in the test population. A similar plot can be produced our situation. The horizontal axis remains the same, but the vertical axis now becomes $\left(m_{n/f} + m_{f/f}\right)\Big/\left(m_n + m_f\right)$. That is, the vertical axis shows the proportion of cases for which the fraud alarm is raised, and the horizontal axis shows the proportion of fraud cases classified as non-fraudulent. Random classification is represented by a diagonal line from the top left to the bottom. This is a different orientation from the ROC curve and was chosen deliberately so that they are not confused.

4. Detection Methods

There are three major types of method used for building detection rules: rule-based approaches, supervised classification, and anomaly detection methods. These different approaches should not be regarded as competitors, since they can be combined in a single system. Likewise, other, less widespread methods can also be combined with these. These include change point detection, multilevel methods, and link analysis. Change point detection methods monitor a system and look for sudden behavioural changes. Multilevel methods combine information at the transaction (or activity record) level with information at the account level (e.g. are certain types of account, or accounts with certain usage patterns intrinsically more vulnerable) and information at merchant level (e.g. certain types of merchant are more likely to be associated with fraud). Link analysis approaches are based on the observation that credit card fraud is typically perpetrated by organised crime syndicates. For example, studying records of where cards which have been used fraudulently were previously used may allow one to detect a common merchant where the details are being stolen.

Turning to the major approaches, they have different strengths and weaknesses (which is why they should be used together). Rule-based approaches need expert knowledge of past fraud behaviour, and are highly effective at detecting known fraud types but less so at detecting novel types. Supervised methods need examples of past fraud and can be effective at detecting similar types, but less so for novel types. Anomaly detection methods are good for detecting novel types of fraud, but less so for known types (they are not optimised for these). There appear to be few studies comparing the relative merits of the different approaches.

Examples of the two-class supervised classification approach, and a comparative study of different methods and of the appropriate length of activity records, are given in [10]. Examples and a comparative assessment of various different approaches to anomaly detection approaches are given in [11].

[11] focus on individual accounts, and investigate departures to the norm for each account. At the other extreme, one might try to build a model for the population of accounts, seeking departures from that population. A novel intermediate approach is provided by *peer group analysis* [12]. In peer group analysis, a ‘target’ account is matched to a subgroup of other accounts which have behaved similarly in the past, and the entire subgroup is followed to see if the target continues to behave similarly or suddenly begins to deviate. Sudden events (e.g. Christmas) which impact all accounts will not throw up anomalies in peer group analysis but will in standard anomaly detection methods. Because peer group analysis is a novel approach, so hopefully providing complementary detection facilities to the more standard approaches, the next section describes the method in detail and illustrates it in action.

5. Peer Group Analysis

Peer group analysis is a general method for monitoring behaviour of a population over time. The technique relies on the assumption that members of a population that are in some sense similar at the present time will behave similarly for some time into the future. To find similar members, we may use any measure suitable for the specific problem, using for example static data associated with each member or, as we will demonstrate, historical behaviour. Introducing some terminology, we say for each *target* member of the population we build a *peer group* of similar members. We monitor the behaviour of the target purely in terms of the behaviour of its peer group. Once again we can measure similarity any way we choose and it can be different from the measure of similarity used to find the peer groups. Peer group analysis has been used to detect stock fraud [13] and to find fraudulent behaviour in business transactions [14].

For fraud detection we use peer group analysis to monitor an account’s *outlier* behaviour with respect to its peer group. This has an interesting property that an outlier to a peer group may not be an outlier to the population and therefore peer group analysis has the potential for discovering unusual behaviour even though that behaviour might itself be considered usual with respect to the entire population. Also, as briefly mentioned above, this method is less likely to find outliers due to population level dynamics, which are unlikely to be sources of fraud.

There are certain technical issues that need to be addressed in order to apply peer group analysis to plastic card fraud detection. To place these issues in their appropriate context we first briefly describe plastic card transaction data.

5.1. Plastic Card Transaction Data

A plastic card issuer needs to be able to screen transactions in real time with the aim of allowing only legitimate transactions to complete. To facilitate this, each transaction has associated with it a detailed record of pertinent information, including, for example, the monetary value of the transaction, the time the transaction occurred, attributes of the card reading device. A Merchant Category Code is also provided, which is a numeric code that identifies the vendor in the transaction with a particular market segment. One issue that affects fraud detection methods in general is selecting appropriate features from this data. An issue that affects peer group analysis in particular is that account transaction histories are asynchronous data streams. That is to say the transactions need not occur at regular intervals. Crucially for peer group analysis we need to compare behaviour at the same time, so ideally we would prefer time aligned time series.

One property of the dataset used in the following analysis has ramifications for the design choices in the following algorithm. The dataset is real historical plastic card transaction data with all the frauds known to have occurred labelled. However, this label is not at the transaction level, but at the account level, to the nearest day. This means that we cannot say for certain which transactions are fraudulent and which are not. For this reason we decided to perform the analysis also at the account level and on a daily basis. That is to say, we perform fraud detection on each account once a day, at midnight.

We time align the data by extracting information at regular time intervals. Figure 1, shows transaction histories for 2 fictitious accounts over a 10 day period. Each bar represents one transaction where the height of the bar represents the amount spent. Accounts at day t are compared using time aligned summaries of their behaviour over a user specified preceding interval. The summary statistic for day t for the behaviour account A over the past d days, $x(t - d + 1, t, \mathrm{A})$, is the total amount withdrawn in this interval, the number of transactions and a measure of the spread of merchant category codes, full details can be found in [12]. There is a balance to be met concerning the length of the summary statistic, the longer the duration the more stable the statistic will be, however, the shorter the duration, the quicker the potential response to fraudulent activity.

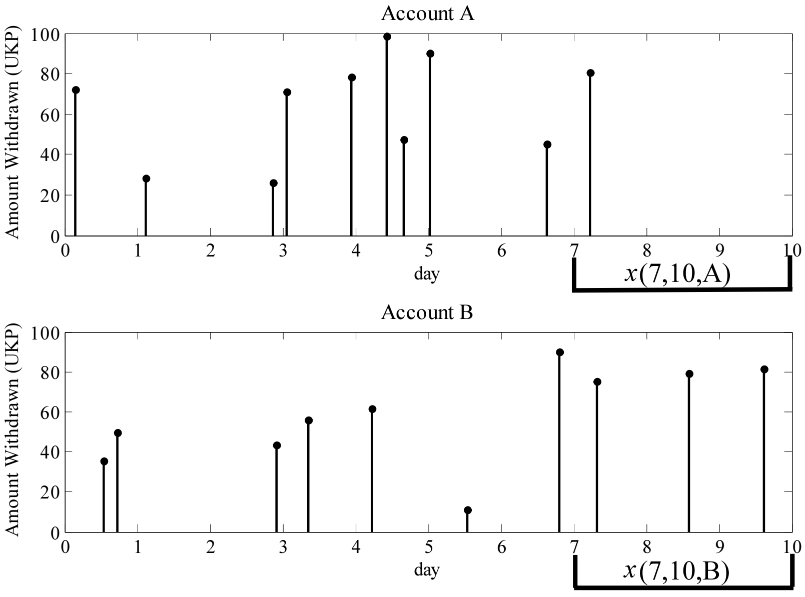


Figure 1. Extracting time aligned features from plastic card transaction data

5.2. Outlier Detection

We monitor the outlier behaviour of a particular target account by measuring the Mahalanobis distance of the target from the mean of its peer group, using the covariance matrix of the peer group. Should the target's distance exceed an externally set threshold, it is flagged as an outlier.

Not all accounts would have made daily transactions; so on each day we only examine those accounts that have been active. (We do not consider extended periods of inactive behaviour as suspicious.)

For an active account we determine those accounts in its peer group that have made at least one transaction within the current summary statistic interval. These are its active peer group. We wish to measure the behaviour of an active target with active accounts in its peer group. There is a possibility the size of the active peer group is not large enough to reliably measure its covariance matrix so we use an active peer group of fixed size.

Using peer groups for outlier detection may have problems arising from outlier masking and swamping. For example should fraudulent activity be missed, a defrauded account or accounts may contaminate other peer groups. This could potentially add outliers to the peer group itself, so disguising the fact that the target's behaviour is unusual. We use a heuristic approach to robustify the covariance matrix. An account that has deviated strongly from its own peer group at time *t* should not contribute to any active peer group at time *t*. We perform outlier detection twice. In the first pass we measure the Mahalanobis distance from each active account to its respective active peer group mean. In the second pass, for each active account we sort its active peer group members in order of ascending distance from their own peer groups. We then calculate the Mahalanobis distance using only the first p% of members. Peer group members that are not active on the day in question use their most recent distance evaluation.

5.3. Building Peer Groups

Currently building peer groups is the most computationally expensive part of the analysis. Fortunately in the context of a real system, this can be done offline. That is to say we do not have to build peer groups in the time allotted for a transaction to be validated. Indeed in the following example we use the same peer groups over an entire month's worth of data. There are a number of ways to measure similarity between time series. In the following we describe a simple method to measure similarity between each account's transaction histories that is robust against population level dynamics.

We partition the training data into n non-overlapping intervals. For the first interval we extract the summary statistic for each account. Then, for each pair of accounts we measure the squared Mahalanobis distance using the covariance matrix of the population of accounts. We repeat this procedure for the remaining n-1 intervals and we measure the separation between two accounts by simply summing their corresponding squared distances from each interval. Accounts that are not active in all the intervals are not considered candidates for peer group membership. To build a peer group for a target account we sort the remaining accounts in increasing order of separation. A peer group of size k is simply the first k accounts in this list.

It is not necessarily the case that peer group analysis can be deployed successfully on all accounts. We wish to identify those accounts where we are more likely to be successful. One way to do this is to measure how well a peer group tracks its target, this can be done by using the measure of separation described above. This time, however, we measure the squared Mahalanobis distance between an account and its peer group mean and use the same covariance matrix as above. The sum of these distances is a measure of peer group quality. We can then order all accounts with respect to their peer group quality, the smaller the value the better. We can use this list to screen accounts that have the worst peer group quality.

5.4. Experiments

From a large dataset consisting of real plastic card transaction data over a 4 month period, we selected accounts that had a large amount of transaction activity. We used only those accounts that had 80 or more transactions in the first 3 months and were entirely fraud free for this period. This created a reduced dataset of just over 4000 accounts. Approximately 6% of those accounts were defrauded in the final month. We used the first 3 months behaviour to build the peer groups and the subsequent month for evaluation. We fixed the active peer group size to 100. For outlier detection we used a summary statistic over 7 days. We robustified the outlier detector by using the first 50% of the active peer group members. Peer groups were built by partitioning the first 3 months of the data into 8 non-overlapping windows (see [12] for experiments using other partitioning granularities).

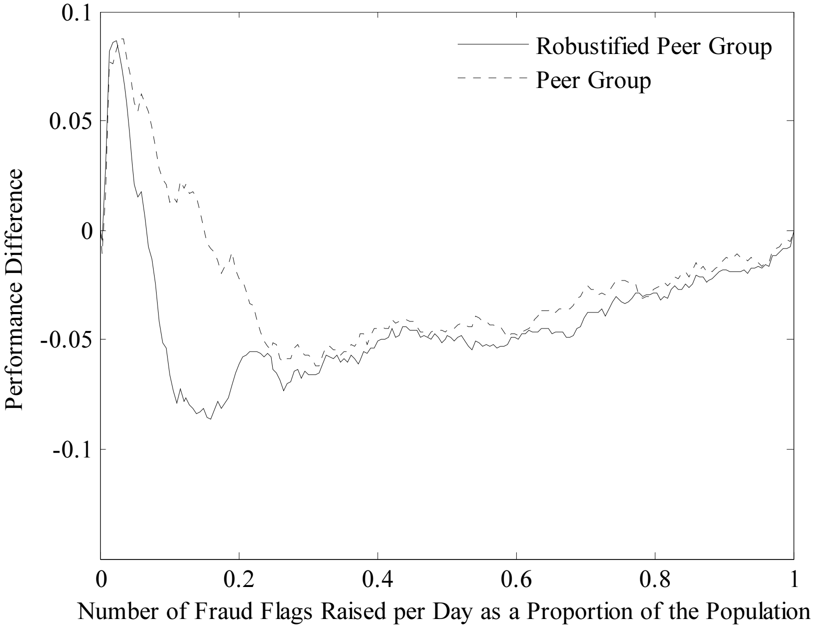


Figure 2. Performance of the robustified and non-robustified peer group analysis compared with the global outlier detector

In order to show that we can use peer group analysis to detect fraud and that it is doing more than simply finding population level outliers, we compared peer group analysis with a global outlier detector.

The method for global outlier detection proceeds as follows. On each day, for each active account we construct a peer group containing all the other accounts that are active during the summary window. Outliers to these peer groups will be outliers from the population.

Fraud detection is performed once a day at midnight. On each day we calculated twice the area under the performance curve (described in Section 3) for the peer group method and the global outlier detector method. A value of zero for the area corresponds to perfect classification whereas a value of one corresponds to random classification. We subtracted the peer group area from the global area. The difference will be negative should the peer group method outperform the global method.

Running the peer group method without robustification yields a difference of -0.0468 with a standard error of 0.0113. We note the standard error is likely to underestimate the true experimental error since the samples are not independent due to the use of a summary statistic interval that is longer than one day. Robustifying the peer group algorithm improves the performance and has a difference of -0.0799 and standard error 0.0090. Removing one third of the accounts, selected using the peer group quality metric, from the performance assessment produces the best performance with -0.1186 and standard error 0.0141. A plot of the mean difference in the proportion of frauds not found between the two peer group methods and the global method for the

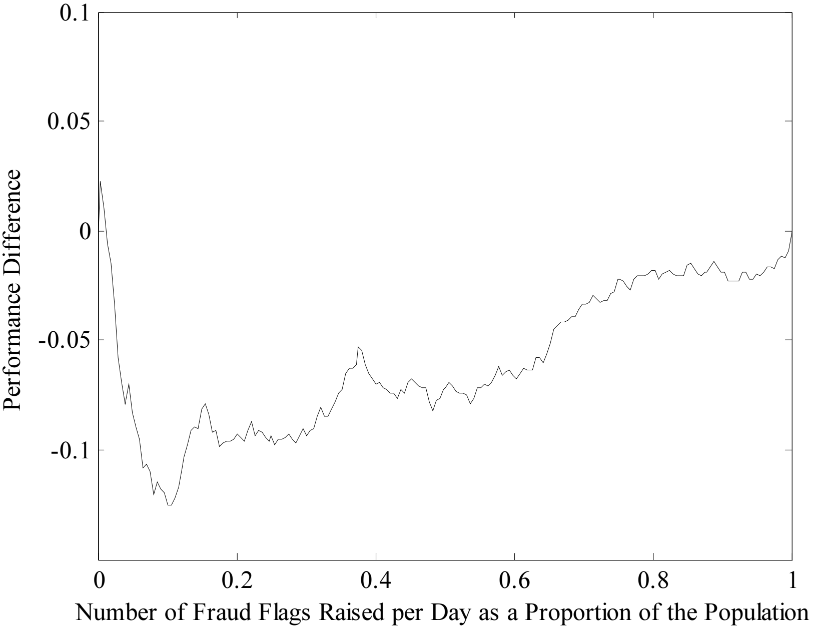


Figure 3. Performance of the robustified peer group analysis compared with the global outlier detector

proportion of fraud flags raised is shown in Figure 2. We see that the global method performs better when the proportion of fraud flags raised is very low, but is out performed for the rest of the domain. Figure 3 shows the performance difference between the global outlier detector and the robustified peer group method once we have removed the worst 33% of accounts (with respect to peer group quality). We see a similar pattern as before, this suggests that there are indeed some frauds that are global outliers, which the global outlier detector is better tuned to detect.

6. Conclusion

Fraud detection presents a challenging problem. In a sense it is a classic data mining problem, involving large high-dimensional data sets. This is further complicated by the dynamic nature of the data, which reacts to the detection methods put in place, and by sample selectivity bias and other distortions in the data.

Things are also complicated by the social and economic dimensions. People and organisations may not wish to admit to having been victims of fraud. There are limits to the amount which can be spent to prevent fraud. Both of these lead to the conclusion that society implicitly accepts a certain level of fraud.

Given that no fraud detection method is perfect, it is important that effective measures of performance of detection systems are developed. An inappropriate measure may not only be misleading, but it may also lead to a suboptimal choice of parameters when tuning a method.

There are various approaches to detecting fraud, but these should be viewed as complementary rather than competitors. Computing resources permitting, different methods should be used in conjunction since they may detect different kinds of fraud. Peer group analysis is a novel fraud detection strategy based on predicting expected future behaviour of an account from the behaviour of similar accounts. By doing so it eliminates the effect of sudden changes which impact on all accounts. It also complements anomaly detection techniques that measure deviations from the target account itself.

Acknowledgements

The work of David Weston described here was supported by the EPSRC under grant number EP/C532589/1: *ThinkCrime: Statistical and machine learning tools for plastic card and other personal fraud detection*. The work of David Hand was partially supported by a Royal Society Wolfson Research Merit Award.

References

[1] Bolton R.J. and Hand D.J. (2002) Statistical fraud detection: a review (with discussion). *Statistical Science*, 17, 235-255.

[2] Fawcett T. and Provost F. (2002) Fraud detection. In *Handbook of Knowledge Discovery and Data Mining*. Oxford: Oxford University Press. 726-731.

[3] Phua C., Lee V., Smith K., and Gayler R. (2005) *A comprehensive survey of data mining-based fraud detection research*. http://www.bsys.monash.edu.au/people/cphua/

[4] Jones S, Lewis D and Maggs P (2000), *The Economic Cost of Fraud: A Report for the Home Office and the Serious Fraud Office*, National Economic Research Associates, London.

[5] Blunt G. and Hand D.J. (2007) *Estimating the iceberg: how much fraud is there in the UK?* Working paper, Department of Mathematics, Imperial College, London.

[6] Mishcon de Reya (2005), *Protecting Corporate Britain from Fraud*, Mishcon de Reya, London.

[7] FBI (2006) http://www.fbi.gov/publications/financial/fcs_report2006/financial_crime_2006.htm

[8] Hand D.J. (2005) Good practice in retail credit scorecard assessment. *Journal of the Operational Research Society*, 56, 1109-1117.

[9] Hand D.J., Whitrow C., Adams N.M., Juszczak P., and Weston D. (2007) Performance criteria for plastic card fraud detection tools. *Journal of the Operational Research Society.* http://dx.doi.org/10.1057/palgrave.jors.2602418

[10] Whitrow C., Hand D.J., Juszczak P., Weston D., and Adams N.M. (2007) Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, Submitted.

[11] Juszczak P., Adams N.M., Hand D.J., Whitrow C., and Weston D.J. (2007) Off-the-peg or bespoke classifiers for fraud detection, *Computational Statistics & Data Analysis*, Submitted.

[12] Weston D.J., Hand D.J., Adams N.M., Juszczak P., and Whitrow C. (2008) Plastic card fraud detection using peer group analysis. *Advances in Data Analysis and Classification*, 2(1), 45–62.

[13] Ferdousi, Z. & Maeda, A. (2006), *Unsupervised outlier detection in time series data*, in 'Proceedings. 22nd International Conference on Data Engineering Workshops', pp. 51–56.

[14] Tang, J. (2006), *A peer dataset comparison outlier detection model applied to financial surveillance*, in 'ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)', IEEE Computer Society, Washington, DC, USA, pp. 900–903.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-271

Fitting Mixtures of Regression Lines with the Forward Search

Marco RIANI [a,1], Andrea CERIOLI [a], Anthony C. ATKINSON [b], Domenico PERROTTA [c] and Francesca TORTI [c]

[a] *Department of Economics, University of Parma, Italy*
[b] *Department of Statistics, London School of Economics, UK*
[c] *European Commission, Joint Research Centre, Institute for the Protection and Security of the Citizens, Support to External Security Unit, Ispra, Italy*

Abstract. The forward search is a powerful method for detecting unidentified subsets and masked outliers and for determining their effect on models fitted to the data. This paper describes a semi-automatic approach to outlier detection and clustering through the forward search. Its main contribution is the development of a novel technique for the identification of clusters of points coming from different regression models. The method was motivated by fraud detection in foreign trade data as reported by the Member States of the European Union. We also address the challenging issue of selecting the number of groups. The performance of the algorithm is shown through an application to a specific bivariate trade data set.

Keywords. Clustering, Outlier detection, Regression

Introduction

The Forward Search (FS) is a powerful general method for detecting outliers, unidentified subsets of the data and inadequate models and for determining their effect on models fitted to the data. The basic ideas started with the work of [11] and [1]. The power of the FS was considerably increased by [2] and [5] through the idea of diagnostic monitoring. They extended its applicability to a wide range of multivariate statistical techniques. Unlike most robust methods that fit to subsets of the data (see, e.g., [14] and [12]), in the FS the amount of data trimming is not fixed in advance, but is chosen conditionally on the data. Many subsets of the data of increasing size are fitted in sequence and a whole series of subsets is explored. As the subset size increases, the method of fitting moves from very robust to highly efficient likelihood methods. The FS thus provides a data dependent compromise between robustness and statistical efficiency.

In this contribution we describe a novel technique in which the FS is applied to detect clusters of observations following different regression models. Our assumptions are comparable to those underpinning latent class and model-based clustering methods [10], but our output is richer. The rationale is that if there is only one population the journey from fitting a few observations to all will be uneventful. But if we have two

[1] Corresponding Author: mriani@unipr.it or, for the anti-fraud application, domenico.perrotta@ec.europa.eu

or more groups there will be a point where the stable progression of fits is interrupted. Our tools for outlier detection and clustering are then developed from forward plots of residuals and distances computed from searches with either robust or random starting points. We also address a number of challenging issues, including selection of the number of groups and use of distributional results for precise identification of the outliers and the clusters.

Our focus is on clustering regression models. However, the ideas of clustering multivariate data are both more familiar and more easily explained, so we start in §1 with a brief review of the FS methodology for multivariate data. Two didactic examples of cluster detection for multivariate data are in §2. The new algorithm for detecting clusters of regression lines is introduced in §3. In §4 we show this algorithm in action for the purpose of detecting fraudulent transactions in trade data sets selected by the anti-fraud office of the European Commission and its partners in the Member States. The paper concludes in §5 with some remarks and suggestions for further development.

1. The Forward Search for Multivariate Observations

Outliers are observations that do not agree with the model that we are fitting to the data. Single outliers are readily detected, for example in regression by plots of residuals. However, if there are several outliers they may so affect the parameter estimates in the fitted model that they are not readily detected and are said to be “masked". Such multiple outliers may indicate that an incorrect model is being fitted.

The basic idea of the Forward Search is to start from a small subset of the data, chosen robustly to exclude outliers, and to fit subsets of increasing size, in such a way that outliers and subsets of data not following the general structure are clearly revealed by diagnostic monitoring. With multiple groups, searches from more than one starting point are often needed to reveal the clustering structure. In this section we restrict attention to data sets of multivariate continuous observations, for which outlyingness is measured through their Mahalanobis distances. The case of multivariate categorical data is addressed by [8].

The squared Mahalanobis distances for a sample $S(n) = \{y_1, \dots, y_n\}$ of n v-dimensional observations are defined as

$$d_i^2 = \{y_i - \hat{\mu}\}^T \hat{\Sigma}^{-1} \{y_i - \hat{\mu}\}, \qquad i = 1, \dots, n, \tag{1}$$

with $\hat{\mu} = \bar{y}$ the vector of sample means and

$$\hat{\Sigma} = \sum_{i=i}^{n} (y_i - \hat{\mu})(y_i - \hat{\mu})^T / (n - v)$$

the unbiased moment estimators of the mean and covariance matrix of the n observations. Throughout ^{T} denotes transpose.

In the FS the parameters μ and Σ are estimated from a subset $S(m) \subseteq S(n)$ of m observations, yielding estimates $\hat{\mu}(m)$ and $\hat{\Sigma}(m)$. From this subset we obtain n squared Mahalanobis distances

$$d_i^2(m) = \{y_i - \hat{\mu}(m)\}^T \hat{\Sigma}^{-1}(m) \{y_i - \hat{\mu}(m)\}, \qquad i = 1, \dots, n. \tag{2}$$

To start the search when the observations are assumed to come from a single multivariate normal population with some outliers, [5] pick a starting subset $S(m_0)$ that excludes any two-dimensional outliers. One search is run from this unique starting point. When a subset $S(m)$ of m observations is used in fitting, we order the squared distances and take the observations corresponding to the $m+1$ smallest as the new subset $S(m+1)$.

To detect outliers we examine the minimum Mahalanobis distance amongst observations not in the subset

$$d_{\min}(m) = \min d_i(m) \quad \text{for } i \notin S(m). \tag{3}$$

If this observation is an outlier relative to the other m observations, its distance will be “large” compared to the maximum Mahalanobis distance of observations in the subset. All other observations not in the subset will, by definition, have distances greater than $d_{\min}(m)$ and will therefore also be outliers.

In order to provide sensitive inferences it is necessary to augment the graph of $d_{\min}(m)$ with envelopes of its distribution. Detailed examples of such envelopes and of their use in the FS with moderate sized data sets are presented by [6] and [4]. A synthesis is provided in the next section.

For small data sets we can use envelopes from bootstrap simulations to determine the thresholds of our statistic during the search. These are found by performing the FS on many sets of data simulated from a standard multivariate normal distribution with the same value of n and v as our sample. Note that in the simulations we can use data generated from the standard normal distribution because Mahalanobis distances are invariant to linear transformation of the data. In the example here we make 10,000 such simulations. For each FS we monitor the values of $d_{\min}(m)$ defined in (3). As a consequence, for each value of m we have the empirical distribution of $d_{\min}(m)$ under the hypothesis of normality. The envelopes we use are the quantiles of this distribution. For example, the 99$\%$ envelope is that value which is the 1$\%$ point of the empirical distribution. With 10,000 simulations, this is the 100th largest value, so that 99 of the simulated values are greater than it. We calculate these quantiles for each value of m that is of interest.

For larger data sets we can instead use polynomial approximations. Theoretical arguments not involving simulation are provided by [13], together with a formal test of multivariate outlyingness and comparisons with alternative procedures.

For cluster definition, as opposed to outlier identification, several searches are needed, the most informative being those that start in individual clusters and continue to add observations from the cluster until all observations in that cluster have been used in estimation. There is then a clear change in the Mahalanobis distances as units from other clusters enter the subset used for estimation. This strategy seemingly requires that we know the clusters, at least approximately, before running the searches. But we instead use many searches with random starting points to provide information on cluster existence and definition.

In §4 we use envelopes to determine cluster membership. Since the size of the clusters has to be established, we need envelopes for several different values of n. Simulation then becomes time consuming unless n is very small. Calculation of the envelopes via the theoretical arguments in [13] become increasingly attractive as n increases.

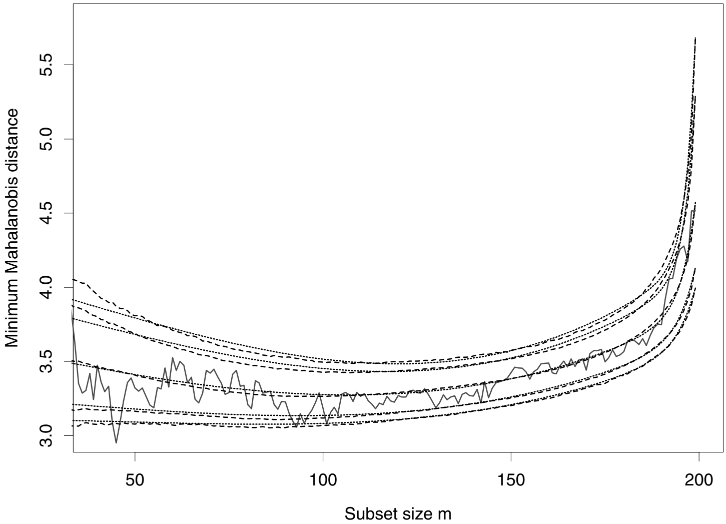


Figure 1. Swiss Heads data (n = 200): forward plot of minimum Mahalanobis distance with 1, 5, 50, 95 and 99% points: continuous lines, 10,000 simulations; dashed lines, interpolation. Simulations and approximation agree well. There is no evidence of any outliers.

2. Didactic Examples

The purpose of this paper is to provide methods for the relatively large and structured data sets that arise in the fraud detection problems illustrated in §4. However, we first look at brief analyses of two smaller examples, as a training of the eye in the interpretation of our forward plots for the detection of clusters.

2.1. *Swiss Heads Data*

The Swiss Heads data set was introduced by [9, p. 218]. It contains information on six variables describing the dimensions of the heads of 200 twenty year old Swiss soldiers. These data were extensively analysed, using the forward search, by [5]. The conclusion is that the observations come from a six dimensional normal distribution, from which there are no outliers. The plot of Figure 1 confirms this opinion. The envelopes in this figure were found both directly by simulation from the multivariate normal distribution and by parametric interpolation. The distances lie inside the envelopes, indicating complete agreement with the multivariate normal distribution.

2.2. *Example With Three Clusters*

We now look at a synthetic example with three clusters of four-dimensional correlated observations, to show how random start forward searches combined with envelope plots of forward Mahalanobis distances lead to the indication of clusters. There are 1,000 units in all, 250 of which are in the first group, 300 in the second and 450 in the third. The observations in each group are highly correlated and the third group lies between the other two, so that there is considerable overlapping. Of course, in our analysis we ignore the information about group structure, or even about the number of groups.

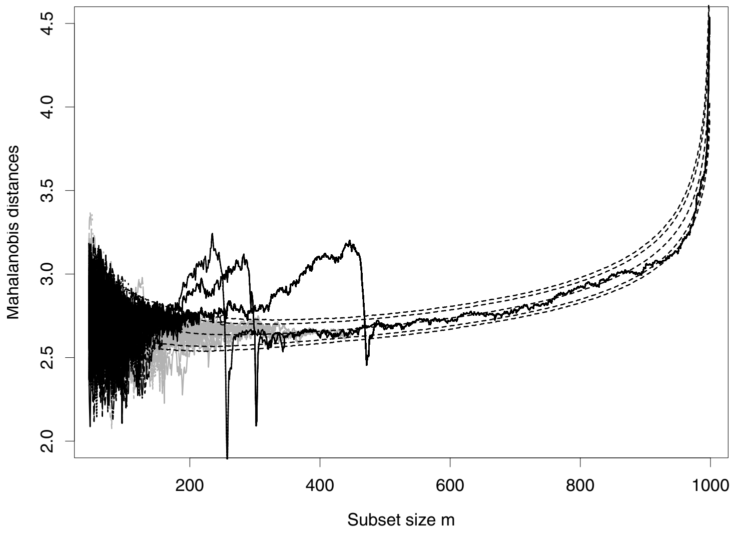


Figure 2. Three clusters of correlated normal variables: forward plot of minimum Mahalanobis distances from 200 random starts with 1%, 5%, 50%, 95% and 99% envelopes. Three clusters are evident.

We run 200 random start forward searches, each one starting with $m_0 = v + 1$, the smallest possible size and that which gives the highest probability of getting a subset consisting solely of observations from one cluster. The resulting forward plot of minimum distances is in Figure 2. The forward searches in this plot fall into four classes: those that start in each of the three groups and those that, from the beginning of the search, include observations from at least two groups. From around $m = 150$ the searches with observations from only one group start to lie outside the envelopes. These curves reach a peak and then suddenly dip below the envelopes as relatively remote observations from the other groups enter the subset used in fitting. From a little after $m = 500$ there is a single forward plot, in which a common mean and common covariance matrix are calculated from observations in more than one group, so that the group structure is no more apparent.

The approximate values of m at the three peaks are: 230, 290 and 450. Despite the overlapping nature of the groups, our method has initially indicated clusters for 97% of the observations. For precise definition of the clusters, we interrogate the subsets $S(m)$ for those trajectories where there is evidence of a cluster structure. The membership of each of the three subsets giving rise to the peaks in Figure 2 can be illustrated using the 'entry' plot of [5]. Cluster 1 includes most of the units of Group 1 and no other units. Cluster 2 contains the majority of the units in Group 2 and some borderline units from Group 3. The intermediate Group 3 is the most misclassified, as is to be expected.

One way to confirm this tentative identification is to run searches on individual clusters. If the peak for a particular cluster in the forward plot analogous to Figure 2 occurs when $m = n_c$, we include the next few units to enter and then run a search on these n_c^+ units, superimposing envelopes for a sample of size n_c^+ as we did in §2.1 for a single population. If no outliers are found, we have a homogenous cluster and increment n_c^+ to check whether we have failed to include some units that also belong to the cluster. If outliers are detected, we delete the last observation to enter, reduce the sample size by one and superimpose envelopes for this reduced sample size. Eventually we obtain the largest group of homogenous observations containing no outliers. Examples of this procedure are given by [4], together with comparisons with other clustering methods such as k-means which completely fails.

3. Mixtures of Regression Hyperplanes

3.1. Regression Diagnostics

The cluster analysis of multivariate data is well established as is the use of the FS to determine the clusters, especially for data that are multivariate normal. However, the regression framework is different from that of multivariate analysis and there is comparatively little work on clustering regression models. Here, our interest is in clustering by regression hyperplanes. The Forward Search is easily adapted to this regression problem, keeping the same philosophy but with regression-specific ingredients. In particular, distances are replaced by regression residuals.

We now have one univariate response Y and v explanatory variables $X_1, \ldots, X_v$ satisfying

$$E(y_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_v x_{iv} \tag{4}$$

with the usual assumptions of independent, additive errors of constant variance (see, for example, [15]).

Let b be a $(v+1)$ vector of constants. For any b we can define n residuals

$$e_i(b) = y_i - (b_0 + b_1 x_{i1} + \cdots + b_v x_{iv}). \tag{5}$$

The least squares estimate $\hat{\beta}$ is the value of b in (5) that minimizes the sum of squares

$$R(n, b) = \sum_{i=1}^{n} e_i^2(b). \tag{6}$$

Likewise, the estimate $\hat{\beta}(m)$, obtained by fitting the regression hyperplane to the subset $S(m)$, minimizes the sum of squares $R(m, b)$ for the m observations $\in S(m)$. From this estimate we compute n squared regression residuals

$$e_i^2(m) = [y_i - \{\hat{\beta}_0(m) + \hat{\beta}_1(m)x_{i1} + \cdots + \hat{\beta}_v(m)x_{iv}\}]^2 \qquad i = 1, \cdots, n$$

the $m+1$ smallest of which are used to define the new subset $S(m+1)$.

The search starts from an outlier-free subset of $m_0 = v+1$ observations found using the least median of squares criterion of [14]. We randomly take an appreciable number of samples of size m_0, perhaps 1,000, estimate b in (5) for the observations in each sample and take as $S(m_0)$ that sample for which

$$M(m_0, b) = \text{median}_{i \in S(M_0)} e_i^2(b) \tag{7}$$

is a minimum. In practice, [14] recommend a slight adjustment to allow for the estimation of β.

To detect outliers in §2 we used the minimum Mahalanobis distance among observations not in $S(m)$. We now instead examine the minimum deletion residual amongst observations not in the subset, which is the t test for detection of individual outliers.

Let X be the $n \times (v+1)$ matrix of the explanatory variables $X_1, \ldots X_v$ with the addition of a column of ones for the constant term in (4). The ith row of X is x_i. Then the minimum deletion residual is defined as

$$r_{\text{min}}(m) = \min \frac{|e_i(m)|}{s(m)\sqrt{[1 + x_i^T\{X^T(m)X(m)\}^{-1}x_i]}} \quad \text{for } i \notin S(m), \tag{8}$$

where $s(m)$ is the square root of $s^2(m) = R\{m, \hat{\beta}(m)\}/\{m-(v+1)\}$, the mean square estimator of the residual variance $\sigma^2 = E\{y_i - E(y_i)\}^2$ and $X(m)$ is the block of X with rows indexed by the units in $S(m)$. The quantity $h_i = x_i^T\{X^T(m)X(m)\}^{-1}x_i$ is the "leverage" of observation i. Observations with large values of h_i are remote in the space of the explanatory variables and can, as we shall see in Figures 4 and 5, cause perturbations in forward plots when they join $S(m)$.

The FS for regression is given book-length treatment by [2]. Inferences about the existence of outliers require envelopes of the distribution of $r_{\text{min}}(m)$, similar to those plotted in §2. Such envelopes are described by [3].

3.2. A Forward Algorithm for Clustering Observations Along Regression Hyperplanes

We now suppose that the observations come from g regressions models (2) with different and unknown parameter values. Our aim is to allocate each unit to its true model and to estimate the corresponding parameters. Also the number g of component models is not known in advance.

Clusterwise regression is the traditional technique for achieving this goal [16]. A more modern probabilistic approach is to fit the joint density of the n observations as a mixture of regressions models [7, §14.5]. However, both methods may suffer from the presence of outliers and/or strongly overlapping clusters. Another shortcoming of these methods is that they do not provide formal tests to justify the need of an additional component. Our proposal is to use the Forward Search for determining and fitting the g components of the regression mixture.

Our forward algorithm combines the strategies outlined in Sections 1 and 3.1. It consists of the following steps:

1. Let $n^*(j)$ be the size of the sample to be analysed at iteration j. At the first iteration $n^*(1) = n$;
2. The FS for regression is applied to these $n^*(j)$ observations. The search is initialised robustly through the least median of squares criterion applied to all n observations and progresses using the squared regression residuals $e_i^2(m), i = 1, \ldots, n^*(j)$;
3. At each step m of the FS, we test the null hypothesis that there are no outliers among the $n^*(j)$ observations. The test is performed using the minimum deletion residual (8);
4. If the sequence of tests performed in Step 3 does not lead to the identification of any outliers, the sample of $n^*(j)$ observations is declared to be homogeneous and the algorithm stops by fitting the regression model (4) to this sample. Otherwise go to Step 5;

5. Let m^* be the step of the FS in which the null hypothesis of no outliers is rejected by the sequence of tests of step 3. Then the observations in $S(m^*)$ identify one mixture component, i.e. one cluster of m^* observations following (4). Fit the regression model (4) to this cluster;
6. Remove the cluster identified in step 5. Return to Step 1 with a reduced sample size, by setting $n^*(j+1) = n^*(j) - m^*$.

The algorithm leads to the identification of g regression models, one for each iteration. The tests performed in step 3 ensure that each component of the mixture is fitted to a homogeneous subset. The tests are robust and are not influenced by outliers or by observations falling between the groups. Indeed, such observations, which are relevant for fraud detection, are clearly revealed by our forward diagnostic plots during the search. Note also that the method does not force all observations to be firmly clustered into one of the g components. Borderline units are recognized as intermediate between clusters and can thus be inspected separately.

4. Application to European Union Trade Data and Anti-fraud

In this Section we show how the FS can be used with European Union (EU) foreign trade data as reported by the EU Member States (MS) to detect anomalies of various kinds (e.g. recording errors), specific market price dynamics (e.g. discounts in trading big quantities of product) and cases of unfair competition or fraud. In particular fraud may be associated with anomalously low reported prices that could result in underpayment of taxes.

We use one concrete example to introduce the application context, the data and the statistical patterns of interest, i.e. *outliers* and *mixtures of linear models*. The European Commission's Joint Research Centre detects these patterns in data sets including millions of trade flows grouped in a large number of small to moderate size samples. The statistically relevant cases are presented for evaluation and feed-back to subject matter experts of the anti-fraud office of the European Commission and its partner services in the Member States.

We use the example to illustrate the regression approach introduced in Sections 3. The method of multivariate clustering of 1 is not informative when the data have a regression structure, so we do not consider it any further for this example.

4.1. European Union Trade Data

The data in Figure 3 refer to the quantity (x axis) and the value (y axis) of the monthly import flows of a fishery product into the European Union from a certain third country. The solid dots are the flows to a Member State that we call MS7, and the black circles are the flows to other Member States. We can clearly see that the two groups of observations are distinct and we could fit two linear regression models using the observations in the two groups. We use the slope of these linear models as an estimate of the import price of the flows in the respective groups.

There is also an observation, the open circle on the bottom-left, that does not follow the regression line fitted to the observations of the same category. Rather, it appears in the distribution of the solid dots. This "abnormal" black circle is a single flow to a Member

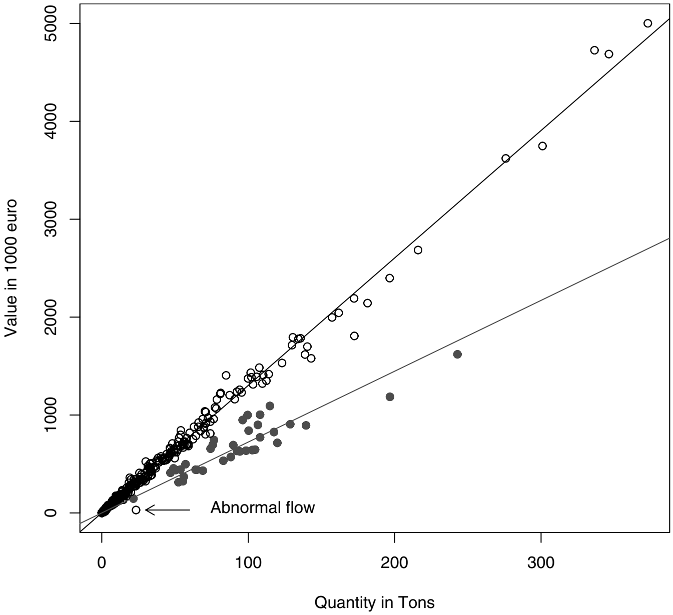


Figure 3. Quantities (in tons) and values (in thousands of euros) of 677 monthly imports of a fishery product from a third country into the EU, over a period of three years. Flows to MS7 (solid dots) and flows to the other Member States (open circles) form distinct groups following different regression lines. On the bottom-left an abnormal single flow to MS11.

State that we identify as MS11. The *unit value* of this flow, obtained by dividing the value by the corresponding quantity, is so small ($\sim$ 1.27€/Kg) compared to the market price of this specific fishery product (12.5€/Kg in 2005[2]) that we may suspect an error in recording the data. Although from a data quality point of view it might be worth investigating the validity of this data record, from the economical point of view we are unlikely to be interested in a trade flow of such volume ($\sim$ 20 Tons).

Much more importantly, the distribution of the solid dots indicates that the imports of MS7 are systematically underpriced in comparison with the imports of the other Member States. This indication is of appreciable economic relevance since MS7 imported about 20% ($\sim$ 3300 Tons) of the total EU imports of this product in our reference period.

Our data sets consist of thousands of samples similar to this example. Therefore we need to detect the outliers and to estimate the mixtures of linear components automatically and *efficiently*. We require high computational and statistical efficiency of the algorithms; they should detect a manageable number of outliers in reasonable time and with reasonable statistical power. But, for the anti-fraud subject matter experts, the concept of efficiency is also related to the problem of extracting cases of possible operational interest from the statistically relevant patterns that we detect with our algorithms. We will not address this issue here.

[2] Source: "European Fish Price Report", a GLOBEFISH publication (http://www.globefish.org).

4.2. Fitting Mixtures of Regression Lines

Our data have a clear regression structure (Figure 3) and we propose now to analyse them with the regression approach of Section 3.1. This approach uses the squared regression residuals for progression in the search and the minimum deletion residual among the observations not in the subset to monitor the search and infer departures from linearity. We show that the iterative application of this approach, detailed in Section 3.2, suggests modelling the data with a mixture of at least four linear components of rather clear interpretability by subject-matter experts.

The initial subset is chosen by robust fitting, using the least median of squares (LMS) criterion. The first iteration leads relatively straightforwardly to the identification of a first linear regression component with 344 homogeneous observations. In accordance with our algorithm, we remove these observations from the data and repeat the procedure on the remaining 333 observations. We describe the second iteration in greater detail.

Figure 4 shows the forward plot of minimum deletion residuals for these 333 observations. Clearly they are not homogeneous, but the question for this iteration is where is the end of the major group? There is a sharp dip in the plot at around $m = 120$ caused by the inclusion in $S(m)$ of a unit with high leverage. Otherwise, we first obtain a signal indicative of model failure at $m = 225$, the first point at which the calculated minimum deletion residual lies above the 99.9% envelope. However, as the plotted envelopes show, we can expect larger values of the residual as m approaches n even when there are no outliers; the value of n for this group may be somewhat larger than 225. Indeed this does seem to be the case.

The upper left-hand panel of Figure 5 shows the envelopes for $n = 225$, together with the deletion residuals up to this sample size. With these new, more curved, envelopes it is clear that the group is homogeneous up to this size. The same is true for the upper right-hand panel for $n = 235$. However, in the lower left-hand panel with $n = 245$, the observed values have already crossed the 99% envelope. For $n = 248$ the 99.9% envelope is crossed, so there is evidence of non-homogeneity. When n is one less, namely 247, there is no exceedance and we take the second component as containing 247 observations.

There are two points about this process. The minor one is that the envelopes are recalculated for each panel but the calculated values of the minimum deletion residuals are from the search plotted in Figure 4; as n increases the plots reveal more of this sequence of values. The major point is that the envelopes we have found have the stated probability, but for each m. Thus the probability of exceeding the 99.9% envelope for any m is 0.1%. However, the probability that the 99.9% envelope is exceeded at least once during a search is much greater than 0.1%. The calculations for regression are in [3]. Here the envelopes are much more like 99% overall, which is a more reasonable level at which to detect a change in structure. However, in our application, the exact significance level of this part of our analysis is not crucial.

The procedure of identification and deletion continues for another three iterations, leading to additional homogeneous populations of 247, 38 and 22 observations. Figure 6 shows the four components of the mixture estimated with this procedure and the remaining 26 observations (the '+' symbols).

The slopes of the four mixture components, from 1 to 4, are: 14.044, 12.542, 13.134 and 5.83. We recall that these values are estimates of the import price of the flows as-

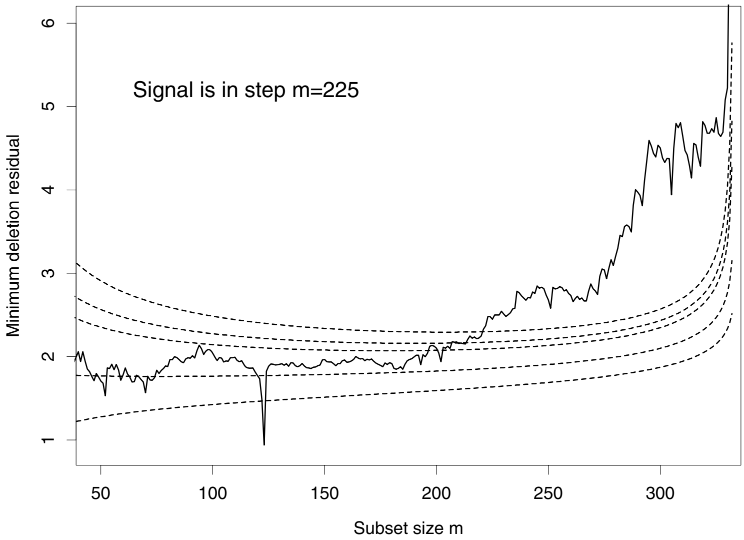


Figure 4. Fisheries data: the 333 observations in iteration 2. Forward plot of minimum deletion residuals with 1, 50, 99, 99.9 and 99.99% envelopes. There is a signal at step 225.

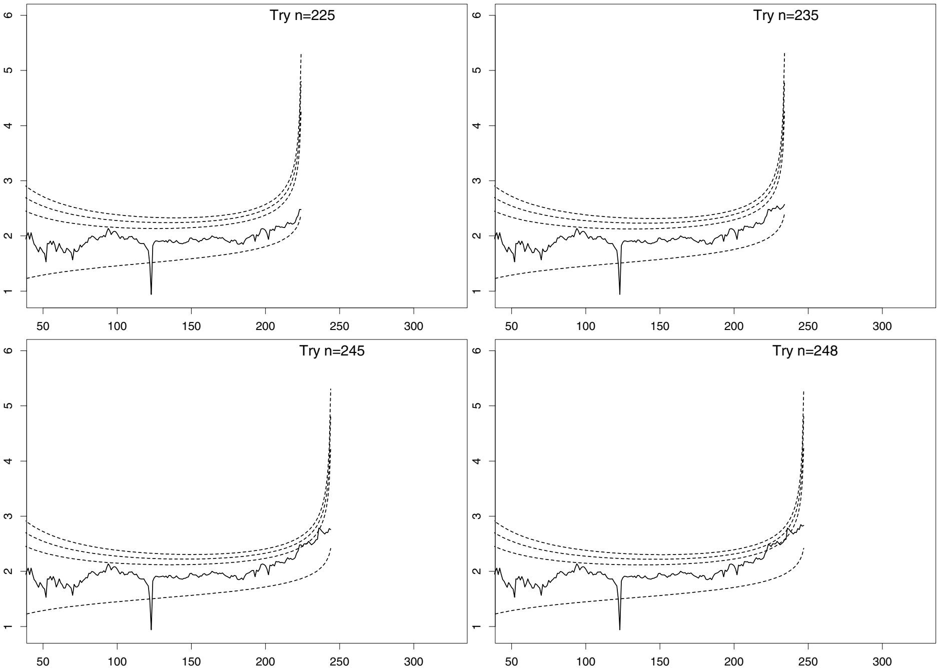


Figure 5. Fisheries data: the 333 observations in iteration 2. Forward plot of minimum deletion residuals with 1, 99, 99.9 and 99.99% envelopes for various sample sizes. Iteration 2 identifies a group of 247 observations.

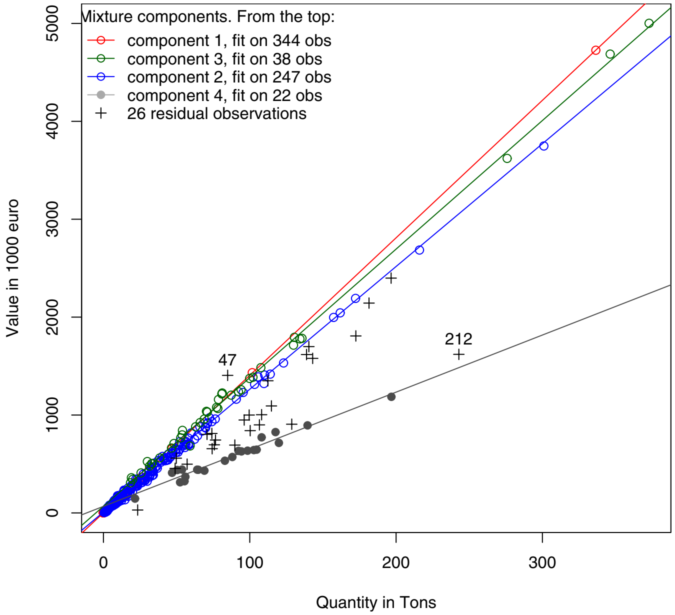


Figure 6. Fisheries data: a mixture of four linear regression lines estimated by iterating the FS on the 677 EU import flows in the dataset. The "residual" flows are identified with a '+' symbol. Those marked with their record number, 212, 47, are detected as outliers by a final FS on the residual flows.

signed by the FS to the four groups. Interestingly, we have verified in our dataset that the group fitted by component 4 (estimated import price $5.83€$) is exclusively formed by import flows to MS7 that took place in 22 consecutive months. In addition, there are no flows to MS7 in the other three groups.

Since the prices, that is the slopes of the regression lines, are the major output of our analysis, we give in Figure 7 the forward plots of the estimated slopes during the forward searches for the different iterations. That for the 344 in group 1 is amazingly stable. That for group 2 is also stable, although it does show some slight fluctuations, as well as a small jump around $m = 120$ that we noticed in Figure 3. The much smaller group 3 has a slope between groups 2 and 3. As our other analyses have shown, the slope for group 4 is markedly different. All of these slopes are sensibly constant during the search. However, that for the residual group decreases rapidly, suggesting that these 26 observations are far from homogeneous.

Information on the degree of homogeneity of the observations in the groups can be also obtained from a plot of the estimates of the squared correlation coefficient R^2 in the five iterations (for the general definition of R^2 see, for instance, [15]). This is shown in Figure 8, which also reflects the high strength of the linear regression fit in the four groups.

At this point we accordingly ran the FS also on the 26 "residual" observations. Among those entering the subset in the last steps, two are detected as outliers and cause a visible increase in the plot of deletion residuals. In fact Figure 6 shows that one of these observations, record number 47 in the original dataset, appears almost in line with the first mixture component while the other, 212, is virtually in line with the fourth component.

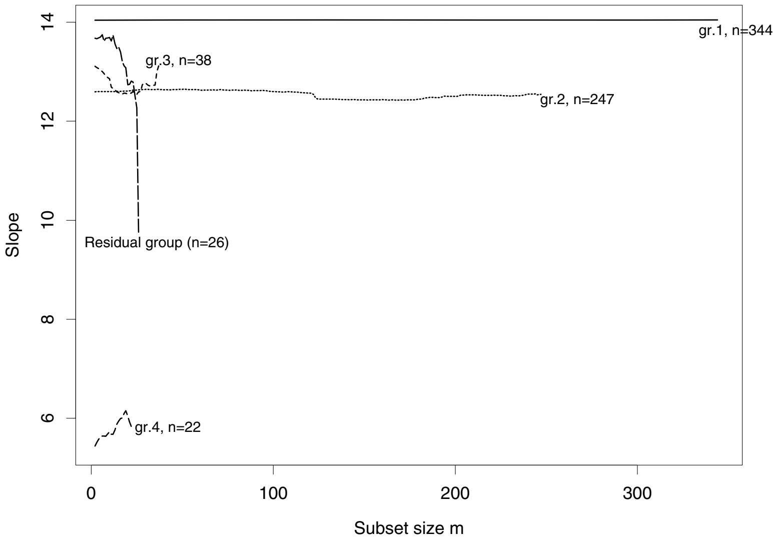


Figure 7. Fisheries data: forward plot of estimated regression coefficients in the five iterations. The first and second groups are very stable with coefficients approximately twice that for the suspicious Group 4.

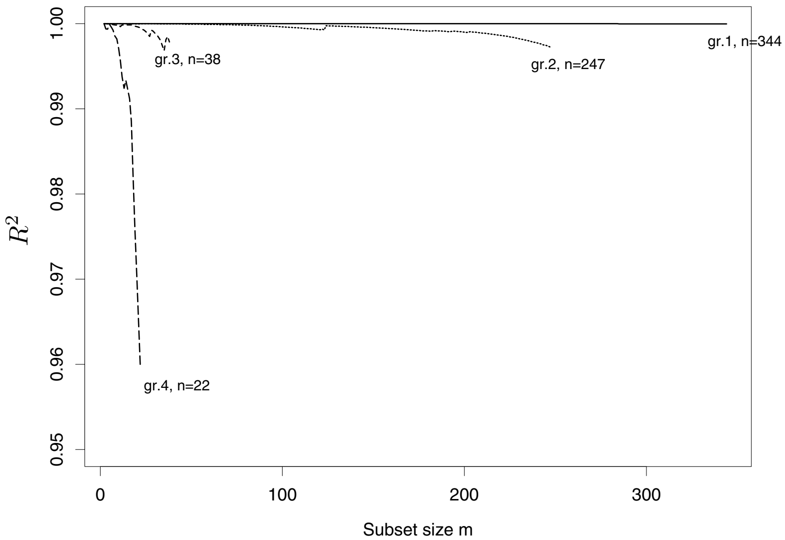


Figure 8. Fisheries data: forward plot of the estimates of the squared correlation coefficient R^2 in the five iterations. The strength of the linear regression fit in the four groups is high.

We now consider in more detail the composition of the residual observations, which are plotted in Figure 9. We represent the flows to MS7 with open circles and the flows to the other Member States with solid dots. The dashed line fits the flows to MS7 but excludes 212 since, as we have just remarked, it is very close to the fourth component. The slope of this line is 7.66. Again this group of 13 flows to MS7 took place in consecutive months. Among the other 11 residual flows, 7 refer to a single Member State, MS2.

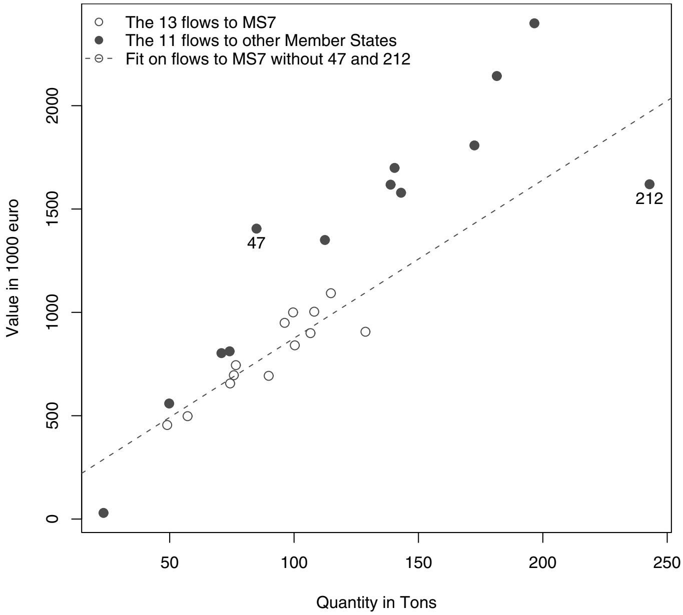


Figure 9. Fisheries data: zoom of the residual flows marked with a '+' in Figure 6. The open circles are the flows to MS7. The solid dots are the flows to other Member States. Again, the two outlying flows are labelled with their record numbers. The regression line is fitted using the flows to MS7 excluding 212, which is outlying.

4.3. Implications for Anti-fraud

The flows to MS7 have been clustered into two homogeneous groups: the first which we called component 4 and the second a fitted subset of the residual observations. Historically, the flows in the first of these two groups took place after those in the second. The estimated import prices for the two periods are considerably different: 5.83€ and 7.66€, and also considerably lower than the prices estimated for the other groups and Member States, 14.05€, 12.54€ and 13.13€. In short, in the period analysed, MS7 lowered the import price of this fishery product, up to half of the import price reported by the other Member States. In earlier analyses this type of pattern was not considered. Its operational evaluation, for example in relation to possible evasion of import duties, is the responsibility of the anti-fraud services.

5. Discussion and Directions for Further Work

The procedure of Section 3.2 indicated four clear sub-populations. A limitation of the procedure is the lack of a criterion to decide automatically about the nature of the residual flows: some of them may form separate homogeneous groups, others are very close to existing groups and could be re-assigned (e.g. flows 47, 212) and yet others may be outliers in the entire dataset (e.g. flow 355). In fact, we have used a rather pragmatic approach to the analysis of the residual observations. A confirmatory analysis invoking simultaneous searches including all established regression lines (not given here for lack of space) can help to infer the degree to which each unit belongs to each group.

The focus in this example has been on clustering linear regression models with motivation for the FS coming from the clustering of multivariate data. We would like to stress that the FS is of much wider applicability; examples, not all in [2], include applications to multiple and curvilinear regression, to nonlinear and generalized linear models and to the estimation of response transformations in regression and data transformation in multivariate analysis. Given any quantity of interest, such as a parameter estimate or a test of departures from a model, its properties can be studied using the FS. The distributional properties of the quantity can be found, often by simulation. Any significant departure from this distribution may indicate outliers, ignored structure or a systematically inadequate model, depending on the quantity being studied. In our anti-fraud application we require techniques for large numbers of observations. For the very large data sets encountered in "data-mining" we use a FS in which $s > 1$ units enter at each forward step; thus we move directly from the subset $S(m)$ to the subset $S(m + s)$.

Acknowledgements

The work of Cerioli, Riani and Atkinson was partially supported by the grants "Metodi statistici multivariati per la valutazione integrata della qualità dei servizi di pubblica utilità: efficacia-efficienza, rischio del fornitore, soddisfazione degli utenti" and "Metodologie statistiche per l'analisi di impatto e la valutazione della regolamentazione" of Ministero dell'Università e della Ricerca – PRIN 2006.

The work of Perrotta and Torti was conducted in the research action "Statistics and Information Technology for Anti-Fraud and Security" of the Joint Research Centre of the European Commission, under the institutional work-programme 2007-2013.

References

[1] A.C. Atkinson, Fast very robust methods for the detection of multiple outliers, *Journal of the American Statistical Association* **89** (1994), 1329–1339.

[2] A.C. Atkinson and M. Riani, *Robust Diagnostic Regression Analysis*, Springer, New York, 2000.

[3] A.C. Atkinson and M. Riani, Distribution theory and simulations for tests of outliers in regression, *Journal of Computational and Graphical Statistics* **15** (2006), 460–476.

[4] A.C. Atkinson and M. Riani, Exploratory tools for clustering multivariate data, *Computational Statistics and Data Analysis* (2007).

[5] A.C. Atkinson, M. Riani and A. Cerioli, *Exploring Multivariate Data with the Forward Search*, Springer, New York, 2004.

[6] A.C. Atkinson, M. Riani and A. Cerioli, Random start forward searches with envelopes for detecting clusters in multivariate data, in: S. Zani, A. Cerioli, M. Riani and M. Vichi (Eds.), *Data Analysis, Classification and the Forward Search*, Springer-Verlag, Berlin, 2006.

[7] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York.

[8] A. Cerioli, M. Riani and A.C. Atkinson, Robust classification with categorical variables, in: A. Rizzi and M. Vichi (Eds.), *COMPSTAT 2006: Proceedings in Computational Statistics, Physica-Verlag, Heidelberg*, 2006.

[9] B. Flury and H. Riedwyl, *Multivariate Statistics: A Practical Approach*, Chapman and Hall, London, 1988.

[10] C. Fraley and A.E. Raftery, Enhanced Model-Based Clustering, Density Estimation, and Discriminant Analysis Software: MCLUST, *Journal of Classification* **20** (2003), 263–286.

[11] A.S. Hadi, Identifying multiple outliers in multivariate data, *Journal of the Royal Statistical Society, Series B* **54** (1992), 761–771.

[12] R.A. Maronna, R.D. Martin and V.J. Yohai, *Robust Statistics: Theory and Methods*, Wiley, Chichester, 2006.

[13] M. Riani, A.C. Atkinson and A. Cerioli, Results in Finding an Unknown Number of Multivariate Outliers in Large Data Sets, Research Report 140, London School of Economics, Department of Statistics.

[14] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, 2006.

[15] G.A.F. Seber, *Linear Regression Analysis*, Wiley, New York, 1977.

[16] H. Späth, *Cluster Dissection and Analysis*, Ellis Horwood, Chichester, 1985.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-287

Money Laundering Detection Using Data Mining

Ekrem Duman[1,2] and Ayse Buyukkaya[3]
[2]*Dogus University, Industrial Engineering Dept.*
[3]*Intertech A.S.*

Abstract. The money made from illegal activities is a main threat to the global economy. If the overall welfare of the people is wanted to be increased the money laundering activities should be stopped. Many software packages which aim to detect suspicious transactions are developed and available commercially. These solutions are mostly rule based. That is a list of suspicious transactions is generated where their suspiciousness is determined by some rules. If these lists are too long, it would be very difficult to inspect them in full but if they are kept shorter then there is the risk of failing to inspect an actually fraudulent transaction. To overcome this drawback of the rule based solutions, we suggest a two phase anti money laundering system which provides more flexibility with the use of data mining.

Keywords. Anti money laundering, data mining.

Introduction

The black economy is one of the biggest troubles that almost every country faces. It mainly influences the economy and the well being of the citizens and this effect is relatively larger in the less developed countries. Financial outputs of these illegal economic activities can be regarded as black money. The solutions against the money laundering actions are expected to identify who are related with black money. In many countries, the financial institutions are expected to inform compliance regulation bodies about any persons or transactions that they think suspicious. To cope with this necessity, various software packages for anti money laundering (AML) have been developed and are commercially available.

The commercial solutions for AML are mostly rule-based. This means that the transactions that satisfy some pre-determined rules are assumed to be potentially suspicious. These rules are settled by the software houses and are usually customizable according to the needs of the customer bank and its country. This list of potentially suspicious transactions is then expected to be investigated by the inspectors. However, at this point a paradox appears: If the bank does not want to fail to identify any suspicious transaction then the rules should somewhat be loose in which case the list would be very long and it would be very difficult to fully inspect it (this would need

[1] Corresponding author Address: Acibadem Zeamet Sok. No. 21, 34722 Kadikoy, Istanbul, Turkey, email: eduman@dogus.edu.tr

too many inspectors) or the rules can be established tightly and a short list can be generated. This time it would be easier to fully inspect the list, but some transactions which are actually fraudulent may not appear on the list. This trade-off usually happens with these software packages.

To overcome this drawback of the existing solutions we propose a framework of an intelligent solution procedure which makes use of data mining (DM). Data mining (DM) can be defined as the art and science of finding the useful information hidden in large databases [1]. It is a powerful tool especially when the amount of data is huge. There can be two broad objectives in using data mining [2]: description and prediction. In the descriptive use, some profiling of data is made in order to describe the data (sometimes the customers) better. On the other hand, in predictive DM, the likelihood of an event to occur is determined. A wide range of classification problems like customers' propensity to buy a product, credit scoring, churn, money laundering, disease diagnosis, fraud, etc. fall into this category.

In machine learning terminology these two categories are named as supervised and unsupervised learning. In predictive DM, some known cases are required whose classes are known. Then using these samples a training set is formed where the relationships defining the class memberships are learned using a DM algorithm. In other words, the learning is supervised by the known cases. In descriptive DM, no training set is required thus, the name unsupervised learning.

In the case of money laundering, the known cases which are proved to be ML are too few to form a training set. Hence, the predictive DM algorithms are not directly applicable. In this study we devise a framework as a two stage solution procedure where both DM types are used.

In the first phase we are content with descriptive DM where the bank customers are clustered according to their bank usage behaviors. Then using the variables defining the clusters together with the customer historical variables, the outliers are detected. For this purpose descriptive data mining methods are utilized. The conditions of being an outlier are somewhat loose so that, even people/transactions which have a very small probability of being fraudulent are considered as outliers and only these outliers are passed to the second phase.

The second phase is related with using predictive data mining techniques to define which transactions should be investigated. Here, a close work with the inspectors is needed. The inspectors will be asked to go over the list generated in the first phase to determine which ones are worth to be inspected. Then, a predictive DM algorithm is used to determine the inspection list automatically. At this phase, some behavioral variables will be used as predicators.

Our solution framework is presented to a bank and some initial steps are taken towards its implementation. The first impressions of the bank authorities are quite positive.

The rest of the paper is organized as follows. In the next section we will provide some background information on money laundering and the global regulations. We will also give a brief survey on related literature. In the second section, we will describe the two phase AML solution framework that we suggest. In section three, we will summarize the paper and give the conclusions.

1. Background

Black money can be defined as the money made from a variety of criminal activities such as illegal arms sales, smuggling, drug trafficking, prostitution rings, activities of organized crime, embezzlement, insider trading, bribery and fraud [3]. The profits made from these activities usually sums up to huge amounts and according to some estimates it could comprise two per cent of the global GDP.

The black money should somehow be introduced to the legal economy. Money laundering can be defined as the processing of criminal profits through the financial system to obscure their illegal origins and make them appear legitimate [3].

Money laundering involves three stages: placement, layering and integration [4]. *Placement* is the introduction of the dirty money to the financial system. After putting the money in the financial system, the money launderer works the money through a complex series of transactions to separate it from its illegal origins. This is known as the *layering*. Finally the *integration* phase, involves reintegrating the washed or cleansed funds with formal sector economic activity.

For laundering the money a variety of different techniques can be used such as structuring, use of front companies, mis-invoicing, use of shell companies, wire transfers, mirror image trading etc. The interested reader can refer to the paper of Buchanan [3] for the details. Also the development of the internet has eased the money laundering process [4].

There are some global and national regulations for fighting the money laundering. The financial institutions (banks) are required to report any suspicious transactions that their customers make to local authorities. There are also some international regulations that coordinate these institutions and put the rules for international cooperation [5]. The Bank Secrecy Act (BSA) was the very first regulation declared in the USA in 1970. The Financial Action Task Force (FATF) is one of the most important regulations. It was formed by the G7 countries in 1989 with the primary purpose of examining measures to combat money laundering. The details of these and other global regulations can be found in the study of Buchanan [3].

Anti money laundering (AML) systems process the financial transactions using some rules or statistical-artificial intelligence tools to flag non-obvious relationships between pieces of data in large input data sets, or to look for unusual patterns of behavior, such as sudden cash movements in a previously dormant account [6, 7, 8]. There are also some AML systems which use artificial intelligence and intelligent agents [9].

Canhoto and Backhouse [10] make a discussion on why the AML systems can not achieve high success rates: First, money laundering (ML) does not correspond to any one particular behavior. Second, ML may involve a variety of actors, ranging from individual criminals who themselves launder the money to highly sophisticated organized crime groups that have their own ‘financial director’ [6]. Third, the form that ML takes is continuously evolving [7]. Fourth, the few elements of information available are held by different institutions that do not exchange information easily owing to legal, strategic and operational reasons.

As most money laundering processes involve more than one financial institution and since the anti money laundering solutions are applied in individual banks then it will not be possible to track the transactions in other banks. If this was possible then possibly more powerful solutions could have been developed. But even if we have the records of only one bank it will still be possible to find some ways of identifying suspicious transactions/customers. In this study, we consider this case of having access to the records of only one bank.

2. Suggested AML Framework

To overcome the basic deficiency of the available commercial AML systems (that is, either the list is too long to inspect or short but having the risk of failing to include actually fraudulent cases) and to utilize the powerful DM techniques, we suggest a two phase intelligent AML framework:

1. Descriptive data analysis phase
2. Predictive data analysis phase

As the names imply, descriptive and predictive DM techniques are used for the first and second phases, respectively. In the first phase, anomaly detection algorithm is used to narrow down the search list. That is instead of investigating the full list of transactions/customers, the ones showing usual behavior are eliminated and the search space is reduced. In the second phase, predictive DM algorithms are used. To cope with the difficulty of establishing a training set, we suggest showing the list obtained from anomaly detection procedure to inspectors and ask them to flag the ones they find worth to inspect and use these as the training set. Also, the ones found suspicious after the inspections can be taken as another training set to learn from. The details of these two phases are given in the subsections below.

2.1. Descriptive Data Analysis

The search for suspiciousness can be made on transaction level, account level or customer level. We prefer the investigations be carried on customer level and give the discussion accordingly. But, the same ideas are also valid for transaction and account levels. Thus, in our terminology, the term "customer level" denotes all three levels without loss of generality.

We can start developing our AML solution by asking questions on what is suspicious. What kind of customer behavior should be defined as suspicious? For example, is a customer who made more than 50 wire transfers last month suspicious or not? Actually, there is not a direct answer to this question. It all depends on what kind of a customer that is. If, for example, he owns a small business or something like that, this can be a normal behavior. On the other hand, even he is an individual but his way of using the bank is so that he is routinely making many wire transfers, we may regard him as usual. So, two things turn out to be important to judge whether a particular behavior is suspicious or not:

1. The type (peer group) of the customer
2. The historical behavior of the customer

Thus, any solution attempting to find out what is suspicious should take both of these factors into account. Actually, most of the commercial AML solutions take the first factor into account (the study of Tianqing [11] suggests considering both factors but in a sequential manner). The peer groups are determined as rule based segments or by clustering algorithms. Whatever the method is, determination of the variables to be used in forming the peer groups is very important. The peer groups should distinguish between the dynamics of different money laundering methodologies. In our opinion, use of clustering should be preferred where variables such as the following should primarily be selected:

- Number of accounts opened
- Number of accounts closed
- Number of accounts opened or closed through internet
- Number of debit and credit transactions
- Amount and number of transactions through internet
- Number of wire transfers, incoming and outgoing
- Total and average amount of wire transfers

Then, to cope for the second factor, we suggest using the deviation of the customer's last behavior (say, last month) from his average behavior in the near past (say, last six months) using the equation:

$$\text{Deviation} = (x-\mu)\ /\sigma \qquad (1)$$

where,
x = the value in the last month
μ = average value in the last six months
σ = standard deviation of the values in the last six months

Then using these normal and deviation variables (say, the number of variables in total is *n*) natural clusters are determined (say, the number of clusters obtained is *k*). Some customers will be close to the cluster centers while some will be apart. The distance of the customer from the cluster center (Customer Distance = CD) can be defined as;

$$\text{CD} = \text{sum of the variable distance (VD) values} \qquad (2)$$

where, the variable distance for variable *k* ($\text{VD}_{\text{k)}}$ is equal to the absolute difference of the customer's variable *k* value from the value at the cluster center.

Then, the customer anomaly index (CAI) can be defined as the customer distance value divided by the average customer distance value in the cluster. That is,

$$\text{CAI} = \text{CD}\ /\ (\text{average CD in the cluster}) \qquad (3)$$

The customers having a CAI close to one, can be regarded as normal, or usual. The ones having a high CAI are outliers; if we are to suspect some customers, these can be the candidates. They showed an usual behavior as compared to their peer group or their past. All these can be legitimate of course but probabilistically we can say that the fraudsters will be among them with a higher probability.

After determining the customers having a high anomaly index, it is also desirable to know the source of this high index. In this regard, we can determine variable *k*'s contribution measure to the CAI (VCM_k) using the formula,

$$\text{VCMk} = \text{VDk} / \text{CD} \tag{4}$$

Then, pointing out the variables with a high contribution could be of great help to the inspectors. An example list from this phase is given in table one. We thank to our commercial partner for letting us use this table here.

Table 1. Anomaly list of phase one.

CAI	Primary Variable	VCM	Secondary Variable	VCM
5.00	INT-LOGON-NUM	0.62	CA-NUM-ACCT-TL	0.22
4.92	CHEQUE-NUM-TRX-DEV	0.34	BRANCH-NUM-TRX-DEV	0.29
4.59	EFT-IN-TRX-NUM-DEV	0.98	CC-NUM-TRX	0.01

This is only a small part of a long list. In the first column we see the customer anomaly index of the customers sorted in non-increasing order (in the software we used, the max index value is normalized to 5.00). Then we see the two variables which contribute most to the anomaly status of the customer. For example, for the first customer, the number of times the customer is logged on to internet banking is responsible for the 62 per cent whereas the number of current accounts in TRL explains 22 per cent of his anomaly behavior.

The list of customers obtained by such an anomaly detection algorithm can be used as a standalone tool where only the customers having anomaly indexes larger than a high threshold can be inspected. Otherwise, if the threshold is lowered, the number of customers to be inspected could become much more than that can be inspected. Or, the list obtained by a lower threshold can be transferred to the second phase as the starting list.

2.2. Predictive Data Analysis

As indicated above, the difficulty in using the predictive DM is that, the cases which are known to be money laundering are too few. This is far beyond being sufficient to form a training set. However, during our discussion with the inspectors on the result of anomaly detection, we noticed that they can quickly judge on whether a customer having a high anomaly index should be inspected or not. So, we developed the idea of using the predictive DM to predict which customers the inspectors will want to inspect. They can go over the full anomaly list once to indicate the ones requiring inspection and this can form the training set to determine how the anomaly detection results produced later can be treated. The model trained could be applied to the list generated in phase one to determine a subset of it for which there is a higher probability that the inspectors might want to inspect.

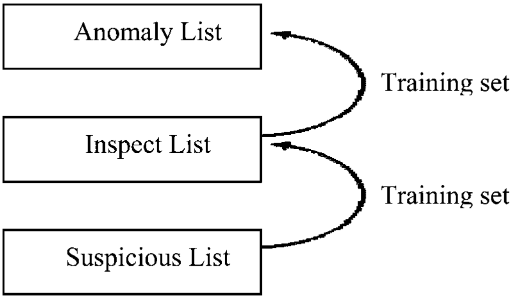


Figure 1. The training sets for predictive DM.

Also, after making the inspection, the inspectors will find some customers suspicious that are to be informed to the local regulatory office. This result can be taken as the training set and predictive DM can once again be applied to predict which customers are actually suspicious (see figure 1). This can further narrow down the list of customers to be inspected. This way the workload of the inspectors could be minimized with a small risk (hopefully) of failing to inspect the customers who are actually fraudulent.

3. Summary and Conclusions

In this study, the definition and the types of money laundering are given. Then, a short information regarding the solutions for anti money laundering is given. It is noted that, the available solutions are mostly rule based and that they have a drawback of either producing too long lists which is difficult to inspect or when they produce shorter lists, they have the risk of failing to include actually fraudulent cases.

To overcome this drawback of the existing solutions we suggested a new AML framework which makes use of data mining. The framework consists of two phases: descriptive data mining and predictive data mining.

When we shared these ideas with the managers of a major bank in Turkey we got quite good impressions towards the implementation of such a solution.

References

[1] Duman, E.: Comparison of Decision Tree Algorithms in Identifying Bank Customers Who are Likely to Buy Credit Cards. Seventh International Baltic Conference on Databases and Information Systems, Proceedings of the Workshop on Information technologies for Business, Kaunas, Lithuania, pp.71-81. July 3-6, (2006).

[2] Dunham, M.H.: Data Mining: Introductory and Advanced Topics. Prentice Hall, New Jersey (2003)

[3] Buchanan, B.: Money laundering – a Global Obstacle. Research in International Business and Finance 18, 115-127 (2004)

[4] Philippsohn, S.: Money Laundering on the Internet. Computers & Security 20, 485-490 (2001)

[5] Webb, L.: A Survey of Money Laundering Reporting Officers and Their Attitudes Towards Money Laundering Regulations. J. Money Laundering Control 7 (4), 367-375 (2004)

[6] Bell, R.E.: An Introductory Who's Who for Money Laundering Investigators. J. Money Laundering Control 5 (4), 287-295 (2002)

[7] Angelli I.O., Demetis, D.S.: Systems Thinking About Anti Money Laundering: Considering the Greek Case. J. Money Laundering Control 8 (3), 271-284 (2005)

[8] Complinet,: Anti-Money Laundering Systems Review. Compliance and Technology Review 1 (4) 6-9 (2005).

[9] Kingdon, J.: AI Fights Money Laundering. IEEE Intelligent Systems, 87-89 (May/June 2004)

[10] Canhoto, A.I., Backhouse, J.: Profiling Under Conditions of Ambiguity – an Application in the Financial Services Industry. J. Retailing and Consumer Services 14, 408-419 (2007)

[11] Tianqing, Z.: An Outlier Detection Model on Cross Datasets Comparison for Financial Surveillance. Proceedings of the 2006 IEEE Asia-Pasific Conference on Services Computing (APSCC'06), (2006)

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-295

Text Mining from the Web for Medical Intelligence

Ralf STEINBERGER [a,1], Flavio FUART [a], Erik van der GOOT [a], Clive BEST [a], Peter von ETTER [b], and Roman YANGARBER [b]

[a] *European Commission—Joint Research Centre, Ispra, Italy*
[b] *Department of Computer Science, University of Helsinki, Finland*

Abstract. Global medical and epidemic surveillance is an essential function of Public Health agencies, whose mandate is to protect the public from major health threats. To perform this function effectively one requires timely and accurate medical information from a wide range of sources. In this work we present a freely accessible system designed to monitor disease epidemics by analysing textual reports, mostly in the form of news gathered from the Web. The system rests on two major components—MedISys, based on Information Retrieval technology, and PULS, an Information Extraction system.

Keywords. Information Retrieval, Information Extraction, multilinguality, medical intelligence, multi-document information aggregation

Introduction

Professionals in many fields need to sieve through large volumes of information from multiple sources on a daily basis. Most European Union (EU) countries have a national organisation that continuously monitors the media for new threats to the Public Health in their country, and for the latest events involving health threats. These threats range from outbreaks of communicable diseases and terrorism cases such as the deliberate release of biological or chemical agents, to chemical or nuclear incidents. Typically, the staff of these organisations search their national and local newspapers and/or purchase electronic news from commercial providers such as Factiva or Lexis-Nexis. Until recently, relevant news articles were cut out from printed press, and compiled into an in-house newsletter, which was then discussed among specialists who had to decide on the appropriate action. As more news sources became available on-line, it became easier to find relevant articles and to compile and manage them electronically. At the same time, the number of available sources rose, and—due to increased travel and the consequent importing of infectious diseases—it became necessary to monitor the news of neighbouring countries and major travel destinations.

These and similar professional communities can benefit from text analysis software that identifies potentially relevant news items, and thereby increases the speed and efficiency of their work, which is otherwise slow and repetitive. The search func-

[1] E-mail address format: firstname.lastname@jrc.it or @cs.helsinki.fi

tions of news aggregators, such as Factiva or Google News, allow users to formulate Boolean search word combinations that filter items from large collections. The European Commission's *Medical Information System*, MedISys, in addition to providing keyword-based filtering, aggregates statistics about query matches, which enables it to provide early-warning signals by spotting sudden increases in media reports about any Public Health-related issue and alerting the interested user groups.

While this functionality is in itself helpful for the communities in question, deeper text analysis can provide further advantages, beyond those provided by classic Information Retrieval (IR) and alerting. In this chapter, we describe the IR and early-warning functionality of MedISys, and how it inter-operates with the information extraction (IE) system PULS, which analyses the documents identified by MedISys, retrieves from them *events*, or structured facts about outbreaks of communicable disease, aggregates the events into a database, and highlights the extracted information in the text. Our evaluation confirms that event extraction helps to narrow down the selection of relevant articles found in the IR step (improving precision), while on the other hand missing a small number of relevant articles (lowering recall).

The next section presents related work; sections 2 and 3 describe MedISys and PULS. Section 4 describes the mechanisms currently used for aggregating information from multiple reports. Section 5 shows how the two systems are combined into one Web application. Section 6 presents evaluation results. The final section draws conclusions and points to future work.

1. Related work

Information retrieval and information extraction have been thoroughly researched over the recent decades, with abundant literature on both topics. Typically they are studied separately, with results reported at different fora, and they are considered different problem areas, since they employ quite different methods. Conceptually, IR and IE both serve a user's information need, though they do so at different levels. It is understood that in real-world settings, IR and IE may be expected to interact, for example, in a pipeline fashion. The possibilities of tighter interaction largely remain yet to be researched.

Gaizauskas and Robertson ([1]) investigated the costs and benefits of combining IR and IE, by first applying a search engine (Excite) and its summary extraction tool, and then extracting MUC-6 "management succession" events, ([2]). The MUC-6 task is to track changes in corporate management: to find the manager's post, the company, the current manager's name, the reason why the post becomes vacant, and other relevant information about the management switch. The authors conclude that using IR as a filter before IE clearly results in a speed gain (since applying IE to *all* documents returned by the search engine would not have been possible), while the cost was a loss of 7% of the relevant documents. In further experiments by Robertson and Gaizauskas ([3], precision rose by 32% up to 100%, though at the cost of losing 65% of the retrieved relevant documents.

From an application point of view, to our knowledge, there are two other systems that attempt to gather information about infectious disease outbreaks from automatically collected news articles: Global Health Monitor [4] and HealthMap [5]. The systems provide map interfaces for visualising the events found.

Global Health Monitor follows about 1500 RSS news feeds hourly, and matches words in the new articles against a taxonomy of about 4300 named entities, i.e., 50 names of infectious diseases, 243 country names, and 4000 province or city names. For place names, the taxonomy contains latitude-longitude information. The 50 disease names are organised into an ontology with the properties relating to synonyms, symptoms, associated syndromes and hosts. The Global Health Monitor processing consists of four steps: (1) relevance decision (using Naïve Bayes classification); (2) named entity recognition (disease, location, person and organisation, using Support Vector Machine classification); (3) filtering of articles containing both disease and location names in the first half of the text. Additionally, only those disease-location pairs are retained that are frequently found in a separate *reference corpus*. Step (4) then visualises the successful matches on a map. Due to the rigorous filtering in steps (1) to (3), the system retains information on 25-30 locations and on about 40 infectious diseases a day. The system currently provides text analysis for English, though the underlying ontology includes terms in several other languages, including Japanese, Thai, and Vietnamese.

HealthMap monitors articles from the Google News aggregator and emails from the collaborative information-sharing portal ProMED-Mail,[2] and extracts information about infectious diseases and locations. After a *manual* moderation process, the results are stored in a database and visually presented on a map. Diseases and locations are identified in the text if words in the text exactly match the entities in the HealthMap taxonomy, which contains about 2300 location and 1100 disease names. Some disambiguation heuristics are applied to reduce redundancy (e.g., if the words "diarrhoea" and "shigellosis" are found, only the more specific entity "shigellosis" will be retained). HealthMap identifies between 20 and 30 disease outbreaks per day. More recent articles and those disease-location combinations reported in multiple news items and from different sources are highlighted on the map. The system developers point out the importance of using more news feeds, as their current results are focused toward the North-American continent. HealthMap currently displays articles in English, French, Spanish and Russian, ([5] describes English processing only).

The system presented in this chapter covers a large number of sources, a wide range of languages (currently, 43) and health-related topics (epidemic, nuclear, chemical and radiological incidents, bio-terrorism, etc.) Its special emphasis is on aggregation of the information collected from multiple sources and languages, and across time, and using the aggregation to provide additional functionality—for example, urgent warnings about unexpected spikes in levels of activity in a given area.

2. Information Retrieval in *MedISys*

The *Medical Information System*, MedISys, automatically gathers reports concerning Public Health in various languages from many Internet sources world-wide, classifies them according to hundreds of categories, detects trends across categories and languages, and notifies users. MedISys provides access at three levels: (1) free public access, (2) restricted access for Public Health professionals outside the European Commission (EC), and (3) full access inside the EC. The public MedISys site[3] presents a quantitative sum-

[2] *http://www.promedmail.org*

[3] *http://medusa.jrc.it/*

mary of latest reports on a variety of diseases and disease types (e.g., respiratory infections), on bio-terrorism-related issues, toxins, bacteria (e.g., anthrax), viral hemorrhagic fevers (e.g., Ebola), viruses, medicines, water contaminations, animal diseases, Public Health organisations, and more. The restricted access site for non-Commission users offers more subject categories (e.g., covering nuclear and chemical contamination) and allows users to subscribe to daily, automatically-generated summary reports on various themes. The most complete functionality and coverage is at the Commission-internal site, which additionally monitors a number of copyright-protected news sources, such as Lexis-Nexis and about twenty news-wires.

MedISys aims to save users time, give them access to more news reports in more languages, and issue automatic alerts. An important feature of MedISys is that early-warning statistics are calculated considering the information aggregated from the news articles across all languages. MedISys is thus able to alert users of relevant events that may not yet be in the news of their country or language.

The development of MedISys was initiated by the European Commission's (EC) Directorate General *Health and Consumer Protection* (DG SANCO) for the purpose of supporting national and international Public Health institutions in their work on monitoring health-related issues of public concern, such as outbreaks of communicable diseases, bio-terrorism, and large-scale chemical incidents.[4] The following sections cover the functionality of MedISys in more detail.

2.1. Collection and standardisation of Web documents

MedISys currently monitors an average of 50,000 news articles per day from about 1400 news portals around the world in 43 languages, from commercial news providers including 20 news agencies, Lexis-Nexis, and from about 150 specialised Public Health sites. The monitored sources were selected strategically with the aim of covering all major European news portals, plus key news sites from around the world, in order to achieve wide geographical coverage. Individual users can request the inclusion of additional news sources, such as local newspapers of their country, but these user-specific sources are normally processed separately in order to guarantee the balance of news sources and their types across languages.

Where available, MedISys collects RSS feeds. RSS (”Really Simple Syndication”) is an XML format with standardised tags used widely for the dissemination of news and other documents. For other sources, scraper software looks for links on pre-defined Web pages, typically those pages that list the most recently published articles. The scraper automatically generates an RSS feed from these pages by means of specialised transformations. These transformations site-specific; they are currently produced and maintained manually, one separate transformation for each news site.

The grabber decides which of the articles in the RSS feed are new, by comparing the titles from earlier requests, and downloads the new articles. Since news pages contain

[4] MedISys users include supra-national organisations, such as the *European Commission*, *the World Health Organisation* (WHO) and the *European Centre for Disease Control* (ECDC), as well as national authorities, including the French *Institut de Veille Sanitaire* (INVS), the Spanish *Instituto de Salud Carlos III*, the Canadian *Global Public Health Intelligence Network* (GPHIN), the US *CDC*. MedISys is part of the *Europe Media Monitor* (EMM) product family, [6], developed at the EC's *Joint Research Centre* (JRC), which also includes the live news aggregation system *NewsBrief*, the news summary and analysis system *NewsExplorer* [7] and the exploratory tool set *EMM-Labs*. See *http://emm.jrc.it/overview.html* for an overview of EMM applications.

not only the news article proper, but also a great deal of irrelevant information, the main news article is extracted from each web page using a (patent-pending) text extraction process. During this process the documents are transformed into Unicode. The result is a standardised document format that allows common processing of all texts. Information about the document's language, source country, download time and source site are preserved as meta-data. The grabber software also checks whether the new text has a unique signature for this particular source. This avoids propagation of duplicate texts through the system.

2.2. Filtering and classification of the documents

MedISys allows the selection of articles about any subject, via Boolean combinations of search words or lists of search words, with positive or negative weights, and the setting of an acceptance threshold. The user may require that search words occur within a certain proximity (number of words), and may use wild cards. Using wildcards is crucial when dealing with highly-inflected languages. Each such subject definition is called an *alert*. Alerts are multilingual, which means that search word combinations mix languages. In addition to the generic alerts pre-defined by the JRC's team of developers, the specialist users may create their own subject-specific alert definitions. Users are responsible for the accuracy and completeness of their own alerts. A dedicated algorithm was developed at the JRC that allows the system to scan incoming articles for thousands of alert definitions in real time. Information about the alerts found in each article is added to the RSS file.

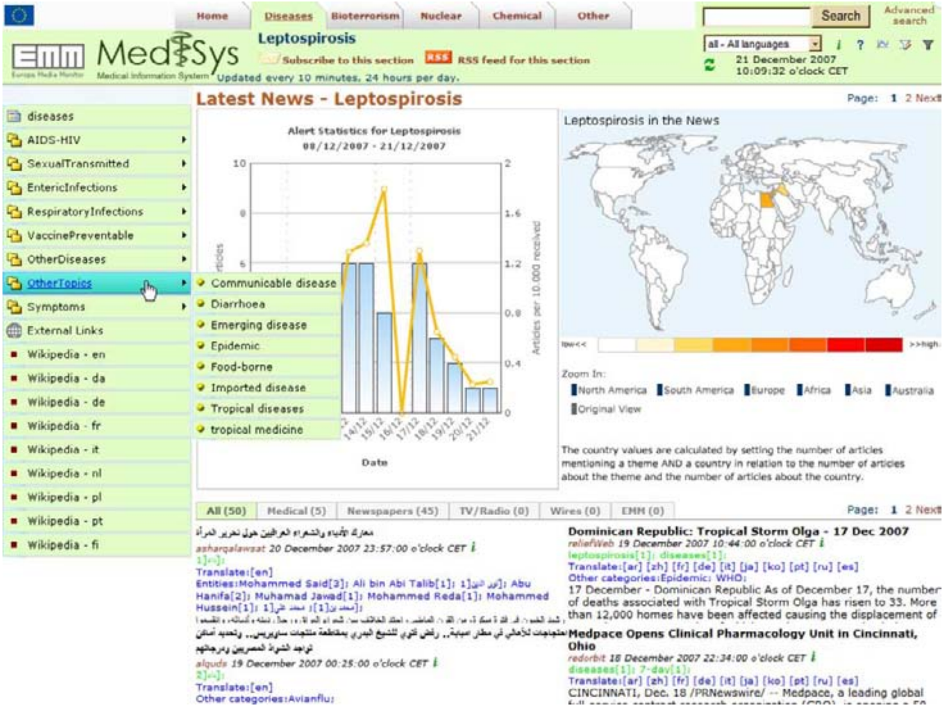
There are about 200 alert definitions for Public Health-related subjects in MedISys. The alerts are organised into a hierarchy of classes, such as Communicable Diseases, Symptoms, Medicines, Organisations, Bio-terrorism, Tobacco, Environmental & Food, Radiological & Nuclear, Chemical, etc., each containing finer sub-groups. On average, 3–4% of the 50,000 news items gathered daily satisfy at least one MedISys alert.

In addition to the subject alerts, there is one alert for each country of the world, including the name of the country and a major city. More fine-grained geo-coding and disambiguation are carried out downstream in the EMM NewsExplorer application, see [8]. Figure 1 shows the page on *Leptospirosis*, which is a specific entry of the group *Enteric Infections* in the main section *Diseases*.

2.3. Detection of early-warning trends across languages and news sources

The alert definitions in MedISys are multilingual, so that the mention of a disease or symptom can be identified in multiple languages. MedISys keeps a running count of all disease alerts for each country, i.e., it maintains a count of all documents mentioning a given disease *and* country, over a time window of two weeks. An alerting function detects a sudden increase in the number of reports for a given category and country, by comparing the statistics for the last 24 hours with the two-week rolling average. The more articles there are for a given category-country combination compared to the expected number of articles (i.e., the two-week average), the higher the alert level. Figure 2 shows a MedISys graph with the combinations having the highest alert level at a given time. The colour codes red (leftmost bars), yellow (middle bars) and blue (rightmost bars) indicate the alert levels high, medium and low.

The alert levels are calculated assuming a normal distribution of articles per category over time. Alert levels are high (or medium), if the number of articles found is at



The country values are calculated by setting the number of articles mentioning a theme AND a country in relation to the number of articles about the theme and the number of articles about the country.

معركة الأنبياء والشعراء العراقيين حول تحرير المرأة
asharqalawsat 20 December 2007 23:57:00 o'clock CET
1]...]:
Translate:[en]
Entities:Mohammed Said[3]; Ali bin Abi Talib[1]; 1]أبو النيل]; Abu Hanifa[2]; Muhamad Jawad[1]; Mohammed Reda[1]; Mohammed Hussein[1]; 1]محمد علي ;[1]محمد بن];
[illegible]
احتجاجات للأهالي في مطار عبابة.. رفض قوي للشيخ البدري بمقاطعة منتجات سابريس.. وتحديد أماكن تواجد الشرطة المصريين ورجالهم
alquds 19 December 2007 00:25:00 o'clock CET
2]...]:
Translate:[en]
Other categories:Avianflu;

Dominican Republic: Tropical Storm Olga - 17 Dec 2007
reliefWeb 19 December 2007 10:44:00 o'clock CET
leptospirosis[1]; diseases[1];
Translate:[ar] [zh] [fr] [de] [it] [ja] [ko] [pt] [ru] [es]
Other categories:Epidemic; WHO;
17 December - Dominican Republic As of December 17, the number of deaths associated with Tropical Storm Olga has risen to 33. More than 12,000 homes have been affected causing the displacement of

Medpace Opens Clinical Pharmacology Unit in Cincinnati, Ohio
redorbit 18 December 2007 22:34:00 o'clock CET
diseases[1]; 7-day[1];
Translate:[ar] [zh] [fr] [de] [it] [ja] [ko] [pt] [ru] [es]
CINCINNATI, Dec. 18 /PRNewswire/ -- Medpace, a leading global

Figure 1. A page on *Leptospirosis* on the restricted MedISys site, with part of the category hierarchy exposed on the left. The bar chart in the middle column shows the absolute and relative number of articles falling into this category. The countries mentioned in articles about leptospirosis are highlighted on the map on the right (mostly Egypt and Iraq). The most recently retrieved mixed-language articles falling into this category are listed in the lower section.

least three times (or twice) the standard deviation. As the total number of articles varies throughout the week (fewer articles on Sunday and Monday), a correction factor is applied to normalise the expected frequencies according to the day of the week. The thin line in the bar chart in Figure 1 shows the relative number of articles for a given alert compared to the total number of articles that day.

2.4. Distribution of the extracted information to the MedISys users

The Web interface of MedISys can be used to view the latest trends and to access articles about diseases and countries. For each page, RSS feeds are available for integration of the results into downstream user applications. Users can also opt to receive instant email reports, or daily summaries regarding a pre-selected disease or country, for their own choice of languages. Users can subscribe to summary reports containing information on groups of alerts (e.g., *Avian Diseases*, including *avian flu*, *duck virus* and others). Registered users can also obtain access to the JRC's *Rapid News Service*, RNS, which allows them to filter news from selected sources or countries, and which provides functionality to quickly edit and publish newsletters and to distribute them via email or to mobile phones (SMS). MedISys displays the title and the first few words of each article, plus a link to the URL where the full news text was originally found.

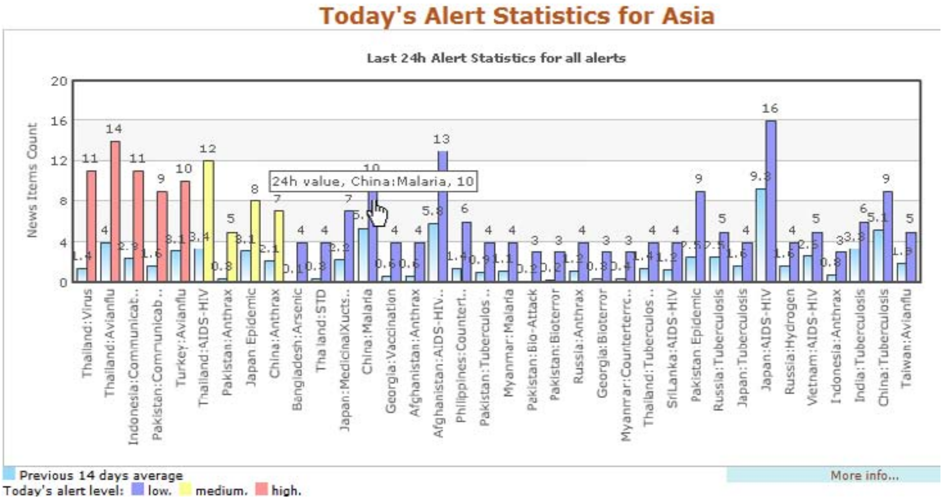


Figure 2. Alert statistics showing which category-country combinations are most active at the moment, compared to the average number of articles for the same combination. For each combination, the lower left bar indicates the expected number of articles, while the higher right bar shows the number found in the last 24 hours.

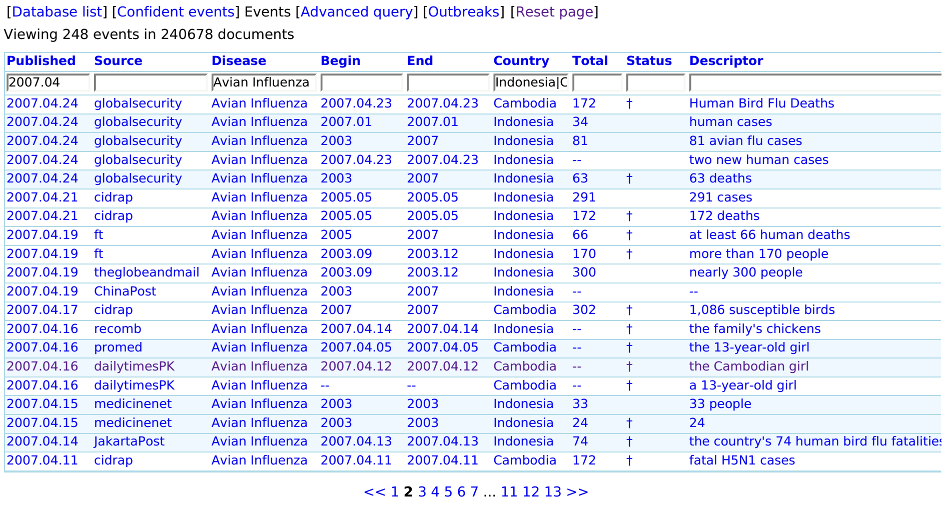
2.5. Implementation details and performance

MedISys currently processes about 50,000 media reports per day in real time. MedISys and EMM [6] are implemented in Java on Microsoft servers. MedISys shares tasks and machines with EMM, but running standalone, it would require three servers (dual processor multi-core with 4 GB of memory each). The system is scalable and could cope with processing many more feeds and news articles, for example, by processing the different languages distributed over several machines. The processes are fast and light-weight so that news monitoring and alerting happen in real-time. The slowest part of the process is downloading the articles from the Web (response time at the source). Computationally, the heaviest process currently is the real-time clustering of news articles (performed every ten minutes). The categorisation and article filtering system (see Section 2.2) matches about 30,000 patterns (multi-word terms and their combinations) against the incoming articles in a few hundred milliseconds for an average article, to categorise the articles according to about 750 categories.

3. Extraction of epidemic events in PULS

MedISys has proven to be useful and effective for finding and categorising documents from a large number of Web sources. To make the retrieved information even more useful for the end-user, it is natural to consider methodologies for deeper analysis of the texts, in particular, information extraction (IE) technology. After MedISys identifies documents where the alerts fire, IE can deliver more detailed information about the specific incidents of the diseases reported in those documents.

IE helps to boost precision, since keyword-based queries may trigger on documents which are off-topic but happen to mention the *alerts* in unrelated contexts. Pattern match-



[Database list] [Confident events] Events [Advanced query] [Outbreaks] [Reset page]

Viewing 248 events in 240678 documents

Published	Source	Disease	Begin	End	Country	Total	Status	Descriptor
2007.04		Avian Influenza			Indonesia|C			
2007.04.24	globalsecurity	Avian Influenza	2007.04.23	2007.04.23	Cambodia	172	†	Human Bird Flu Deaths
2007.04.24	globalsecurity	Avian Influenza	2007.01	2007.01	Indonesia	34		human cases
2007.04.24	globalsecurity	Avian Influenza	2003	2007	Indonesia	81		81 avian flu cases
2007.04.24	globalsecurity	Avian Influenza	2007.04.23	2007.04.23	Indonesia	--		two new human cases
2007.04.24	globalsecurity	Avian Influenza	2003	2007	Indonesia	63	†	63 deaths
2007.04.21	cidrap	Avian Influenza	2005.05	2005.05	Indonesia	291		291 cases
2007.04.21	cidrap	Avian Influenza	2005.05	2005.05	Indonesia	172	†	172 deaths
2007.04.19	ft	Avian Influenza	2005	2007	Indonesia	66	†	at least 66 human deaths
2007.04.19	ft	Avian Influenza	2003.09	2003.12	Indonesia	170	†	more than 170 people
2007.04.19	theglobeandmail	Avian Influenza	2003.09	2003.12	Indonesia	300		nearly 300 people
2007.04.19	ChinaPost	Avian Influenza	2003	2007	Indonesia	--		--
2007.04.17	cidrap	Avian Influenza	2007	2007	Cambodia	302	†	1,086 susceptible birds
2007.04.16	recomb	Avian Influenza	2007.04.14	2007.04.14	Indonesia	--	†	the family's chickens
2007.04.16	promed	Avian Influenza	2007.04.05	2007.04.05	Cambodia	--	†	the 13-year-old girl
2007.04.16	dailytimesPK	Avian Influenza	2007.04.12	2007.04.12	Cambodia	--	†	the Cambodian girl
2007.04.16	dailytimesPK	Avian Influenza	--	--	Cambodia	--	†	a 13-year-old girl
2007.04.15	medicinenet	Avian Influenza	2003	2003	Indonesia	33		33 people
2007.04.15	medicinenet	Avian Influenza	2003	2003	Indonesia	24	†	24
2007.04.14	JakartaPost	Avian Influenza	2007.04.13	2007.04.13	Indonesia	74	†	the country's 74 human bird flu fatalitie
2007.04.11	cidrap	Avian Influenza	2007.04.11	2007.04.11	Cambodia	172	†	fatal H5N1 cases

<< 1 **2** 3 4 5 6 7 ... 11 12 13 >>

Figure 3. A table view of the extracted incidents.

ing in IE provides the mechanism that assure that the keywords appear in relevant contexts only. This is of value to users who are interested in *specific* scenarios involving diseases—outbreaks and epidemics, vaccination campaigns, etc.—as opposed to users who wish to monitor documents that mention the diseases in a broader context.

PULS, the *Pattern-based Understanding and Learning System*, is developed at the University of Helsinki to extract factual information from plain text. PULS has been adapted to analyse texts in the epidemiological domain, for processing documents that trigger MedISys alerts.[5] Earlier, PULS's medical event detection had been applied to two sources dedicated to epidemiological reports—ProMED-Mail and WHO epidemic and pandemic alert and response.[6] We next describe the processing of epidemics-related texts in PULS.

3.1. Event extraction in the medical domain

For each document, the IE system extracts a set of *incidents* reported in the text. An incident is a structured representation of an event[7] involving some communicable disease, described in the text in natural language. An incident consists of a set of attributes: the location and country of the incident, the name of the disease, the date of the incident, and descriptive information about the victims—their type (people, animals, etc.), their number, whether they survived, etc. The incident may cover a single occurrence "80 chickens died on the farm on Wednesday," or larger time interval, as in "Two people in the region have contracted the disease since the beginning of the year." Text may also contain 'periodic' incidents: "according to authorities, 330 people die of malaria in

[5] *http://doremi.cs.helsinki.fi/jrc*

[6] *http://www.who.int/csr/don/en/*

[7] The term *event* is used differently in the medical and the computational communities. To the medics, an event denotes the *entire course* of an epidemic episode, from its inception to completion. In the computational literature on IE, event denotes a single, atomic "factoid", that may be isolated, or may belong to a group of factoids that together describe the entire course of an epidemic. In the rest of this chapter, the computational reading is intended.

Uganda daily” (these are not currently handled by the system). The system also identifies events in which the disease is *unknown*, or undiagnosed, which are especially important for surveillance. For example, the sentence:

The deadly Ebola outbreak has so far killed 16 people in Gabon”

will trigger the creation of an incident—a record in a relational database—and assign the underlined values to the corresponding attributes. Each record extracted from the document is permanently stored, together with links to the exact offsets in the text where its attributes were found within the document.

Figure 3 presents a view of the database, as it appears on the Web site. This collection of rows was returned in response to a user query, which is specified by constraints on some of the attribute columns. Here, the constraints are on publication date (April 2007), disease (avian influenza) and country (Indonesia or Cambodia). The constraints are entered into the text boxes below the column names. (The table is ordered by publication date by default.) Blue rows in the table correspond to confident events (defined below in section 5), and white rows are less confident.

For detailed information about the design principles behind PULS, see, e.g., [9,10]. The system relies on several kinds of domain-independent and domain-specific *knowledge bases*. An example of domain-independent knowledge is the location hierarchy, containing names of countries, states or provinces, cities, etc. An example of a domain-specific knowledge base is the medical ontology, containing names of diseases, viruses, drugs, etc., organised in a conceptual hierarchy. The ontology currently contains 2,400 disease terms; 400 vectors (organisms that transmit disease, like rats, mosquitoes, etc.); 1,500 political entities—countries, their top-level divisions and name variants; over 70,000 location names (towns, cities, provinces).

PULS uses pattern matching to extract events; the system contains a domain-specific pattern base—a cascade of finite-state patterns, which map information from its syntactic representation in the sentence to its semantic representation in the database records. For example, the above sentence about Ebola will be matched by a pattern like:

NP(disease) VP(kill) NP(victim) [’in’ NP(location)]

The pattern first matches a noun phrase (NP) of semantic type *disease*; “Ebola” is a descendant of the *disease* node in the ontology. Then it matches a verb phrase (VP) headed by the verb *kill* (or its synonyms in the ontology). The verb phrase also subsumes modifier elements, such as the auxiliary verb *has*, the adverbial phrase *so far*, etc. The square brackets indicate that the locative prepositional phrase is optional; in case the location is omitted in the sentence, it is inferred from the surrounding context.

Populating the knowledge bases requires a significant investment of time and manual labour. PULS employs weakly-supervised learning to reduce the amount of manual labour as far as possible, by bootstrapping the knowledge bases from large, un-annotated document collections, [11,12].

Various views may be used to present the relational data. Especially important are views that aggregate the information according to the user’s criteria. Examples of views the PULS system provides include geographic maps, as in Figure 4 and charts or histograms, as in Figure 5.

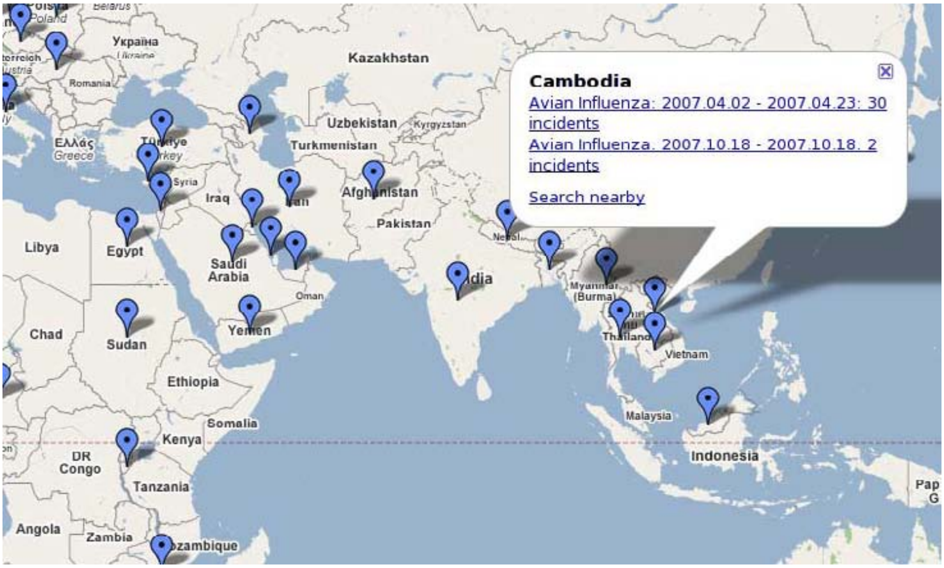


Figure 4. A geographic map of bird flu incidents

4. Cross-document aggregation

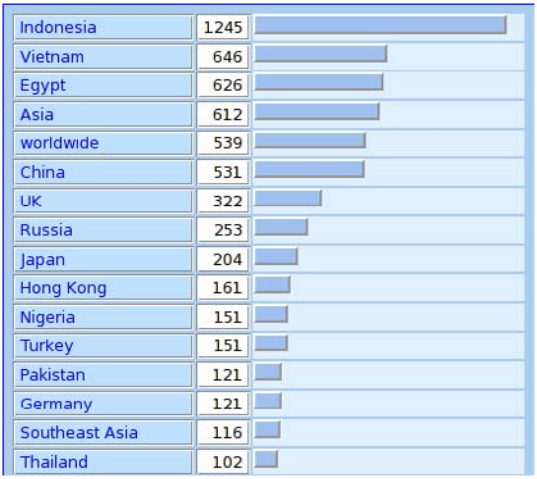
Besides the accuracy of the MedISys filtering and categorisation, an important issue for users is multiple reporting: due to the high number of independent news sources, MedISys captures many reports that readers of one or a few news sources would miss, but the flip-side of the coin is multiple reporting. This causes extra work for the users and makes monitoring daily news a time-consuming task. The solution to this problem lies in the aggregation of reports into larger units. MedISys and PULS use different approaches to aggregation, which are not currently integrated.

4.1. Clustering in MedISys

MedISys presents news *clusters* to the users, grouping similar news reports arriving within at most 8 hours of each other. The short time window means that clusters normally contain articles published within the same day. If reporting continues steadily, articles from different days will be grouped into the same cluster. The similarity measure for the news articles is based on cosine similarity on a simple vector-space representation of the first 200 word tokens of each article. This means that not only multiple reports of the same story, but also similar reports about different cases for the same disease may be grouped together. This method also allows users to discard entire groups of *non-relevant* articles (e.g., discussions about vaccination campaigns) at once.

4.2. Grouping disease events into outbreaks in PULS

PULS provides another means of aggregating multiple individual facts into larger units. PULS goes beyond the traditional IE paradigm in two respects. First, in a typical IE system, documents are processed separately and independently; facts found in one document



Indonesia	1245
Vietnam	646
Egypt	626
Asia	612
worldwide	539
China	531
UK	322
Russia	253
Japan	204
Hong Kong	161
Nigeria	151
Turkey	151
Pakistan	121
Germany	121
Southeast Asia	116
Thailand	102

Figure 5. A chart of bird flu reports (over a time period specified by a user's query)

do not interact with information found in other documents. Second, for each attribute in an extracted incident, the IE system stores only one value in the database record—the locally best guess for that attribute.

1. After PULS extracts information from each document locally, it attempts to globally unify the extracted facts into groups, which we call *outbreaks*. An outbreak is a set of related incidents. Currently, incidents are related by simple heuristics: they must share the same disease name and the same country, and occur reasonably 'close' in time. Closeness is determined by a time window, currently fixed at 15 days.[8] A chain of incidents, any pair of which are separated by no more than the time window, are aggregated into the same group. Thus, an outbreak is a kind of a 'bin' containing related incidents, and provides an added level of abstraction above the 'low-level' facts/incidents.

2. When PULS stores a record in the database, for each attribute, in general, rather than storing a single value, PULS stores a distribution over a set of possible values. For example, the sample text (in the first paragraph of this section) might read instead *"Five more people died last week."* PULS will then try to fill in the missing attributes (i.e., the disease name, location) by searching for entities of the corresponding semantic type elsewhere in the discourse. In general, for a given attribute of an event, the document will contain several possible candidate entities, and each candidate will have a corresponding score—measuring how well it fits the event. The score depends on certain features of the candidate value. These features include whether the value is mentioned inside the *trigger*—the piece of text that triggered some pattern from the pattern base; whether it appears in the same sentence as the trigger; whether it appears before or after the sentence containing the trigger; whether this value is the unique value of its type in the sentence that contains the trigger (e.g., the sentence mentions only a single country, or disease); whether the value is unique in the entire document; etc.

Using a set of candidate values rather than a single candidate is helpful in two ways. First, it allows us to compute the *confidence* of an incident, which is used in cross-

[8] The time window could be made more sensitive, e.g., dependent on the disease type.

document aggregation (in section 5). Second, it allows us to explore methods for recovery from locally-best but incorrect guesses by using global information.[9]

The next section shows how these features are used in the combined system.

5. Integration

This section describes the integration between MedISys and PULS, and tries to demonstrate that even in this early state, the whole is greater than the sum of its parts.

A special RSS tunnel has been set up between MedISys and PULS. At present, PULS processes only English-language documents. MedISys forwards documents which it categorises as relevant to the medical domain through the tunnel to PULS. Currently, the documents arrive as plain text. This is done in addition to the normal processing on the MedISys side, where running averages are monitored for all alerts, etc. A document batch is sent every 10 minutes, with documents newly discovered on the Web.

On the PULS side, the IE system analyses all documents received from MedISys, and returns information that it extracted from the received documents back through the tunnel—in structured form (also at 10 minute intervals). This communication is asynchronous, while both sites are operating in real-time.

When PULS receives documents from MedISys, it performs the following steps:

First, the IE system analyses the documents, extracts incidents, and stores them in the database (at *http://doremi.cs.helsinki.fi/jrc*). Second, PULS uses document-local heuristics to compute the *confidence* of the attributes in the extracted incidents.

The confidence of an attribute is computed from the set of candidate values for that attribute, based on their scores, which are in turn based on the features, as explained in Section 4.2. If the score of the best value exceeds a certain threshold, the attribute is considered *confident*. Some of the attributes of an incident are considered to be more important than others: here, in the case of epidemic events, these *principal* attributes are the disease name, location and date. If all principal attributes of an incident are confident, the entire incident is considered confident as well.[10]

Third, the system aggregates the extracted incidents into *outbreaks*, across multiple documents and sources. The aggregation process requires that at least one of the incidents in each outbreak chain must be confident (that is, chains composed entirely of non-confident incidents are discarded).

Finally, PULS returns a batch of recent incidents to MedISys, for displaying on its pages. The goal is to return a set of incidents with high confidence and low redundancy—a complete yet manageably-sized set for the user to explore. The batch is restricted to documents published within the last 10 days; from this period, PULS returns the most recent 50 incidents, filtering out duplicates: if multiple incidents of the same disease in the same location are reported, PULS returns only the most recent one.[11]

[9] This line of current research is not covered in the present chapter.

[10] In the PULS tables, confident attributes appear in bold, and confident incidents are highlighted.

[11] Note that this implies that a recent event that was last reported more than 10 days ago, will not appear in the result list, while an event from several months ago may appear—if it is mentioned in a very recently published report. This is a design choice that aims for a balance between recency of *publication* vs. recency of *occurrence* of an incident: both may be important to the user. Note also that in any case *all* events are available for browsing in the PULS database.

On the MedISys side, the returned events are displayed in two views. The main MedISys page shows the five most recent events—these correspond to the most urgent news. For more detail, this box has a link to the batch of 50 most recent incidents. For the complete view, the recent list has a link to the PULS database.

6. Evaluation—Summary of Results

The public MedISys site is currently accessed by an average of 1,700 distinct users every day and over twenty Public Health authorities make use of the restricted site, the customisable Regional News Service view of the data, and the automatically sent notifications. Generally, the feedback from users has been positive. When giving feedback, users typically ask for more news sources or alert categories. Heavy MedISys users sometimes express the wish for a more thorough filtering of the news in order to reduce the number of articles that mention an alert out of context. For instance, news about a a celebrity in the context of which a disease is mentioned is considered to be unwanted noise. One way to tackle this issue would be through using automatically trained classifiers that separate relevant from irrelevant news (e.g., sports, film, etc.). Another option would be to tighten the filter by tuning the binary alert definitions, using more search words and applying weights. This tuning of alert definitions is an on-going process, as alert definitions are updated all the time. The third option is to combine the Information Retrieval approach used in MedISys with a subsequent Information Extraction phase, as presented in this chapter.

6.1. Evaluation of the filtering and classification in MedISys

We evaluated disease-related alerts in MedISys by selecting 100 English-language articles.[12] The articles were selected randomly, with a maximum of ten articles taken from a given day, in order to investigate a broader variety of news articles and alerts. Only those articles were considered which triggered some MedISys alert definition.

In these 100 news articles, 156 relevant alerts were found. For these alerts, a human expert judged the accuracy of the alert, by assigning it to one of four categories:

1. The MedISys alert assigned to the news article is appropriate and the article mentioned a disease outbreak event. 63 alerts fell into this category.
2. The disease name was mentioned in the article, but in the context of vaccines, new drugs, or similar. 74 alerts fell into this category.
3. The MedISys alert was inappropriate. This category consists of cases where the disease name was mentioned, but the article was about generic issues such as politics, literature, finance or sports. In a small number of cases, person names or other words are homographic with a disease name triggered the category. For example, the term *Mobility Aids* triggered the category HIV. 19 alerts fell into this category.

[12] Note that the mandate of MedISys is broader than monitoring communicable diseases, as some users are interested in chemical or nuclear incidents, in mentions of vaccines or new medicines, etc. However, because PULS at present identifies only events describing outbreaks of diseases, the evaluation was limited to this subset of alerts. All other alerts were ignored.

4. The fourth group consists of articles mentioning a disease that should have been identified, but were not, corresponding to the *false-negative* measure (which has an impact on recall). Some examples: The word *HIV-positive* did not trigger the disease HIV, due to a tokenisation error; *Foot & Mouth Disease* was not recognised because the disease name variant with the ampersand was not part of the alert definition. 13 alerts fell into this category.

Problems in category 4, such as tokenisation errors and missing disease name variants, are easy to correct. This example shows that continuous quality control is essential, and that the performance of queries involving Boolean operators is highly dependent on the quality of the search words. Since MedISys and NewsBrief have hundreds of alert categories for a wide variety of users, it is clear that the users have to use their subject knowledge and control their own alerts. For that purpose, an alert-editing interface is available to specialist users.

The evaluation shows that the results for concrete alerts involving one or more disease names are: 137 (63+74) out of 156 alerts were relevant, which corresponds to a *precision* of 0.88. The performance for more abstract alerts which cannot so simply be defined via the occurrence or non-occurrence of specific words (e.g., 'fraud' or 'stress at the work place') is expected to be lower.

6.2. Evaluation of event detection

PULS receives on the order of 10,000 documents from MedISys each month. From 27% of these documents, PULS extracts about 6,000 incidents, on average.[13] The remaining 73% of the documents processed by PULS yield no incidents. This is as expected, since MedISys does not explicitly select for outbreaks, but for mentions of disease names in *any* context, and many documents may mention diseases in contexts unrelated to epidemics and outbreaks.

To estimate the proportion of documents rejected by PULS that contain missed events—false negatives—we manually checked 200 MedISys documents that produced no events. Among these documents, 14% contained an event that the IE system had missed.[14] As PULS filters out 73% of the incoming documents, adding back the incorrectly filtered documents (14% of all filtered), yields that about 63% of the documents that arrive to PULS contain no *epidemic* events. In this way, the IE phase helps to distinguish reports about epidemic outbreaks from documents that mention diseases in other contexts.[15]

We tried to estimate the accuracy of event detection and the confidence heuristic. Twenty percent of all extracted incidents are rated as confident by PULS. We selected 100 confident incidents at random, and checked their correctness manually. In this evaluation we took a conservative (strict) approach: we considered an incident to be correct

[13] In IE, it is typical for a relevant document to contain more than one incident, since often there is one or more main events, and other, related events are mentioned as part of background discussion.

[14] NB: this does not correspond to the false-negative rate. To compute the false-negative rate, recall at the document level, and recall at the level of events would require a more detailed evaluation, to be conducted in the future.

[15] MedISys has an optional boolean filter that tries to capture outbreaks by requiring the name of the disease to occur in combination with keywords like *bedridden*, *hospital**, *deadly*, *cases*, etc. This has not yet been evaluated.

only if all of its 'principal' attributes are correct, i.e. no credit is granted for partially correct events, unlike in standard IE evaluation. The result was: 72% of the confident incidents are correct; in 14% of the cases, the information extraction is spurious, i.e., PULS extracts an incident where there should be none; in another 14% of the cases, the confident incident is incorrect—i.e., at least one of the attributes has an incorrect value (the top-ranked value is wrong). The latter category of error is difficult to correct, since it is usually due to an inherent complexity in the text. The former type of error may be simpler to correct, through further tuning of the knowledge bases.

Since outbreak aggregation is our primary means of reducing redundant information in the flow of news to the user, it is important to estimate the accuracy of the outbreak grouping. We analysed a randomly chosen set of medium-sized outbreaks: 20 outbreaks with approximately 10 incidents in each. For each incident we determined whether it was appropriately included in the outbreak. 68% of the incidents were correctly identified with their outbreaks. Three of the outbreaks (about 15%) were erroneous, i.e., based on incorrect confident incidents.[16]

Of all the incidents examined in this evaluation 22.5% were confident (i.e., on average, the outbreaks contained only 2–3 confident incidents).

7. Conclusion and future work

The combination of the two initially independent systems MedISys and PULS has lead to a stronger application offering users complementary functionality through a unified user interface. For communicable disease outbreaks, which are covered by both systems, the combination of IR in MedISys and IE in PULS leads to additional advantages: Firstly, PULS's computationally heavier methods only need to be applied to the document collection pre-filtered by MedISys. Secondly, the medical event extraction patterns act as an additional filter to identify only disease outbreak reports. MedISys is designed to capture not only disease outbreak reports, but also other news articles mentioning diseases. For users interested specifically in disease outbreaks, PULS's event recognition helps reduce the number of reports by filtering out just under three quarters of incoming reports, of which about 14% are incorrectly filtered relevant reports.

The current status of integration can be taken further: the systems don't yet make full use of the other's information aggregation methods. The categorisation of news items by MedISys can be useful for the analysis performed by PULS, and is yet to be utilised. The taxonomies used by the systems are overlapping, but have not yet been fully integrated. These and other issues are to be tackled in future work.

While we believe that the combination of IR in MedISys and IE in PULS provides added value, it is not a universal solution. An important strength of MedISys is its multilinguality: it monitors media reports in currently 43 languages. Developing PULS-style event extraction grammars for so many languages is not currently possible: porting the IE

[16] It was interesting to observe that aggregation is often useful even when the outbreak consists entirely of incorrectly analysed incidents. For example, in high-profile cases picked up by main news agencies, reports are re-circulated through multiple sites worldwide. Because the text is very similar to the original report, the IE system extracts similar incidents from all reports, and correctly groups them together. Although some attribute is always analysed incorrectly, the error is *consistent*, and the grouping is still useful: it helps reduce the load on the user by aggregating related facts.

system to a new language requires pre-existing robust lower-level linguistic components (named entity tagger, ontology, parser) for each new language, which are unlikely to be available for all the languages covered by MedISys in the near future. However, focusing on the major languages for which lower-level linguistic resources have been developed is planned for future extensions.

Acknowledgements

The development of MedISys would not have been possible without the members of the Web Mining and Intelligence team, who contributed over the years. Work on PULS was supported in part by the Academy of Finland grant 118653, National Center of Excellence *"Algodan" (Algorithmic Data Analysis).*

References

[1] R. Gaizauskas and A. Robertson, "Coupling information retrieval and information extraction: A new text technology for gathering information from the web," in *Proceedings of the 5th RIAO Computer-Assisted Information Searching on Internet*, Montreal, Canada, 1997.

[2] Defence Advanced Research Projects Agency, "Information extraction task: scenario on management succession," in *Proc. 6th Message Understanding Conf. (MUC-6).* Columbia, MD: Morgan Kaufmann, 1995.

[3] A. Robertson and R. Gaizauskas, "On the marriage of information retrieval and information extraction," in *Information retrieval research 1997: Proceedings of the 1997 annual BCS-IRSG colloquium on IR research, Aberdeen, Scotland*, J. Furner and D. Harper, Eds. London: Springer-Verlag, 1997.

[4] S. Doan, Q. Hung-Ngo, A. Kawazoe, and N. Collier, "Global Health Monitor—a web-based system for detecting and mapping infectious diseases," in *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2008.

[5] C. Freifeld, K. Mandl, B. Reis, and J. Brownstein, "HealthMap: Global infectious disease monitoring through automated classification and visualization of internet media reports," *J Am Med Inform Assoc*, vol. 15, pp. 150–157, 2008.

[6] C. Best, E. van der Goot, K. Blackler, T. Garcia, and D. Horby, "Europe Media Monitor—system description," EUR, Tech. Rep. 22173 EN, 2005.

[7] R. Steinberger, B. Pouliquen, and C. Ignat, "Navigating multilingual news collections using automatically extracted information," *Journal CIT*, vol. 13, no. 4, 2005.

[8] B. Pouliquen, M. Kimler, R. Steinberger, C. Ignat, T. Oellinger, K. Blackler, F. Fuart, W. Zaghouani, A. Widiger, A.-C. Forslund, and C. Best, "Geocoding multilingual texts: Recognition, disambiguation and visualisation," in *Proceedings of LREC-2006*, Genova, Italy, 2006.

[9] R. Yangarber, L. Jokipii, A. Rauramo, and S. Huttunen, "Extracting information about outbreaks of infectious epidemics," in *Proc. HLT-EMNLP 2005*, Vancouver, Canada, 2005.

[10] R. Grishman, S. Huttunen, and R. Yangarber, "Information extraction for enhanced access to disease outbreak reports," *J. of Biomed. Informatics*, vol. **35**, no. 4, 2003.

[11] R. Yangarber, "Counter-training in discovery of semantic patterns," in *Proc. ACL-2003*, Sapporo, Japan, 2003.

[12] W. Lin, R. Yangarber, and R. Grishman, "Bootstrapped learning of semantic classes from positive and negative examples," in *Proc. ICML Workshop: Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington, DC, 2003.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-311

Learning to Populate an Ontology of Politically Motivated Violent Events

Hristo TANEV [1], Pinar WENNERBERG
European Commission, Joint Research Centre, Ispra, Italy

Abstract. In this paper we present a working event extraction and classification infrastructure, which monitors news articles, detects and extracts structured descriptions of Politically Motivated Violent Events (PMVE) and feeds them into an ontological knowledge base. A knowledge base of about 2800 PMVE was built semi-automatically.

Keywords. ontology, ontology population, event extraction, news monitoring

Introduction

State-of-the-art artificial intelligence systems use knowledge bases for representing generic or domain-specific knowledge, which is typically realized by an ontology. Some domains require encoding large amounts of knowledge that may change rapidly. Therefore, handcrafting and maintaining a comprehensive ontology, even for a specific domain, can be quite expensive in terms of time and resources. For this purpose, methods and tools for automatic ontology population are adopted as a part of the ontology engineering cycle.

There are several factors, which make such knowledge bases and ontology population tools relevant to the security domain. Firstly, analysts are interested in many different types of relations between people, organizations and events. Secondly, automatic reasoning mechanisms integrated in knowledge bases can be used to suggest non-obvious links between people, places, organizations, and events. Thirdly, the dynamic nature of the security domain and the large number of names and relations present in the relevant news articles make it very difficult to keep these knowledge bases updated. Therefore, it is essential to exploit the tools for automatic and semi-automatic ontology population.

Automatic Event Extraction (EE) [1] is a task highly relevant for the security domain. Its goal is the detection of events of interest (e.g. violent events) in natural language texts and automatic extraction of the time and place of each event, as well as its participants (e.g. perpetrators and victims in the security related domain), instruments used, weapons, etc. The output of a security EE system may be used to populate automatically or semi-automatically an ontological knowledge base of security-related events.

[1] Corresponding Author: Hristo Tanev, Via E.Fermi 2749, 21027 Ispra, Italy, htanev@gmail.com

On the knowledge modeling side there have been several initiatives to gather and represent security related data. Some approaches are based on text document collections [2], some on structured databases [3] and there is at least one approach that uses a knowledge base [4]. Typically, these resources store information about significant terror organizations, key persons and terror acts over a certain period of time. With the emergence of the Semantic Web, web-based terrorism ontologies have been constructed [5], which act as components of larger systems and provide them with machine-processable domain knowledge.

A common major problem in this area is the so-called *knowledge acquisition bottleneck*, i.e. the acquisition of related data to populate these ontologies. Even though some systems [6] provide intelligent user interfaces for knowledge entry, the human experts, especially in the security domain, are not easily accesible or they are expensive. Therefore, automatic extraction of the security related information from readily available sources, such as the news reports is an appropriate solution to this problem.

In this paper we address the problem of automatic detection and extraction of Politically Motivated Violent Event (PMVE) instances from online news articles and using them to semi-automatically populate the ontology of PMVE. The PMVE knowledge base is a result of these two activities and the entire process is thus realized by two components. The first component is the NEXUS event extraction system that detects the descriptions of the PMVE in the news articles. After the validation of the detected events by a human expert, the second component, which is the semantic engine, populates the knowledge base by devising the PMVE ontology. The PMVE ontology that provides the system with machine-processable knowledge about conflict events is only concerned about the events that have an impact on the society, for example terror attacks, bombings, hijackings etc. Single criminal acts such as bank robberies or murders are out of the scope of this work.

Gathering and modeling knowledge about the PMVE is an extremely important task for a better understanding of the situation in politically unstable regions. A proper analysis not only results in early warning for incoming conflict and political crises but also in actionable decisions of the policy-makers.

1. Background

1.1. Event extraction

In 2004 eVent Detection and Recognition task (VDR) was introduced at the Automatic Content Extraction initiative (ACE) [7] organized by the National Institute of Standarts (NIST). Different event types were considered in this task: business events, security-related events, life events, such as birth and death, etc. At ACE 2007 only one participant submitted results for the VDR task, gaining score 13.4% of the possible maximum. These figures show that the EE is a hard task. [8] describe different linguistic and semantic phenomena, which make EE complex and hard to tackle. The two most important phenomena are event descriptions being scattered and overlapping.

Typically, the EE systems use structural patterns (see among the others REES [9] and PULS [10]), which are created manually, as it was done for the REES system, or via machine learning [11]. Other systems use an annotated corpus to learn context features via standart machine learning techniques [12].

When working with news sources, it becomes important to detect the mentions of the same event in different documents. Currently, few state-of-the-art EE systems are capable of aggregating information across documents. For example, the PULS - MediSys system [10] combines information about communicable disease incidents from different reports into disease outbreak events. This system finds the time period, the place, and the disease for each outbreak detected. However, it cannot estimate the total number of the victims of the outbreak, even though it detects the number of the victims for each incident report.

1.2. Ontologies

In Artificial Intelligence (AI) ontologies have been used to declare the knowledge embedded in a knowledge base system and to facilitate knowledge sharing and re-use. In the aftermath of September 11 attacks there has been increased research in security intelligence. Thus various ontologies and knowledge bases have been constructed that concentrate on studying the relationships between terrorist organizations, individual terrorists, related locations etc. One of the recent works on this topic is reported in [13]. This is an ongoing project to collect terrorism related data from publicly available sources and to structure it by using a terrorism ontology. The Terrorism Knowledge Base (TKB) mentioned earlier is one of the most comprehensive non-ontology based terrorist knowledgebases, however work has been done [14] to incorporate ontologies in order to improve its search facilities. In a separate initiative, the Cyc Project enhanced its existing knowledge base from diverse sources to provide intelligence analysis in the terrorism domain [15].

Social Network Analysis, (SNA) [16] is an approach to analyze social entities such as people and organizations and their interrelations by focusing on a network-based view. It has been used in the security intelligence domain, particularly in the analysis of terrorist [17,18] or criminal networks [19].

Our approach here differs from these approaches in that it combines the automatically extracted violent events with a hand-crafted domain ontology, which provides a semantically complete and consistent knowledge model. This model can then be used in future reasoning processes such as automatic hypothesis generation. Deriving from Swanson's *ABC model* [20] we assume that there is interrelating information about the PMVE in distinct news reports and if these events were mapped to a common semantic model the non-obvious interrelations would be discovered or inferred. More concretely, we propose the following scneario: *News report 1:* Person A isRoommateOf Person B. *News report 2:* Person B actorOf Violent Event E. Hypothesis: Person A also relatedTo Violent Event E?

In the following sections we first describe, how we automatically obtain violent event information from news reports and then we explain how this information is mapped to the PMVE ontology.

2. NEXUS

NEXUS is our prototype EE system, developed to detect PMVE (Politically Motivated Violent Events) in news clusters, created by the Europe Media Monitor (EMM) family of applications [21]. EMM is a dedicated infrastructure for news monitoring, which downloads news articles from the Web around the clock and automatically clusters them according to their topics. Each news cluster ideally contains all the news from the previous day which refer to the same event.

NEXUS pre-procecesses linguistically the documents from each cluster. The pre-processing step includes tokenization, sentence splitting, named entity recognition (i.e. names of organizations, persons, and locations), and unnamed person groups detection (e.g. recognizing *five workers*). In the text pre-processing phase, a geo-coding of documents in each cluster is performed, which is relevant for defining the place, where the main event of the cluster took place.

Once news articles are pre-processed, the pattern engine applies a set of extraction rules on each document within a cluster. For creating extraction patterns we apply a blend of machine learning and human validation, which is explained in the following subsection.

The extraction rules are matched against the first sentence and the title of each article from the cluster. By processing only the top sentence and the title, the system is more likely to capture important facts.

Next, our system filters out all the clusters, where no extraction pattern was applied in the previous stage. Remaining clusters do not necessarily have to be descriptions of PMVE, since our patterns for extraction of dead and wounded people can match in the descriptions of natural and man-made disasters, as well as in the descriptions of criminal acts. For this reason, we apply a second filter, which accepts only those clusters, which trigger certain keyword-based alerts like “security” and “terrorist attack”. The alert system is an integral part of the EMM.

Since information about events is scattered over different documents, the last step consists of cross-document cluster-level information fusion, i.e. NEXUS aggregates and validates automatically information extracted locally from each single document in the same cluster. The quantitative estimation of the victims and the event type are the most important outcomes of this stage. The system finds at most one main PMVE for a cluster. Each PMVE is described through a record, whose main fields are: date, place, number of killed, number of injured, number of kidnapped, perpetrators (e.g. “gunmen”, “Hizballah”), weapons used (e.g. “napalm”), and event type (e.g ‘TerroristAttack, Assassination, SocioPolitical, etc.)

In the following sub-sections we will explain, how we acquire patterns and keywords, which are used to extract entities that fill these fields.

2.1. Pattern acquisition schema

We adopted an automatic pattern acquisition method, whose output is validated by a human moderator. The automatic approach generates many templates in a short time - this contributes to the recall of the pattern library. On the other hand, the human validator rejects patterns, which are erroneus or too generic thus improving the accuracy. Here are the basic steps of our pattern acquisition schema:

1. Annotate a small corpus with event-specific roles, e.g., “date”, “place”, “perpetrators”, “killed”, “injured”, etc.
2. Learn automatically linear extraction patterns. For example, the pattern [PERSON] “was slaughtered” extracts names of people, who are victims in a violent event. We used a novel entropy-based pattern learning algorithm described earlier in [1]. This algorithm finds patterns, which appear in more varying contexts than their sub-patterns.
3. Manually filter out low quality patterns. If no patterns remain after this step, end the algorithm.
4. Match the patterns against the corpus or part of it. Next, entities that fill the pattern slots and comply to the semantic constraints of the slot are taken as anchor entities. If an anchor entity A (e.g., “five people”) is assigned an event-specific role R (e.g., “killed”), in the news cluster C, we assume with high confidence that in the whole cluster C entity A appears mostly in the same role R. This assumption is valid in most of the cases, since all the articles from a news cluster refer to the same event. Consequently, annotate automatically all the occurrences of A in C with the label R.
5. Go to step 2.

2.2. Local Entropy Maximum algorithm for pattern learning

In step 2 of the pattern acquisition schema we learn single-slot patterns from the annotated corpus. We developed a statistical learning algorithm to perform this task.

For each event-specific semantic role (e.g. “wounded”) we extract from the annotated corpus the left and the right contexts of the entities which are annotated with this semantic role. Then we run our learning algorithm separately on the left and right contexts and acquire patterns specific for the semantic role under consideration.

We will explain the algorithm using an illustrative toy example: Suppose we have the right contexts of several entities which are annotated with the semantic role “wounded”. These entities are names of people (e.g. “Marry Sullivan”) or person groups (e.g. “two people”) and in this example their position is denoted by P. Since we consider only right contexts, P is always on the left side: *P was heavily wounded in a mortar attack*; *P was heavily wounded when a bomb exploded*; *P was heavily injured in an accident*; *P was heavily injured by a bomb*; *P was heavily injured when a bomb exploded*

From these contexts we can extract different patterns. For example, we may consider the most general one: *P was heavily* (P is the slot), however such a pattern will erroneously annotate “George” as “wounded” in phrases like *“George was heavily drunk”*. It is obvious that we need a criterion for extracting patterns at the right level of generality. In this clue, for each candidate pattern we consider all its occurencies and their immediate context (i.e., adjacent words). In the above example, for the pattern *P was heavily* we may find two immediate context words:

"wounded" and "injured". Then a context entropy for each pattern t can be calculated using the following formula:

$$context\_entropy(t) = \sum_{w \in context(t)} p(w|t) \cdot ln(p(w|t)^{-1})$$

, where *context(t)* is the set of the words in the immediate context of the pattern t and $p(w|t)$ is the probability that a word appears in this context.

Intuitively, the more words we add to a phrase, the lower its context entropy becomes. However, when a pattern is semantically consistent and complete, it may have higher context entropy than some of its sub-patterns. This is because a complete phrase is less dependent on the context and may appear in different linguistic constructions, while the incomplete phrases appear in a limited number of immediate contexts which complete it. For example, the phrase *P was heavily injured* has higher right-context entropy than *P was heavily.*

A pattern t satisfies the LOCAL CONTEXT ENTROPY MAXIMUM CRITERION only when all the patterns which are its prefixes (suffixes in the case of left contexts) or for which t is a prefix (suffix) have the same or lower context entropy Finally, we select only the patterns which satisfy this crierion In the example above we will select only *P was heavily wounded* and *P was heavily injured.*

2.3. Learning lexicons for event extraction

Event extraction needs lexicons, in which words are provided together with their semantic classes.

For example, in order to detect the weapons used in some violent event, NEXUS has to recognize all the words in the event description that refer to weapons. Moreover, we consider weapons in the context of some semantic categories, e.g. "pistol" and "shotgun" are "light arms", "artillery" and "tank" belongs to the category "heavy weapons", etc.

Manually constructing a lexicon of weapons was not an option for us, since a military expert is necessary for such a specific domain. However, it turned out that in WordNet 2.0 [22] there are many words referring to weapons. We used this lexical database to extract such words. The problem of using WordNet is that the weapons are scattered across different WordNet synonym sets. Moreover, we noticed that some weapons mentioned in the news are not present in WordNet. To overcome these problems, we performed first automatic lexicon learning, next we semi-automatically mapped the acquired lexicon to WordNet, extended it and assigned semantic classes to the words.

Our lexical acquisition approach has four main steps. We will illustrate these steps considering the construction of the lexicon of weapons, however this algorithm is applicable to different semantic categories.

1. Extract words, which can be weapons, by matching seed templates against a syntactically parsed corpus. In our experiments we used these two seed templates: "kill $-with \rightarrow$ [weapon]" ("killed him with a gun") "kill $-subject \rightarrow$ [weapon]" ("the bomb killed him"). We extracted all words and multiwords that match the "[weapon]" slot in the seed patterns. This

way we obtain words denoting weapons, yet also words, which are not related to weapons at all("his envy killed him").

2. Using an approach described in [23], we automatically find a list of syntactic features, which co-occur in the corpus with the words extracted in step 1.
3. Our algorithm represents each word as a vector of its syntactic features and clusters the word vectors.
4. We manually select these cluster (or clusters), which contain mostly entities of interest (e.g. weapons) and clean the irrelevant words, if this is necessary.

Using this procedure, we obtained a list of weapons. Here is a sample of some relevant terms acquired automatically by our algorithm: "bomb", "explosive device", "hand grenade", "grenade", "explosive", "gun", "katyusha", "kassam".

The WordNet expansion of the weapon list is performed in three steps:

1. For each term,which is in WordNet we extract its hypernym chain. As an example, consider the hypernym chain of "mortar": "mortar" - "high-angle gun "- "cannon" - "artillery"... In such a way we obtain a list of WordNet concepts that are semantically more general than the extracted terms.
2. However, some concepts like "object" or "artifact" are too generic. Therefore, we select manually only those WordNet classes that contain only weapons in their synonym sets and whose direct and indirect hyponyms are also weapons.
3. Expand the selected WordNet concepts - we take all the hyponyms and synonyms of each selected WordNet concept and add these words to the list of weapons.

After we exract the lexicon, we manually assign semantic classes to the terms. In order to partially automate this process, we use the structure of WordNet: For example, we manually assign class "heavy weapon" to the WordNet concept "artillery" and this class is automatically propagated to all the synonyms and hyponyms of "artillery".

Using the approach we explained so far, we built a lexicon of weapons, a lexicon of violent event terms like "explosion", "kidnapping", "assassination",etc. and a list of perpetrators like "suicide bomber" and "gunman". These lexicons were used successfully in NEXUS to detect the event type.

3. Knowledgebase of Politically Motivated Violent Events

The PMVE knowledge base has two components, namely the PMVE ontology and the PMVE instances. The development of the knowledge base has been accomplished via several steps.

1. Automatic detection and extraction of violent events from news reports
2. Manual design and development of the PMVE ontology
3. Human verification and classification of the event instances

4. Semi-automatic ontology mapping of the event instances and relations
5. Design and development of the query and user interface

Currently, the PMVE knowledge base stores approximately 2800 violent event instances. For each event, (so far as present) the news resource, the happening date, the number of killed, injured, wounded, the happening location, the names of the perpetrators and victims, a short description of the event, its title, and its relation to other events and other relevant information are available.

The PMVE Ontology

The PMVE ontology was designed by following a formal ontology engineering methodology called METHONTOLOGY [24] in an iterative process. Consequently, the concepts and the relationships of the ontology have changed over time in order to better adapt to the specifics of the domain. It was eventually implemented in the OWL [25] language.

The PMVE ontology is a DAG (Directed Acyclic Graph), so that one sub-concept may have two parent concepts. This makes it possible to arrive at one concept following different paths, which allows a more intuitive way of finding information. For example, the concept SocioReligious event (e.g. Christmas, Ramadan, Yom Kippur) is a sub-concept of both ReligousEvent and SocialEvent,therefore it can be reached starting from either one of its parent concepts.

The higher level concepts of the ontology, as demonstarated in [1], are the Location, Time, Agent and Event, which answer the question *"Who Did What to Whom Where and When?"* and thus reveal the common structure of violent events.Typically, violent event incidents, as they are reported in the news articles, occur at a place and at a time and they involve people and organizations.

The concepts Agent and Event are further broken down to include other relevant sub-concepts, some of which are PeopleGroup, Person, Organization, ConflictEvent, MilitaryEvent, GovernmentalEvent, LegalEvent, SocialEvent, PeaceEvent respectively. Even though it may seem contradictory to include the concept PeaceEvent in the PMVE ontology, we have observed that many conflict events lead to peace events or vice-versa. As we wanted to capture the (non-obvious) relationships between disparate event instances, it was essential to include these connections.

Using the PeopleGroup concept we quantify over unknown people such as "five people", who can be victims or perpetrators of an event, without having to specify them person by person. This allows us to work efficiently with incomplete information, since news reports quite often provide very generic information such as *"five killed in a bombing near......"*. The unknown perpetretors of the ConflictEvents are marked as Insurgents, thus they automatically become the *"organizerOf"* ConflictEvents.

Most frequently observed violent events are HostageTaking, Clash, Bombing, SuicideBombing and Raid, which are the sub-concepts of ConflictEvent. Their sibling concepts EventsUnderInvestigation and PossibleThreat model conflict events that may not have yet taken place or no final statement about the status of the event exists. They help keeping track of the situation for early warning and conflict event hypothesis generation purposes.

The concepts Escape and Flee are equivalents (synonyms) to allow finding one conflict event instance under both concepts. Generally, modeling with equivalent concepts is a useful technique, which helps solving problems that are due to the ambiguity of the natural language. However, it should be used sparingly as it creates cycles in the ontology, which may later have consequences on the efficiency and scalability of the system.

The relationships in the ontology associate two concepts to each other. In our domain People are *organizerOf, victimOf, funderOf* Events or they will be *killedAt, woundedAt, injuredAt* Events that *happenAt* a Time. Similarly, Organizations can be *organizerOf* Events, they can be *locatedAt* a Place, where another Person *residesAt* . People can be *siblingOf, housemateOf, friendOf, parentOf* other People.

The relationships can also be represented in a hierarchy. For example, the *sisterOf, brotherOf* relationships are specializations (i.e. the sub-relationship) of the *siblingOf* relationship. To achieve semantic completeness, the ontology defines for every relationship an inverse relationship. More concretely, if it is true that a relationship *killedAt* exists then it is true that its inverse *kills* also exists.

Additionally, there are reflexive, symmetric and transitive relationships in the PMVE ontology. The *subeventOf* (inverse: *hasSubevent*) is an example of both a reflexive and a transitive relationship. As explained in [26],through reflexivity a particular event will be the subevent of another. For example, many Bombing events include Killing events, as people get killed, when bombs explode. Through transitivity, we can relate two distinct events with each other if they both share the subevent relationship with a third event. With symmetric relationships such as *colleagueOf*, it is possible to discover cases, where two People are *colleagueOf* each other and one is a policeman by profession. Consequently, we can conclude that the other person is also policeman as they are colleagues.

Hierarchy of relationships and concepts enables a convenient exploration of the model as users can navigate from specific to generic relationships and vice versa. For example, *numberKilled* and *numberWounded* properties are subproperties of the *damage* property. Finding the number of people, who are *killedAt* a given event, one can also find the number of people that are *woundedAt* that event by first navigating from the *numberKilled* property up to *damage* and then down again to *woundedAt* property.

Currently, the PMVE ontology includes approximately 100 concepts, more than 100 relations and properties as well as some axioms. An important design criterion was to keep the ontology as simple as possible, yet as expressive as necessary. As a result, complex axioms and rules were avoided that can cause scalability problems in future reasoning processes.

In the next sub-section we will elaborate on retrieval of event instances and the user interaction with the model.

3.1. PMVE Instances, Semantic Queries and the User Interaction

Encoding coherent, consistent and valid knowledge into the knowledge base is the premise to an efficient inference mechanism. Therefore, we included a human moderation process that verifies the quality of the event instances, before they are inserted to the knowledge base.

Another advantage of the human moderation process is that commonsense knowledge can be incorporated. For example, when the news report says *“Attacks come on 2nd anniversary of U.S. invasion”*, we know by commonsense that there are some people, who organize the attacks and that the attacks do not come on their own. Consequently, the information about the perpetrators can be entered by the human moderator even if it is mentioned elsewhere in the same or in a seperate article. As such, the human moderation process consists of modifying and entering information to the knowledge base and eventually assigning each event instance to one of the ConflictEvent concepts of the ontology.

The related event instances are retrieved from the knowledge base using the semantic query language SPARQL [27]. SPARQL is a SQL like query language that operates on the RDF model [28] and it matches the query relevant parts of the ontology graph. For example, the user can retrieve all the instances of a specific event type (e.g. Arrest) listed by their titles using the following query:

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX PMVE:<http://www.owl-ontologies.com/PMVE.owl#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT ?instances ?titles
WHERE {?instances rdf:type PMVE:Arrest.
?instances PMVE:eventTitles ?titles}
```

Interaction with the knowledge base is done via the PMVE browser, which allows users to navigate through the concepts of the PMVE ontology. In particular, the ConflictEvent concept and its sub-concepts are displayed to the user. Using another semantic query all the specifics of one particular violent event instance can be examined. As the relations between seperate events (so far as they exist) are also displayed, the user can navigate from one event instance to the other thereby exploring the knowledge base.

Acknowledging that the analysts and policy-makers are typically non-technicians, the user interface (as well as the ontology model) has been designed to provide a convenient and intuitive information exploration and retrieval. Therefore, all semantic queries happen in the background so that the user does not have to write his own query. This has one disadvantage that customized queries are not possible, however it abstracts the user away from the intimidating technicalities of the system.

4. Experiments and Conclusions

We set up a working PMVE ontology population infrastructure, which finds candidates for PMVE on a daily basis. The PMVE are then passed to a human moderator, who validates and feeds them in a semi-automatic manner into the ontological knowledge base. The events from the past were also processed by running this infrastructure on our archive data.

We carried out a preliminary precision evaluation of the NEXUS performance on 26333 English-language news articles grouped into 826 clusters from the period 24-28 October 2006. NEXUS found 47 candidates for PMVE, 39 of them were real PMVE. The precision of detecting of dead and wounded people was found to be 91% and the precision of finding kidnapped people was 100%. NEXUS detected perpetrators of PMVE with accuracy 69%. These figures show acceptable performance in terms of precision. In a separate evaluation on randomly taken 61 clusters, our system detected all the 6 PMVE reported. Our experiments demonstrated that NEXUS can be a quite useful tool for country monitoring and political analyses of any type. Furthermore, the development of a knowledge base of PMVE may be a step towards more intelligent and integrated data view and analysis.

The hand-crafted PMVE ontology constitutes the core of the knowledge base. Some 2800 event instances from 01.01.2005 to 31.05.2007 were inserted to the knowledge base, after having been verified by a human expert. The PMVE knowledge base builds upon the NEXUS event extraction system and demonstrates how data-driven and knowledge-driven information processing technologies can be combined to develop systems that assist security intelligence analysts in their decision making processes. The PMVE knowledge base realizes this by first organizing the automatically detected conflict event instances under navigatable, generic categories (concepts) such as Bombing, Killing etc. by using the PMVE ontology that can be navigated. Finally, analysts can retrieve all relevant information about a specific event instance and its relations to other event instances at one place, which gives them the necessary insight to the domain.

5. Future Work

There are several tasks identified for future work. Firstly, we would like to improve the automatic event extraction in NEXUS: In particular, we will concentrate on more precise body counting, recognizing references to past events and using syntactic analysis to detect long distance dependencies. Secondly, we aim at implementation of an inference engine that will allow the automatic discovery of new relations and potential violent events. Another challenging task is to include temporal reasoning to model the evolution of events over time. We also target a finer grained classification of subevents. In particular, some events can be considered as “precedent”/“motivator” or “antecedent”/“outcome” of the main event. Finally, an instance-based visualization interface is planned to provide the analysts with additional convenient tools to explore knowledge.

References

[1] J.Piskorski, H.Tanev, and P. Wennerberg: Extracting Violent Events From On-Line News for Ontology Population, *Business Information Systems, Lecture Notes in Computer Science*, Springer, Berlin, 2007

[2] V.E.Krebs: Mapping networks of terrorist cells, *Connections*, 2001, 24(3), 43–52

[3] ICT, 2006, Available: http://www.ict.org.il/

[4] TKB, 2006, Available: http://www.tkb.org/

[5] A.Sheth, Aleman-Meza B., Arpinar I. B., Halaschek C., Ramakrishnan C., Bertram C., Warke Y., Avant D., Arpinar F. S., Anyanwu K., and Kochut K.: Semantic association identification and knowledge discovery for national security applications, *Special Issue*

of Journal of Database Management on Database Technology for Enhancing National Security, 2004

[6] D. B. Lenat, R. V. Guha. Cyc: A Midterm Report, *AI Magazine*, vol.11, no.3, 33-59, 1990.

[7] ACE, http://www.nist.gov/speech/tests/ace/index.htm

[8] S.Huttunen, R.Yangarber, and R.Grishman: Diversity of Scenarios in Information Extraction, *Proceedings of the 3rd International Conference on Language Resources and Evaluation LREC-2002*, Las Palmas, 2002

[9] Chinatsu Aone and Mila Ramos-Santacruz: REES: A Large-Scale Relation and Event Extraction System, *Proceedings of the sixth conference on Applied natural language processing*, Seattle, Washington, 2000

[10] R.Yangarber, C.Best, P. von Etter, F.Fuart, D.Horby, and R.Steinberger: Combining Information about Epidemic Threats from Multiple Sources, *Proceedings Multi-source, Multilingual Information Extraction and Summarization at RANLP-2007*, Borovets, Bulgaria, 2007

[11] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen: Automatic Acquisition of Domain Knowledge for Information Extraction, *Proceedings of the 18th conference on Computational linguistics*, 2000

[12] D.Ahn: The Stages of Event Extraction, *Workshop ARTE* , 2006

[13] A. Mannes, J. Golbeck: Building a Semantic Web Portal for Counterterror Analysis, *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2007

[14] L. Gruenwald, G. McNutt, A. Mercier: Using an Ontology to Improve Search in a Terrorism Database System, *DEXA Workshops 2003*, 753-757, 2003

[15] D. Schneider, C. Matuszek, P. Shah, R. Kahlert, D. Baxter, J. Cabral, M. Witbrock, D. Lenat: Gathering and Managing Facts for Intelligence Analysis, *Proceedings of the International Conference on Intelligence Analysis*, McLean, Virginia, 2005

[16] S.Wasserman, K.Faust: *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994

[17] J. Robb. Scale-free terrorist networks: Jef AllbrightsWeb Files; URL: www.jefallbright.net/node/view/2632, 2004.

[18] K.M. Carley, J.S. Lee, and D. Krackhardt: Destabilizing networks, *Connections*, 24(3):79 92, 2002.

[19] J.Xu, H. Chen: Intelligence and Security Informatics: First NSF/NIJ Symposium, ISI 2003, Tucson, AZ, USA, June 2-3, 2003.

[20] D.R.Wanson, DR.: Undiscovered public knowledge, *Libr. Q.*, 56(2), pp. 103–118, 1986

[21] C.Best, E. Van der Goot, and Monica de Paola: Thematic Indicators Derived from World News Reports, *Intelligence and Security Informatics, Lecture Notes in Computer Science*, Springer, Berlin, 2005

[22] C.Fellbaum (ed.): *WordNet An Electronic Lexical Database*, MIT Press, 1998

[23] H.Tanev and B.Magnini: Weakly Supervised Approaches for Ontology Population, *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, Trento, Italy, 2006

[24] J.P. Sierra, A.P. Sierra, A. Gomez Perez and M. Fernendez Lopez: Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment, *IEEE Intelligent Systems*, Jan/Feb. 1999, 37-46

[25] G. Schreiber, M. Dean: OWL Web Ontology Language Reference, tech. report, *World Wide Web Consortium (W3C) Recommendation*, 10 February 2004, Available: http://www.w3.org/TR/owl-ref/

[26] P.O.Wennerberg, H.Tanev, J.Piskorski, C. Best: Ontology Based Analysis of Violent Events. IEEE ISI 2007: 373

[27] E.Prudhommeaux and A.Seaborne: SPARQL Query Language for RDF, 2004 *W3C, Working Draft*, Available: http://www.w3.org/TR/rdf-sparql-query/

[28] D.Brickley and R.V.Guha: RDF Vocabulary Description Language 1.0, *RDF Schema World Wide Web Consortium (W3C) tech. report*, February 2004,Available: http://www.w3.org/TR/2004/REC-rdf-schema-20040210

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-323

Filtering Multilingual Terrorist Content with Graph-Theoretic Classification Tools

Mark LAST
Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel. Email: mlast@bgu.ac.il

Abstract. Since the web is increasingly used by terrorist organizations, the ability to automatically detect multilingual terrorist-related content is extremely important. In this chapter, we present an efficient detection methodology based on the recently developed graph-based web document representation models. Evaluation is performed on real-world corpora in English and Arabic languages.

Keywords. Text mining, web content mining, document categorization, graph theory, terrorism informatics.

Introduction

Terrorists have quickly learned to use the Internet as an accessible and cost-effective information infrastructure. Secure and non-secure web sites, online forums, and file-sharing services are routinely used by terrorist groups for spreading their propaganda, recruiting new members, communicating with their affiliates, and sharing knowledge on forgery, explosive preparation, and other "core" terrorist activities. Thus, according to MSNBC News Services [1], a Pentagon research team was monitoring more than 5,000 Jihadist Web sites back in May 2006. The current number of known terrorist sites and active extremist forums is so large and their URL addresses are so volatile that a continuous manual monitoring of their multilingual content is definitely out of question. Moreover, terrorist web sites often try to conceal their real identity, e.g., by masquerading themselves as news portals or religious forums. This is why *automated detection methods* are so important in the research of the Internet misuse by terrorists. Particularly, there is a need of effective *filtering rules* for accurate and fast identification of real terrorist content associated with various terrorist groups.

Most web document categorization methods are based on the vector-space model of information retrieval. This popular method of document representation does not capture important structural information, such as the order and the proximity of word occurrence or the location of a word within a document. It also makes no use of the markup information that can be easily extracted from the web document HTML tags. One may expect that a representation that contains more information about a document should increase the accuracy of classification methods.

The graph-theoretic web document representation model introduced in [2] has the ability to capture important structural information contained in the document and its HTML tags. This model has been reported to outperform the vector-space representation using several instance-based classification algorithms [3]. However, the computational complexity of such algorithms is relatively high, which makes them a poor choice for online categorization of massive web document streams. Since the eager (model-based) classifiers cannot work with the graph-based representation directly, we have introduced in [4] a *hybrid approach* to web document representation, built upon both graph and vector space models, thus preserving the benefits and overcoming the limitations of each. In hybrid representation models, terms (discriminative features) are defined as subgraphs selected to represent a document already converted into a graph form. Three optional subgraph selection procedures include the Hybrid Naïve, Hybrid Smart, and Hybrid Smart with Fixed Threshold algorithms.

In this chapter, we present two case studies performed on two corpora of web documents in Arabic and English languages, respectively. In the first case study, we try to classify real-world web documents into two categories: *terrorist* and *non-terrorist*[1]. The relevant corpus consists of 648 Arabic documents where 200 belong to Palestinian terrorist web sites and 448 to popular news web sites that are not related to any terrorist organization. In the second case study, we try to identify the *source* of terrorist web documents using a corpus of 1,004 English documents obtained from a Hezbollah web site and a Hamas web site. Making a distinction between the content provided by Hezbollah and Hamas is a non-trivial task, since these two Jihadi organizations are known to have close ties with each other resulting from their common cause and ideology. The results of both case studies demonstrate that the hybrid methods outperform, in most cases, existing approaches in terms of classification accuracy, and in addition, achieve a significant reduction in the classification time.

The chapter is organized as follows. In Section 2, we briefly describe the graph-based methodology for content representation and filtering. Two case studies based on collections of authentic web documents in English and Arabic languages are presented in Section 3 and some conclusions are drawn in Section 4.

1. Graph-Based Methodology for Representation and Filtering of Web Documents

1.1. Graph Based Representations of Web Documents

In traditional information retrieval techniques, single words are used as terms. This method is called a 'set' or 'bag-of-words' [5] and it is widely used in document categorization studies and applications. According to this approach, the vocabulary is constructed from either all or *N* most weighted words that appear in the training set documents. Though this representation is most often used for information retrieval tasks, its limitations are obvious. The 'bag-of-words' does not capture important structural information, such as the ordering and the proximity of term occurrence or the location of a term within a document. Moreover, the vector-space models, which were developed for representation of plain text documents, do not make any use of the meta-tags present in any HTML document.

The *n*-gram language models have also been used for various text classification tasks including authorship attribution, language identification, and topic detection (e.g.,

[1] The terrorist organizations mentioned in this chapter are included in the list of U.S.-Designated Foreign Terrorist Organizations, which is periodically updated by the U.S. Department of State, Office of Counterterrorism.

see [6] and [7]). An *n*-gram is simply a consecutive sequence of characters or words of a fixed window size *n*. The authors of [6] have enhanced the classical Naïve Bayes Classifier model by forming a Markov chain of consecutive attributes. They have experimented with various character and word level models where the order *n* was limited to the values of eight and four, respectively. However, on the topic detection task in a large collection of English documents, the absolute accuracy improvement vs. the state-of the-art text classification methods was quite marginal: at most 0.5% using the word level and at most 1.5% using the character level. The results of another study [7] have suggested that using bigrams *in addition to* unigrams can improve the recall of some document categories.

The Graph-Theoretic Web Document Representation Technique introduced in [2] has the ability to capture important structural information hidden in the document and its HTML tags. It has been reported to outperform the vector-space model using several classification algorithms [3]. This advanced representation model is briefly described below.

All graph representations proposed in [2] are based on the adjacency of terms in an HTML document. Thus, under the *standard method*, the most frequent unique terms (keywords) appearing in the document become nodes in the graph representing that document. Distinct terms (stems, roots, lemmas, etc.) can be identified by a stemming algorithm and other language-specific normalization techniques that are also used with the vector-space models. Each node is labeled with the term it represents. The node labels in a document graph are unique, since a single node is created for each distinct term even if a term appears more than once in the text. Second, if a word *a* immediately precedes a word *b* somewhere in a "section" *s* of the document, then there is a directed edge from the node corresponding to term *a* to the node corresponding to term *b* with an edge label *s*. The ordering information is particularly important for representing texts in languages like English and Arabic, where phrase meaning strongly depends on the word order[2]. An edge is not created between two words if they are separated by certain punctuation marks (such as periods). Sections defined for the standard representation are: *title*, which contains the text related to the document's title and any provided keywords (meta-data); *link*, which is the anchor text that appears in hyper-links on the document; and *text*, which comprises any of the visible text in the document (this includes hyperlinked text, but not the text in the document's title and keywords). Graph representations are language-independent: they can be applied to a normalized text in any language.

The second type of graph representation is a *simple* representation. It is basically the same as the standard representation, except that we look at only the visible text on the page (no title or meta-data is examined) and we do not label the edges between nodes. Thus we ignore the information about the "section" of the HTML document where the two respective words appear together. Under the *n-distance* representation, there is a user-provided parameter, *n*. Instead of considering only terms immediately following a given term in a web document, we look up to *n* terms ahead and connect the succeeding terms with an edge that is labeled with the distance between them. The

[2] In some other languages, like Russian, the role of each word in a phrase is determined mainly by its *inflection* rather than its position. For example, the phrases *terrorist ubil soldata* ("terrorist killed soldier") and *terrorista ubil soldat* ("soldier killed terrorist") have an opposite meaning though the order of words is the same.

n-simple distance is identical to *n*-distance, but the edges are not labeled, which means we only know that the distance between two connected terms is not more than *n*. The *absolute frequency* representation is similar to the simple representation (adjacent words, no section-related information) but each node and edge is labeled with an additional frequency measure. Finally, the *relative frequency* representation is the same as the absolute frequency representation but with normalized frequency values associated with the nodes and edges.

Available distance measures between two graphs, such as MMCSN Measure [2], allow us to classify graphs with some distance-based *lazy algorithms* like *k*-Nearest Neighbors. The computational complexity of such algorithms is relatively high, which makes them a poor choice for real-time categorization of massive web document streams. On the other hand, we cannot induce a classification model from a graph structure using available data mining algorithms, which need a feature table as input for the induction process. In the next sub-section, we present the hybrid method of feature extraction designed specifically for the model-based classification task with documents represented by graphs.

1.2. Web Document Categorization with the Hybrid Approach

The hybrid methodology is based on the graph document representation described in the previous section. In the hybrid representation methods, terms (discriminative features) are defined as subgraphs selected to represent a document already converted into a graph form. Since all possible subgraphs in a document graph cannot be taken as attributes, some subgraph selection criteria and techniques need to be applied. In [4], three optional subgraph selection procedures are proposed, called Hybrid Naïve, Hybrid Smart, and Hybrid Smart with Fixed Threshold. We use the FSG (Frequent Subgraph Generation) algorithm [8] for frequent subgraph extraction with all selection methods.

The *Naïve* method is based on a simple postulation that a feature explains the category best if it appears frequently in that category disregarding its frequency in other categories. All graphs representing the web documents are divided into groups by class attribute value (for instance: *terrorist* and *non-terrorist*). A frequent sub-graph extraction algorithm is then applied to each group using a frequency threshold value of t_{min}. Every subgraph more frequent than t_{min} in a given group is selected by the algorithm to be a term (discriminative feature) and stored in the vocabulary. All obtained groups of subgraphs (discriminative features) are combined into one set.

If a sub-graph *g* is frequent in more than one category, it will be chosen as a feature by the Naïve method though it cannot make an effective distinction between documents belonging to those categories. The *Smart* extraction method has been developed to overcome this problem. According to the Smart method, CR_{min} (minimum classification rate) is defined by the user and only sub-graphs with *CR* (Classification Rate) higher than CR_{min} are selected as terms and entered into the vocabulary. The *CR* measure definition in [4] implies that the Smart method selects subgraphs, which are more frequent in a certain category than in other categories. Under the *Hybrid Smart with Fixed Threshold* approach to term extraction we specify the minimal classification rate CR_{min} as well as the minimal frequency threshold t_{min} in order to select subgraphs that are frequent in a specific category *and* more frequent than in other categories. The empirical evaluation in [4] shows that hybrid representation models are considerably faster than the 'bag-of-words' models, while usually outperforming them in terms of predictive accuracy.

2. Case Studies

2.1. Case Study 1: Identification of Terrorist Web Sites in Arabic

2.1.1. About the Document Collection

In this case study originally presented in [9], we try to classify real-world web documents into two categories (Boolean classification approach): *terrorist* and *non-terrorist*. Our collection consists of 648 Arabic documents where 200 belong to Palestinian terrorist web sites and 448 to non-terrorist categories. The collection vocabulary contains 47,826 distinct Arabic words (after normalization and stop word removal). Non-terrorist documents were taken from four popular Arabic news sites:

- www.aljazeera.net/News
- http://arabic.cnn.com
- http://news.bbc.co.uk/hi/arabic/news
- http://www.un.org/arabic/news.

Terror content documents were downloaded from http://www.qudsway.com and http://www.palestine-info.com/, which are associated with Palestinian Islamic Jihad and Hamas, respectively according to the SITE Institute web site (http://www.siteinstitute.org/). A human expert, fluent in Literary Arabic, has manually chosen 100 pages from each web site and labeled them as terror based on the entire *content* of each document rather than just occurrence of any specific keywords.

2.1.2. Preprocessing Arabic Documents

Text analysis of the Arabic language is a major challenge, as Arabic is based on unique grammar rules and structure, very different from the English language [10]. The first stage in text analysis is term extraction. We have defined a subset of Arabic characters in the Standard Unicode Table to be considered by the text analysis tool. The extracted terms are later stored in a data structure (array, hash table) which is called "term vocabulary". We tend to make the vocabulary as small as possible to improve run-time efficiency and data-mining algorithms accuracy. This is achieved by normalization and stop word elimination, which are standard dimensionality reduction operations in information retrieval.

Our normalization process for Arabic included the following simple rules:

- Normalizing orthographic variations (e.g., convert the initial Alif Hamza أ to plain Alif ا)
- Normalize the feminine ending, the Ta-Marbuta ة, to Ha ه
- Removal of vowel marks
- Removal of certain letters (such as: Waw و, Kaf ك, Ba ب, and Fa ف) appearing before the Arabic article THE (Alif + Lam ال)
- Removal of pre-defined stop words in Arabic. The list was compiled by an Arabic language expert.

2.1.3. Experimentation and Evaluation of Results

In order to evaluate our classification approach we used the C4.5 decision-tree classifier [11]. Decision tree models are widely used in machine learning and data mining, since they can be easily converted into a set of humanly readable if-then rules [12,13]. The goal was to estimate classification accuracy and understand how it is affected by user-defined parameters such as document graph size *N*, t_{min} in case of the Naïve and CR_{min} in case of the Smart approach. The document graph size was limited to 30, 40, 50 and 100 nodes.

We used *ten-fold cross validation* method to estimate classification accuracy. According to this method, the training set is randomly divided into ten parts with an approximately equal number of items. Then a classification algorithm is executed ten times where each time one different part is used as a validation set and the other nine parts as the training set. The percentage of correctly classified documents is reported as the classification accuracy rate. According to the experimental results presented in [9], the accuracy rate varied between 95.5% and 98.5% for both Naïve and Smart approaches.

Using the Smart method, the maximum classification accuracy was obtained with 100 nodes graph and the minimum classification rate CR_{min} of 1.25. The resulting decision tree is shown in Figure 1. The tree contains five binary attributes: four attributes representing single-node subgraphs (the words "The Zionist" in two forms, "The martyr", and "The enemy") and one two-node subgraph ("Call [of] Al-Quds" in the document text, standing for the alias name of the Hamas web site). This simple decision tree can be easily interpreted as follows: if *at least one* of these five terms appears in the graph representation of an Arabic web document, it can be safely classified as "terrorist". On the other hand, a document represented by a graph that contains *none* of these terms should be classified as "non-terrorist". It is noteworthy that one of the discriminative words ("The Zionist" الصهيوني) did appear in six normal documents out of 448. However, it was never used by the same normal document more than once, thus excluding it from the graph representation, which includes the most frequent terms only.

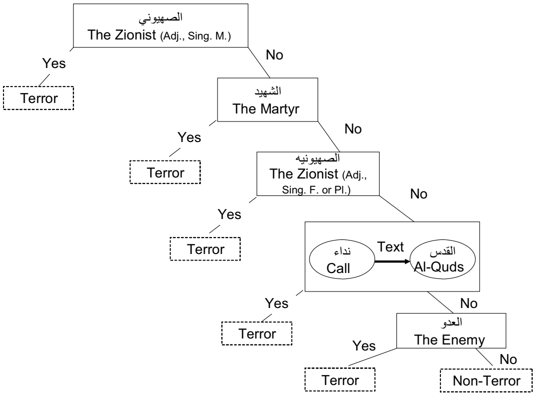


Figure 1. C4.5 Decision Tree for Classification of Web Pages in Arabic

2.2. Case Study 2: Categorization of Terrorist Web Sites in English

2.2.1. About the Document Collection

In this case study we try to identify the *source* of terrorist web documents. Our collection consists of 1,004 English documents obtained from the following two sources:

- 913 documents downloaded from a Hezbollah web site (http://www.moqawama.org/english/). These documents contain 19,864 distinct English words (after stemming and stop word removal).
- 91 documents downloaded from a Hamas web site (www.palestine-info.co.uk/am/publish/). These documents contain 10,431 distinct English words.

Both organizations are located in the Middle East with Hezbollah based in Lebanon and Hamas operating from the Palestinian Authority territory. Making a distinction between the content provided by Hezbollah and Hamas is a non-trivial task, since these two Jihadi organizations are known to have close ties with each other resulting from their common cause and ideology.

2.2.2. Experimentation and Evaluation of Results

In this case study, we also used the C4.5 decision-tree classifier [11], the Hybrid Smart approach, and the maximum graph size of 100 nodes. Classification accuracy was estimated with the ten-fold cross validation procedure. The optimal trade-off between the classification accuracy (99.10%) and the tree size (9 nodes only) was obtained with $CR_{min} = 0.55$.

The resulting decision tree is shown in Figure 2. The tree contains four binary attributes: two attributes representing single-node subgraphs (the words "Arab" and "PA" – Palestinian Authority) and two two-node subgraphs (a hyperlink to the topic of "Zionist Terrorism" and the expression “Holy Land” in the document text). This simple decision tree can be interpreted as follows: if *at least one* of these four terms appears in the graph representation of an English document coming from one of these two Web sites, it can be safely labeled as "Hamas". On the other hand, a document represented by a graph that contains *none* of these terms should be labeled as “Hezbollah".

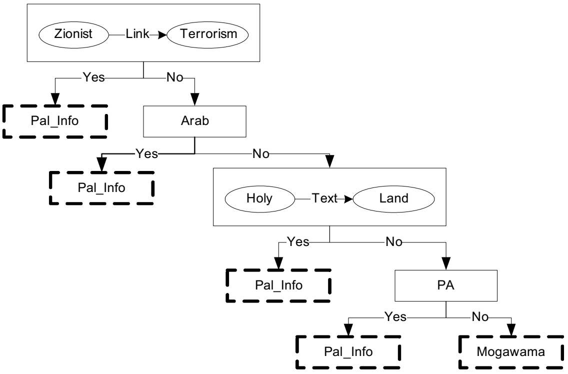


Figure 2. C4.5 Decision Tree for Classification of Terrorist Web Pages in English

3. Conclusions

In this chapter, we have demonstrated a multilingual document classification methodology, which can help us to automatically identify and filter terrorist content on the WWW. The proposed approach is utilizing the novel, graph-theoretic representation of web documents combined with the hybrid classification techniques. It was demonstrated on collections of real-world web documents in Arabic and English using the C4.5 classification algorithm. The results of both case studies show that the hybrid document classification methods can be effectively used for fast and accurate detection of multilingual terrorist content published by specific terrorist organizations. Finding the optimal parameters of the proposed methodology is a subject for future research. Experimentation with other graph-based document representations and categorization algorithms can also be performed. Another important research direction is developing and evaluating graph representations of web documents for additional languages.

Acknowledgments

We are grateful to Dror Magal, an expert in Arabic, for his valuable help with analysis of Arabic web sites and to graduate students Alex Markov, Slava Kiselevich, and Dani Alberg for their assistance with the execution of experiments presented in this chapter.

References

[1] *MSNBC News Services*, Pentagon Surfing 5,000 Jihadist Web Sites, Updated 8:30 p.m. ET May 4, 2006, Retrieved 22,03,2008 from http://www.msnbc.msn.com/id/12634238/, 2006.

[2] A. Schenker, H. Bunke, M. Last, A. Kandel, *Graph-Theoretic Techniques for Web Content Mining*, World Scientific, Series in Machine Perception and Artificial Intelligence, Vol. 62, Singapore, 2005.

[3] A. Schenker, M. Last, H. Bunke, A. Kandel, Classification of Web Documents Using Graph Matching, *International Journal of Pattern Recognition and Artificial Intelligence*, Special Issue on Graph Matching in Computer Vision and Pattern Recognition 18, 3 (2004), 475-496.

[4] A. Markov, M. Last, and A. Kandel, The Hybrid Representation Model for Web Document Classification, to appear in the *International Journal of Intelligent Systems* (2008).

[5] G. Salton, A. Wong, C. Yang, A Vector Space Model for Automatic Indexing, *Communications of the ACM* 18 (11) (1971), 613–620.

[6] F. Peng, D. Schuurmans, "Combining Naive Bayes and n-Gram Language Models for Text Classification", in Advances in Information Retrieval: Proc. 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14-16, pp. 335 – 350, 2003.

[7] C.-M. Tan, Y.-F. Wang, and C.-D. Lee, "The use of bigrams to enhance text categorization", Information Processing and Management, Vol. 38, 2002, pp. 529–546.

[8] M. Kuramochi, G. Karypis, An Efficient Algorithm for Discovering Frequent Subgraphs, *IEEE Transactions on Knowledge and Data Engineering*, 16, 9 (2004), 1038-1051.

[9] M. Last, A. Markov, A. Kandel, Multi-Lingual Detection of Terrorist Content on the Web, In Proceedings of the PAKDD'06 International Workshop on Intelligence and Security Informatics (WISI'06), Lecture Notes in Computer Science, Vol. 3917, Springer, 2006, 16-30.

[10] L.S. Larkey, L. Ballesteros, M.E. Connell, Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis, In Proceedings of the 25th Annual international ACM SIGIR Conference on Research and Development in information Retrieval, SIGIR '02, ACM Press, New York, 2002.

[11] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, 1993.

[12] J. Han, M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2001.

[13] T.M. Mitchell, *Machine Learning*, McGraw-Hill, Boston, 1997.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-331

Open Source Intelligence

Clive Best
Joint Research Centre
clive.best@jrc.it

Open Source Intelligence can be defined as the retrieval, extraction and analysis of information from publicly available sources. Each of these three processes is the subject of ongoing research resulting in specialised techniques. Today the largest source of open source information is the Internet. Most newspapers and news agencies have web sites with live updates on unfolding events, opinions and perspectives on world events. Most governments monitor news reports to feel the pulse of public opinion, and for early warning and current awareness of emerging crises. The phenomenal growth in knowledge, data and opinions published on the Internet requires advanced software tools which allow analysts to cope with the overflow of information. Malicious use of the Internet has also grown rapidly, particularly on-line fraud, illegal content, virtual stalking, and various scams. These are all creating major challenges to security and law enforcement agencies. The alarming increase in the use of the Internet by extremist and terrorist groups has also emerged. The Joint Research Centre has developed significant experience in Internet content monitoring through its work on media monitoring (EMM) for the European Commission. EMM forms the core of the Commission's daily press monitoring service, and has also been adopted by the European Council Situation Centre for their ODIN system. This paper will review this growing area of research using EMM as an example.

Keywords: information retrieval, media monitoring, topic tracking, web mining

1. Introduction

The term Open Source Intelligence originates from the security services and from law enforcement agencies. It refers to intelligence derived from publicly available sources of information, as opposed to closed or classified sources. The 9/11 Commission Report [1] exposed the weaknesses of traditional intelligence bodies as being too secretive and too complex. Although much information of the threat posed by Al Qaeda to the US homeland was publicly available, the intelligence services "failed to join the dots". The report also recommends implicitly the formation of an Open Source Agency and bringing the intelligence services into the information age.

Traditionally there are three intelligence sources as shown in Figure 1. SIGINT is intelligence gleaned from signal intercepts, wire taps and the like and HUMINT is intelligence from usually clandestine human sources. High levels of secrecy are placed on derived intelligence to protect the value of these sources. The last 10 years have seen enormous growth in the third OSINT area. Open sources by definition are non-classified although a report derived from pure open sources itself can be classified. One main advantage of OSINT is that information derived from it can be shared with other agencies and services of friendly countries. This then helps to improve information sharing and to reduce complexity, both elements having been criticized in the 9/11 report.

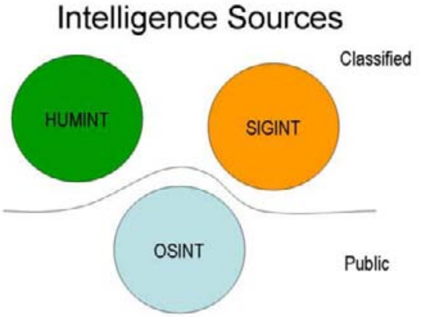


Figure 1: The three traditional intelligence sources

Military organization, including NATO, recognizes open sources as strategic, cost-effective and rapid intelligence sources [2]. However it is not just governmental bodies that are turning to open sources for current awareness and insight. Large multinational companies, banks and various industries are increasingly relying on so-called business intelligence for decision making and for protecting assets and staff. As a result, there are a growing number of OSINT-related service providers in the commercial sector who market OSINT tools. In Europe a new initiative was started in 2006 called EUROSINT Forum [3]. This forum brings together government agencies, the private sector and service providers, where the needs and processes of OSINT can be discussed with the goal of identifying technology gaps and providing training. There is also a growing research community in this area developing tools and techniques to support the OSINT process. The IEEE Conferences on Intelligence and Security Informatics [4] were started in 2005 and address related technologies.

1.1. The OSINT process

The intelligence cycle is a process which begins with a request for an intelligence study. This request comes from a senior manager in a commercial company, a minister of state, a military commander, or a director of an intelligence agency. Firstly the task is planned and information sources identified. A researcher or OSINF specialist then collects relevant information using specialized tools. The data is then indexed and processed by extracting and tagging relevant metadata. The analyst should be an expert in the relevant field with deep understanding of the problem being addressed. The analyst accesses the collected OSINF in order to author the intelligence report. Intelligence is nearly always a written document including imagery, maps etc where relevant. The web mining and information extraction research discussed in this paper concentrates on the collection and processing of OSINF. This can be broken down into five technical processes. 1) Collection: Information retrieval 2) Process: Information extraction 3) Analyse: Trend analysis/Link analysis 4) Visualise: Data visulisation 5) Collaboration.

International organizations like the UN and EU, as well as national governments, maintain so-called situation rooms for crisis management, early warning and conflict monitoring. These operations produce regular OSINT products:

1. News Flash Alerting
2. Daily news highlight reports
3. Situation updates: regular updates on an on-going crisis or conflict
4. Specific study reports: i.e. classic intelligence report

The first three products illustrate the importance of real-time news monitoring services and the systematic processing and archival of news. Such systems also form the core of media monitoring units in governments and large corporations.

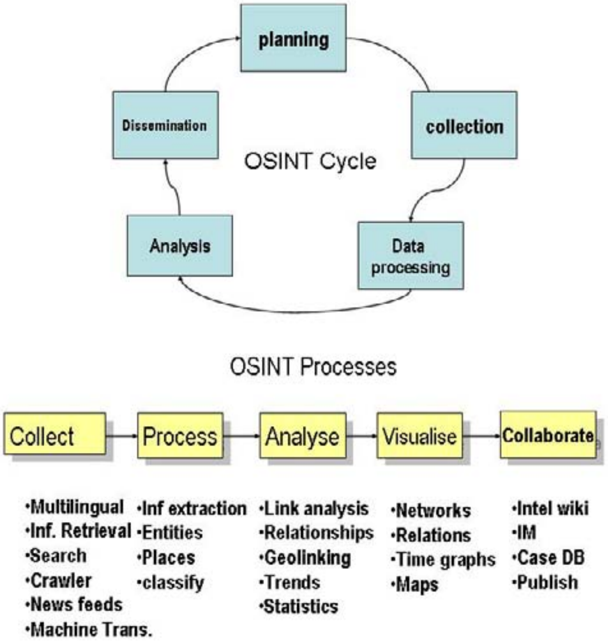


Figure 2: The OSINT cycle and process

1.2. Open Source Information

Open Source information (OSINF) is information trawled from the internet, periodicals, newspapers and radio/TV broadcasts. It is not necessarily free information and includes commercial subscription services like BBC Monitoring or Factiva, and commercial satellite imagery. The main source of information however is the Internet. In just 12 years the Internet has grown to become a major source of human knowledge. A study [5] by IDC in February 2007 estimated the total on-line digital content at $1.6x10^{20}$ bytes and that by 2010 70% of content will be user generated. The so-called Web 2 phenomenon [6] is all about on-line communication and opinion and has been driven by the widespread deployment of broadband. Broadband is enabling greater user participation and interaction unavailable during the first DOTCOM bubble. The phenomenon of Blogs and social networking sites with global participation means that opinions, local news reports and eye witness accounts are available real time.

The phenomenal growth in knowledge, data and opinions published on the Internet requires advanced software tools which allow analysts to cope with the overflow of information. Malicious use of the Internet has also grown rapidly particularly on-line fraud, illegal content, virtual stalking, and various scams. These are all creating major challenges to security and law enforcement agencies. The alarming increase in the use of the Internet by extremist and terrorist groups has emerged. The number of terrorist-linked websites has grown from about 15 in 1998 to some 4500 today. These sites use slick multimedia to distil propaganda whose main purpose is to 1) enthuse and stir up rebellion in embedded communities 2) instill fear in the "enemy" and fight psychological warfare. Anonymous communication between terrorist cells via bulletin boards, chat rooms and email is also prevalent.

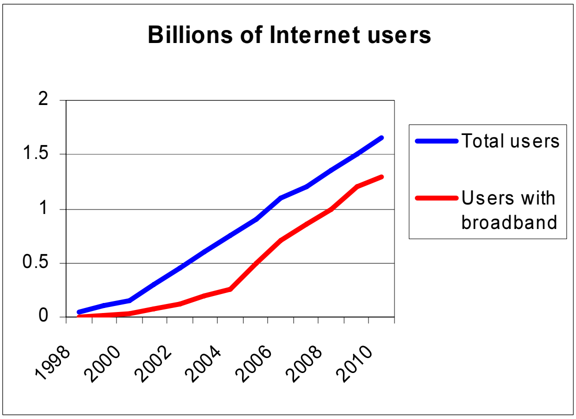


Figure 3: Worldwide Broadband Internet Users

1.3. Commercial Satellite Imagery

High resolution satellite data is now readily available commercially. This allows governments and NGOs to have unprecedented access previously the preserve of the military in elite states. However, the possibility of terrorists using such imagery to plan attacks is a real threat as identity checks on customers is difficult. Interpretation of imagery can require sophisticated processing and correction algorithms. Humanitarian applications for disaster response can rapidly perform damage assessment. JRC has been involved in damage assessment exercises for the 2005 Tsunami, the Lebanon war and Pakistan earthquakes. IAEA use satellite imagery to control proliferation at nuclear plants in signature states. Currently available satellite systems offer 60 cm IKONOS and 1 meter Quickbird producing images of stunning accuracy.

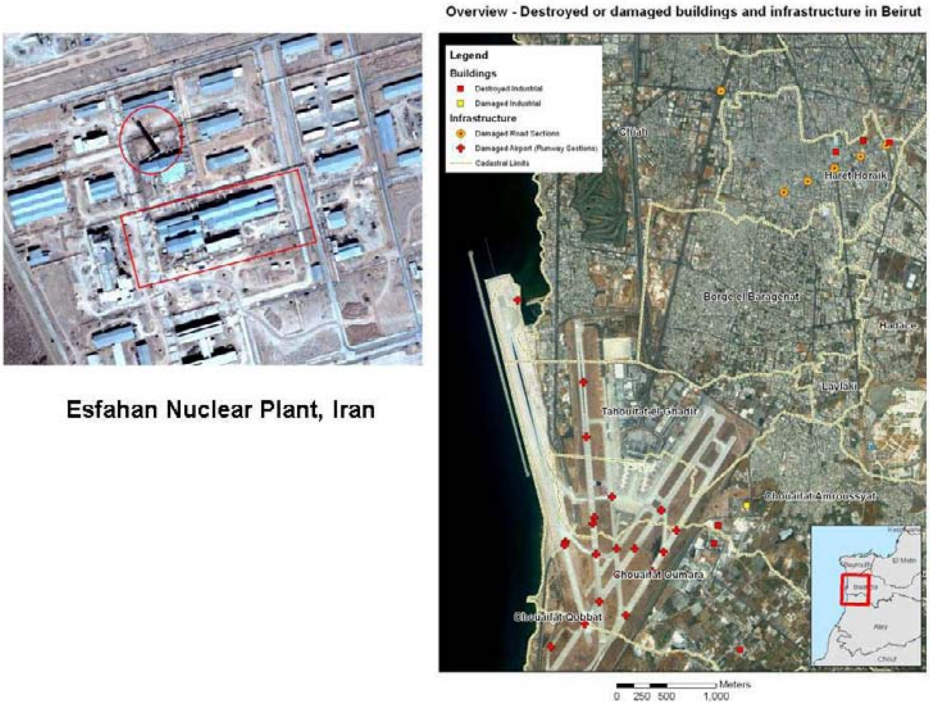


Figure 4: Two examples of image analysis using high resolution satellite Left: Esfahan Nuclear Plant, Iran showing a new Chimney and processing plant. Right: damage assessment of Beirut using before and after change detection.

Current high resolution satellites are Ikonos up to 60 cm pan-chromatic, Quickbird – 1m and Spot 2.5 m pan-chromatic. Pre- recorded data can be ordered on-line. However, for crises and damage assessment work it is necessary to task the satellite for a scene. Data needs ortho-correcting and scenes must be aligned at the pixel level for time ordered change detection [7].

1.4. Commercial Aggregators and Intelligence providers

Another important source of Open Source Information is the large commercial aggregation systems and Intelligence houses. Aggregators arrange contracts with newspapers and data providers to cover delivery and distribution rights to subscribers. Each provider feeds content to the aggregator who consequently allows customers to search their archives for data. The aggregator normally reformats all content to a standard form and indexes articles and reports by subject, source and time. These providers keep huge databases of information for on-line access. Such aggregators represent a one-stop shop for information, but they impose restrictions on the redistribution and storage of their content. Providers in this field include Lexis Nexis, Factiva and Dialog. Intelligence providers, on the other hand, generate their own reports based on open sources for specialist subjects. They are based on subscription services and include Janes Defence, Oxford Analytica and BBC Monitoring Service (multilingual news reviews)

1.5. News Aggregation systems

In recent years a number of automated news aggregation services have become available on the web. These systems monitor round the clock thousands of web based news sites and filter reports according to topics. They have a partly automated algorithm for identifying the top stories of the moment, while providing links to topic specific pages. Some systems like Google News are free, while others offer a free component and a subscription based component. This market is evolving fast as add-on analytic tools on extracted information are deployed.

The Joint Research Centre has deployed a media monitoring system for the European Commission called Europe Media Monitor (EMM). EMM is a real-time system which detects news reports in 30 languages published on Internet news sites and 15 news agencies. The EU depends on EMM for breaking news alerts, daily press reviews and early warning. Research continues on analytical tools to derive insight from news reports. A detailed description of the monitoring techniques used in EMM is given here. The other systems use similar techniques.

1.6. Web Mining Tools

The major search engines like Google are the main tools for open source information retrieval. CIA agents have once been quoted as saying 80% of intelligence is from Google. Search engines suffer from a slow update time due to the sheer size of the crawling and indexing problem. Dedicated crawlers which can monitor a user-specified set of web sites, detect and perhaps download new additions are used both by law enforcement agencies and business intelligence units.

The phenomenal growth in Blog publishing has given rise to a new research area called opinion mining. Blogs are particularly easy to monitor as most are available as RSS feeds. Blog aggregators like Technorati and Blogger allow users to search across multiple Blogs for postings. Active monitoring of Blogs applies information extraction techniques to tag postings by people mentioned, sentiment or tonality or similar [8]

2. Real Time News Monitoring

The JRC has developed the Europe Media Monitor (EMM) [9] for the European Commission. EMM scans all principal news sites in Europe and around the world up to every 10 minutes and detects new articles as they are published. The text of each article is processed and sorted into one or more of 600 topics (alerts). A breaking news algorithm detects major news stories within minutes and alerts subscribers. Inside the Commission 15 news agencies, and daily reviews of the printed press from 50 capitals, are also aggregated. The daily reviews also include reviews of newspaper clippings attached through the web authoring interface. A team of reviewers based in Brussels author a twice daily briefing newsletter and reviews of the printed press using the EMM's Rapid News Service which is a web based editorial and alerting tool. These reports are printed and sent to all Commission spokespersons and cabinets. EMM has become the EU's integrated media monitoring tool. Other agencies have adopted EMM technology and variants of it are used for specialist areas including Medical Intelligence – Medisys [10]. EMM processes about 40,000 articles in 30 languages each day. News monitoring is fundamental to Open Source Intelligence, especially where fast reliable and accurate information is needed in crisis situations. The requirements and techniques used are described in the following sections.

2.1. News Monitoring Requirements

There is a characteristic distribution of news publishing during a normal weekday. Normally very few articles are published overnight, while the main flux starts begins around 5am and slows down in the evening. This is complicated in a language like English as several time zones are involved. Then there are the rare cases when a major story breaks and publication peak can occur at any time including the nighttime. The 2005 tsunami was an example where all news sources were active out of hours. Another factor that needs to be considered in real-time monitoring is the different update rates of various sources. Some sites like CNN are updated very often while others - particularly weekly reviews - only once per day. An effective monitoring system needs to be able to cope with large variations in news fluxes as well as optimizing the scan rate for individual sites.

Users have specific interests, so a pre-defined topic detection system is needed to feed targeted news reports to that interest group. At the same time, a breaking news story can be about any subject and is therefore not predefined. Similarly a search engine of news should allow recall of any search terms in a time-ordered page ranking.

2.2. Web Site Scraping

EMM uses a technique known as headline scraping. The scraper system visits a single given news section page containing headline links to articles. The page itself is defined in HTML which describes layout, not content, and can be ill-formed with adverts and the like. Scraper first simplifies the layout and converts the page to XHTML, then for each site applies an XSLT transform to convert the content to RSS 2.0 (Really Simple Syndication). By keeping a rolling cache of RSS feeds scraper detects new articles.

EMM uses a simple REST architecture to communicate between Web applications using a queuing system to handle bottlenecks. The Grabber subsystem accesses the URL of the new article and uses a proprietary text-extraction algorithm based on an HTML scanner/parser. In a first analysis, text nodes are extracted from the HTML data stream and tagged with a synthetic 'distance' value. Based on this distance value the normal minimum distance is calculated and text nodes clustered using this minimum distance. In a second analysis a value (a so-called magic number) is attributed to each cluster based on the total length of the article, number of cluster fragments, total length of the cluster, parse tree depth with respect to structural HTML nodes (e.g. table, div, frame etc). From these numbers a minimum value is derived and all clusters with a value greater than the minimum value are considered to be part of the text. The strength of this method is that it uses the layout and style of the page itself to determine what is likely to be the most important 'block' of text.

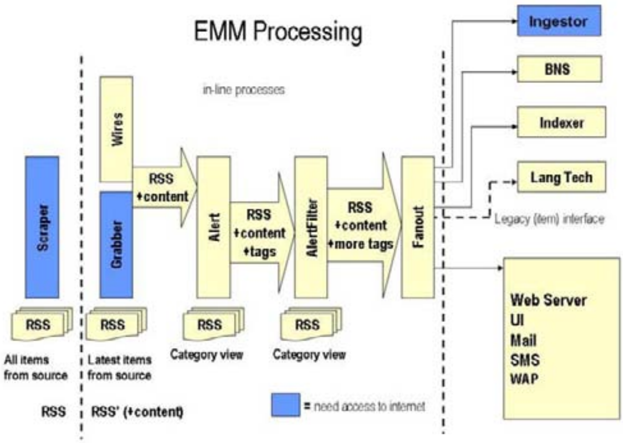


Figure 5: Overview of the EMM real time processing chain

2.3. Topic Detection

Topic Detection and Tracking has been a study area under the DARPA TIDES programme [11]. TDT aims to develop methods and algorithms for threading together related texts in streams of data - especially multilingual news. News monitoring requires two distinct tasks in this field. Firstly we want to detect articles which refer to a pre-defined subject e.g. Finance, and secondly we want to detect breaking news on any subject. Breaking News detection is discussed in the next section. Pre-defined subject detection is needed for regular updates and alerts about an area or interest, and in the European Commission concerns policy areas of the various directorates. Each incoming article in any language needs classifying into one or more topics. The EMM categorization engine is based on a proprietary and original parallel finite state machine algorithm that allows extremely fast 'feature extraction' from article text. Normally the article content is considered the 'fixed' data, and stored in a database. Categorization is then performed by running queries on the stored data. In EMM the process is turned around and the category definitions are considered the 'fixed' data. The article text 'flows' through the categorization engine and triggers the relevant categories.

Categories are defined in two ways. Firstly through weighted lists of multilingual keywords, and secondly through Boolean combinations of keywords which can include a proximity measure. Keywords can include wildcard characters to cover language dependent endings. In the first method each matched keyword contributes a weighted score, and the total sum of scores must exceed a threshold. Weights can also be negative to avoid wrong hits. The second method is a simple Boolean combination of keyword sets. One or more words in each set triggers to apply the condition and conditions are AND/NOT as illustrated in Table 1. Each category or “Alert” is hand-tuned by an editor to cover the category area concerned. For EU applications these categories can be rather broad, requiring several hundred keywords.

Weighted Definition	Boolean Definition	
alert=EuropeanParliament	alert=IrishReferendum	or=
maxArticles=50	maxArticles=50	referendum
	Proximity=5	volksabstimmung
words, threshold=20	combination	not=
european+parliament 20	or=	rugby
parl_ment%+euro% 20	ireland	football
euro%+parlament% 20	irish	
europa+parlamentet 25	iers	
europaparlamentet 25	ierland	
	irland%	

Table 1: Examples of the 2 methods for defining alerts. A single Alert can contain one or more combinations and/or a single weighted list.

2.4. Breaking News Detection

The objective of a breaking news detection algorithm is to detect as rapidly as possible a sudden flux of similar articles referring to a single undefined event. This is a different problem to pre-determined topic classification, since there is no a priori definition of what topic the articles refer to. Early warning systems require automatic alerting when a major story breaks. The first algorithm used by EMM is based on a statistical analysis of the usage of proper nouns (capitalized words) in news articles for each language. This is based on the assumption that a new story will refer to a person, a place or an organization and this assumption has proved to be correct. For each language, statistics are kept on all proper nouns found in texts, which fall outside a tuned list of stop words. Typical stop words are days of the week, newspaper titles and the like. These lists are tuned manually until false triggers are kept below 5%. A database tracks about half a million terms over a rolling 3 week period. Every 10 minutes an hourly expectation rate of occurrences over the last 3 weeks is calculated for each term. This is then compared to the actual occurrences over the last hour. Since there is always a risk of duplicates from a single source, a factor is applied to ensure independent reporting from typically three or more sources and a scaling factor is applied for each language. The score is calculated for each term as given in Equation 1.

$$S = \frac{n_t}{\bar{n}_t} \beta \frac{n_F}{N_F}$$

Equation 1: Topic score

where

S = score of topic

n_t = number of occurrences of topic in last hour

$\bar{n}_t$ = rolling average number of occurrences of topic per hour

β = multiplying coefficient

n_F = number of feeds (sources) from which the topics are derived

N_F = feed scaling coefficient

The advantage of a rolling average is that it suppresses an old breaking news story to allow increased sensitivity to other independent emerging stories. Three levels of breaking news level have been defined. The precision for detecting the highest level which represents the top 1-2 stories per week is 100% and the recall is over 95 %.

A related statistical breaking news detection concerns a sudden increase in flux of articles within one of EMM's predefined alerts. For example, a sudden rise in articles about an infectious disease may not trigger the overall breaking news system but is still of interest to epidemiologists, who may need alerting. To address this problem statistics are kept on the correlation of articles triggering a theme (topic) and a country alert (EMM has one alert defined for each world country). The alert system logs hourly counts for all theme/country correlations per hour.

$$A_i = \frac{C_i}{T_i}$$ is the normalized count of articles for country/theme last hour.

$$\overline{A} = \frac{\sum_{j=0}^{N-2} A_j}{N-1}$$ is the averaged normalised count for the last N-1 days

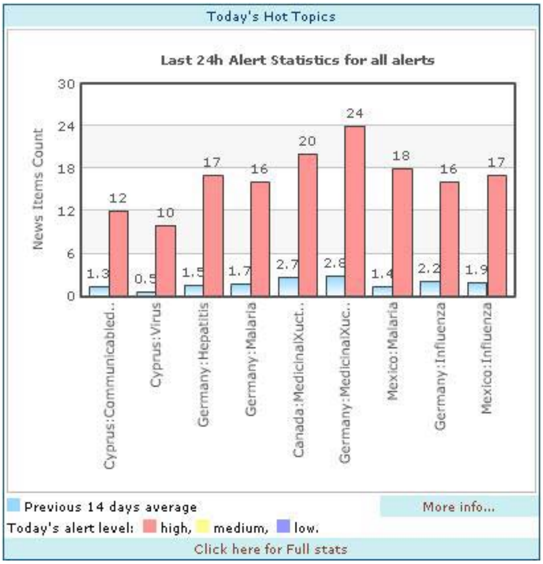


Figure 6: High Alerts for infectious diseases and countries

We assume that A has a normal distribution and then define levels of breaking news for a given theme and country according to their probability. The highest level corresponds to a random probability < 0.01.

This automated monitoring of alert levels across many topics provides a convenient way to visualize the top topics reported at the current time. The alert levels can be used to send automatic emails to specialists.

2.5. Real Time Clustering

The objective of real time clustering is to identify the current top stories and to track their evolution on a minute to minute basis. Clustering of texts is a well developed technique for topic detection and tracking [12]. Such a clustering technique has previously been applied on EMM data for the News Explorer application [13] which extracts a daily analysis of news in 18 languages. This clustering technique uses an agglomerative algorithm [8] based on hierarchical clustering. This algorithm is rather accurate, however it is still not fast enough to apply directly in real-time. Therefore a new approach was developed as described below.

The articles for each language for the last n-hours are processed. At least 200 articles are required in each language, so for major languages n=4(en, fr etc.) and minor languages n=6(ar, tk etc.). Stop words are removed from the texts automatically, based on the top 100 most frequently used words. A further reduction removes high entropy words across documents and words which appear only once. Typically this leaves 6000 unique words across 400 documents, resulting in 400 vectors in the word space.

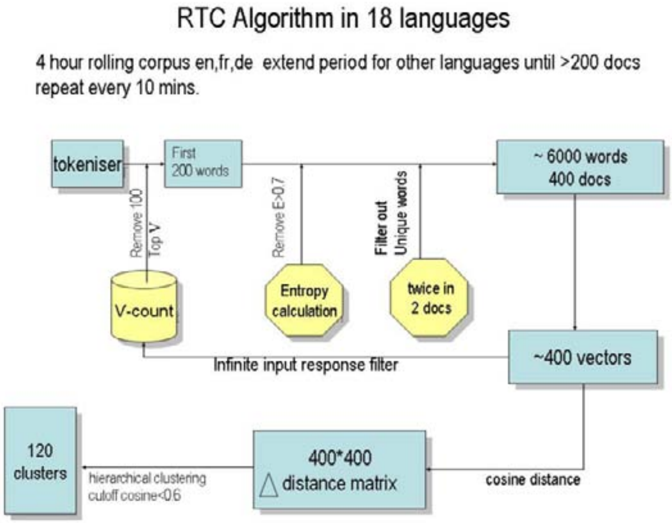
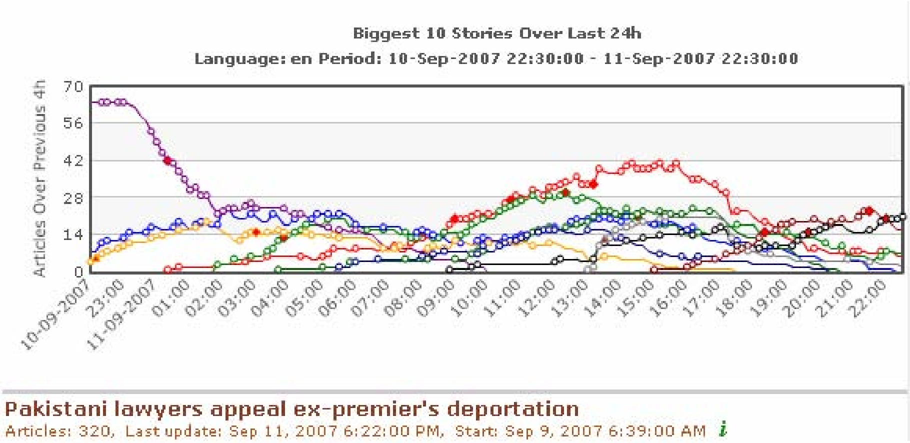


Figure 7: Schematic of Real Time Clustering

A simple hierarchical clustering algorithm is applied merging the 2 nearest vectors at each stage using a cosine distance function. Tests have shown that optimum results are when only the first 200 words in the article are used, since this avoids secondary information outside the main story, and when a cosine cutoff of 0.6 is used to distinguish stories. The full algorithm is illustrated in Figure 7. 18 languages are clustered every 10 minutes using a rolling time window of 4(6) hours. Each clustering is done independently and the central article is selected as being that closest to the centroid vector. Stories can easily be tracked across time simply through the overlapping articles with a cluster 10 minutes earlier. Stories can grow very fast within several 10 minute intervals and this is then used to trigger breaking news alerts and news update alerts. News updates are triggered when an old story suddenly receives more articles within a 10 minute period than expected. Figure 8 shows the evolution of 24 hours of news recorded automatically using this technique.



Pakistani lawyers appeal ex-premier's deportation

Articles: 320, Last update: Sep 11, 2007 6:22:00 PM, Start: Sep 9, 2007 6:39:00 AM

Figure 8: Real Time Clustering over a 24 hour period. Each story is represented by a coloured trace plotted against time where the magnitude is the number of articles in the cluster i.e. size of story.

3. News Visualisation

Information overload is a growing problem for OSINT analysts. Consequently, research focuses on techniques to visualize a summary of textual information. Information extraction techniques, entity tracking and event extraction are described in other papers in this book. Here we concentrate on techniques for visualizing news overviews and social networks based on relation extraction.

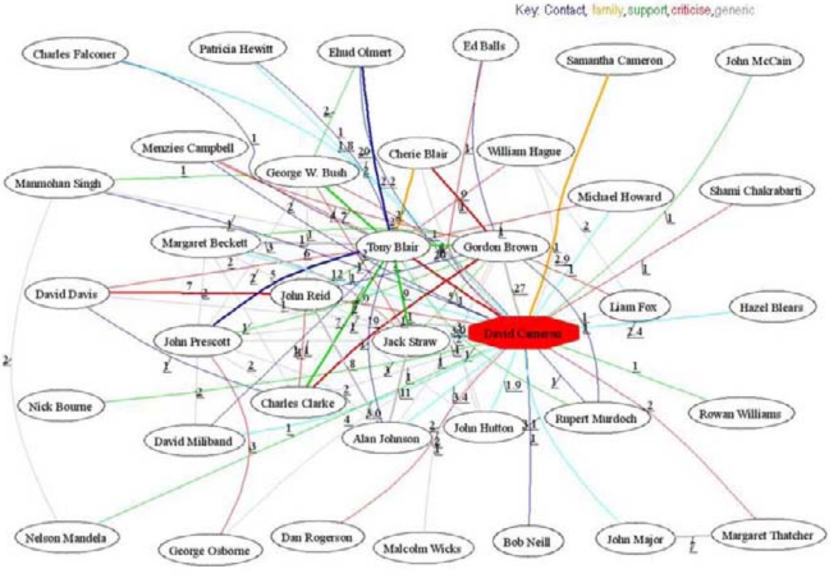


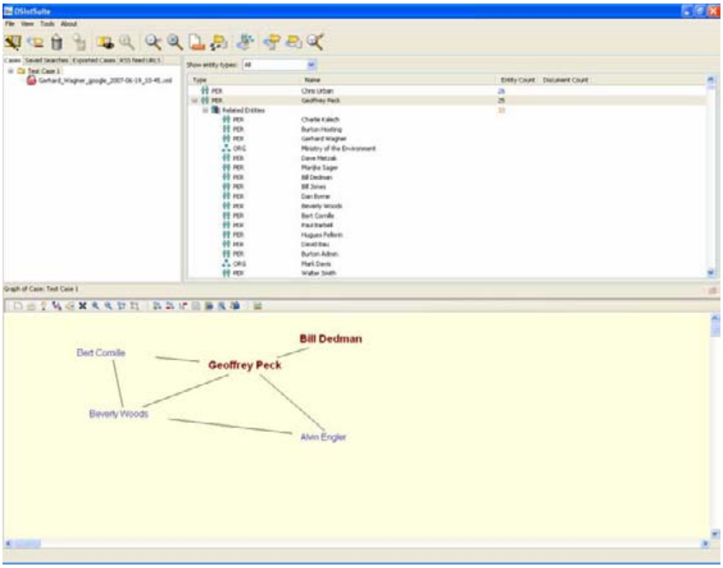
Figure 9: Social network based on relation extraction for David Cameron

News Maps [14] are intended to show country hot spots for thematic news in analogy to weather maps. They display the relative media reporting for a given time period for all countries of the world. In the case of EMM they are driven by statistical correlation data recorded by the EMM alert system. Such statistical data can be used to derive normalised indicators which “measure” the relative media coverage for a given topic and a given country. This technique has been applied to detect so-called “forgotten crises” [15] which are countries with humanitarian needs arising from famine or conflict, but under-reported in the world’s media. An analysis of such forgotten crises forms one of the bases for planning the yearly aid budget of the EU office for humanitarian aid.

Entity extraction is a well developed technique and has been applied to EMM [16]. The statistical co-occurrence of entities in texts can then be used to derive social networks. Relation extraction goes further and identifies phrase patterns linking two entities in a sentence [17]. EMM has studied three relations so far: contact (met, phoned, etc), support, criticize and family. Figure 9 shows such a derived social network.

4. OSINT Suite

Key elements of EMM have been combined into a standalone Java application for web mining. Case Management, Site monitoring, text extraction and entity extraction tools are packaged in a visual interface. A case consists of documents retrieved through Google searches, or from crawling specific sites and a database of extracted linked entities. The information retrieval components are an adaptation of the scraper/grabber from EMM and the information extraction component uses the named entity recognition tools from News Explorer. Finally an interactive link analysis tool visualizes identified relationships.



Show entity types: All

Type	Name	Entity Count	Document Count
PER	Chris Urban	26	
PER	Geoffrey Peck	25	
Related Entities		33	
PER	Charlie Kalech		
PER	Burton Hosting		
PER	Gerhard Wagner		
ORG	Ministry of the Environment		
PER	Dave Metzak		
PER	Marijke Sager		
PER	Bill Dedman		
PER	Bill Jones		
PER	Dan Byrne		
PER	Beverly Woods		
PER	Bert Cornille		
PER	Paul Barbell		
PER	Hugues Pelloni		
PER	David Bau		
PER	Burton Adrian		
ORG	Mark Davis		
PER	Walter Smith		

Figure 10: OSINT Suite user interface

5. Conclusions

Open Source Intelligence is a growing field with important applications in the security domain. Security services, law enforcement and military intelligence rely on open sources in addition to traditional classified sources. The internet now dominates information sources and the spread of global news, rumour and propaganda requires sophisticated monitoring techniques. This paper has focused mainly on the current awareness tools for on-line media monitoring as implemented by the Europe Media Monitor. The author would like to acknowledge the EMM development team and in particular the contributions of Erik van der Goot, Flavio Fuart, Ralf Steinberger, Teofilo Garcia, David Horby, Bruno Pouliquen, Hristo Tanev and Jakub Piskorsky. The satellite image analysis results are from Martino Pasereri and his group at JRC.

6. References

[1] 9/11 Commission Report, http://www.gpoaccess.gov/911/pdf/fullreport.pdf

[2] NATO Open Source Handbook, Intelligence Reader, and Intelligence Exploitation of the Internet, 2002, http://www.oss.net

[3] EUROSINT Forum, http://www.eurosint.eu

[4] IEEE International Conference on Intelligence and Security Informatics, ISI 2004,2005,2006,2007 Lecture Notes in Computer Science, Springer

[5] IDC study on Internet Broadband usage, http://www.idc.com

[6] C. Pascu, D. Osimo, M. Ulbrich, G. Turlea, JC Burgelman, “The potential disruptive impact of Internet 2 based technologies", First Monday, March 2007 http://www.firstmonday.org/issues/issue12_3/pascu/

[7] M. Pesaresi and E. Pagot, “Post-conflict reconstruction assessment using image morphology profile and fuzzy multicriteria approach on 1-m resolution satellite data, IEEE GRSS/ISPRS Joint workshop on Remote Sensing and Data Fusion over Urban Areas – URBAN/URS, Apr 2007 Paris-France

[8] Bing Liu, Web Data Mining, Chapter 11, Springer ISBN-10: 3-540-37881-2

[9] Best Clive, Erik van-der Goot, Ken Blackler, Teofilo Garcia, David Horby, 2005. Europe Media Monitor - System Description. EUR Report 22173 EN.

[10] Yangarber Roman, Ralf Steinberger, Clive Best, Peter van Etter, Flavio Fuart, David Horby. Integration of Information Retrieval with Information Extraction for Medical Intelligence. this edition, IOS Press

[11] Wayne, C., Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation, Language Resources and Evaluation Conference (LREC) 2000, pages 1487-1494.

[12] A. Jain, M. Murty, P. Flyn, Data clustering: a review, ACM Computing Surveys, Vol. 32(3), pp. 264-323, 1999

[13] Steinberger Ralf, Bruno Pouliquen, Camelia Ignat (2005). Navigating multilingual news collections using automatically extracted information, Journal of Computing and Information Technology - CIT 13, 2005, 4, 257-264. Available online at: http://cit.zesoi.fer.hr/downloadPaper.php?paper=767. ISSN: 1330-1136.

[14] Clive Best, Erik van der Goot, Ken Blackler, Teofilo Garcia, David Horby(2005) Mapping World Events, GeoiInformation for Disaster Management, ISBN 3-540-24988-5, Springer p 683.

[15] Clive Best, Erik Van der Goot, Monica de Paola, Thematic Indicators Derived from World News Reports, IEEE Conference on Intelligence and Security Informatics, ISI2005, Springer Lecture Notes in Computer Science 3495, P436-447.

[16] Steinberger Ralf & Bruno Pouliquen (2007). Cross-lingual Named Entity Recognition. In: Satoshi Sekine & Elisabete Ranchhod (eds.), Journal Linguisticae Investigationes, Special Issue on Named Entity Recognition and Categorisation, LI 30:1, pp. 135-162. John Benjamins Publishing Company. ISSN 0378-4169.

[17] Hristo Tanev (2007). Unsupervised Learning of Social Networks from a Multiple-Source News Corpus. Proceedings of the Workshop Multi-source Multilingual Information Extraction and Summarization (MMIES'2007) held at RANLP'2007. Borovets, Bulgaria, 26 September 2007.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-898-4-345

Detecting Core Members in Terrorist Networks: A Case Study

Nasrullah Memon and David L. Hicks
The European Center for Counterterrorism Research and Studies
Department of Computer Science and Engineering
Aalborg University, Niels Bohrs Vej 8, DK-6700, Esbjerg, Denmark

Abstract. This chapter focuses on the study and development of recently introduced new measures, theories, mathematical models and algorithms to support the detection of core members in terrorist networks. Specific techniques and tools are described to demonstrate their applicability to the area.

Keywords. Position role centrality, Dependence Centrality, Detecting hidden hierarchy in terrorist networks, iMiner knowledge base

1. Introduction

The events of 9/11 instantly changed American perceptions of the words “terrorist” and “network”, and the United States and other countries rapidly started to gear up to fight a new kind of war against a new kind of enemy. In conventional warfare, conducted in specific locations, it was important to understand the terrain in which the battles will be fought. In the war against terror, there is no specific location. After 9/11, we know that the battleground can be anywhere. It is now clear after 9/11 that the terrorists’ power base is not geographic; rather, they operate in networks, with members distributed across the globe. To fight such an enemy, we need to understand the new “terrain”: networks—how they are constructed and how they operate.

Advanced and emerging information technologies like investigative data mining offer key assets in confronting a secretive, asymmetric networked enemy. Investigative data mining (IDM) is a powerful tool for intelligence and law enforcement agencies fighting against terrorism [1]. Investigative data mining is a combination of data mining and subject-based automated data analysis techniques. Actually data mining has a relatively narrow meaning: the approach which uses algorithms to discover predictive patterns in datasets. Subject-based automated data analysis applies models to data to predict behaviour, assess risk, determine associations or perform other types of analysis [2].

How can we mine terrorist networks? Traditional methods of machine learning and data mining, taking a random sample of homogeneous objects from a single relation as input, may not be appropriate. The data comprising terrorist networks tend to be heterogeneous, multi-relational and semi-structured. IDM embodies descriptive and predictive modeling. By considering links (relationships between the objects), more information is made available to the mining process. Mathematical methods used in our research on IDM [1] [2][3] [4] are clearly relevant to law enforcement intelligence work and may provide tools to discover terrorist networks in their planning phase and thereby prevent terrorist acts and other large-scale crimes from being carried out. Relevant patterns to investigate include connections between actors (meetings, messages), activities of the involved actors (specialized training, purchasing of equipment) and information gathering (time tables, visiting sites).

Investigative Data Mining (IDM) offers the ability to firstly map a covert cell, and to secondly measure the specific structural and interactional criteria of such a cell. This framework aims to connect the dots between individuals and "map and measure complex, covert, human groups and organizations". The method focuses on uncovering the patterning of people's interaction, and correctly interpreting these networks assists "in predicting behavior and decision-making within the network". IDM borrows social network analysis (SNA) and graph theory techniques for connecting the dots; our goal is to propose mathematical methods for destabilizing terrorist networks after linking the connections between them.

In investigative data mining, a number of variations exist in the literature. One is known as link analysis (see for example [5] [6]). Link analysis research uses search and probabilistic approaches to find structural characteristics in the network such as hubs, gatekeepers, pulse-takers [7], or identifying potential relationships for relational data mining. Link analysis alone is insufficient as it looks at one side of the coin and ignores complex nonlinear relationships that may exist between the attributes. Another approach depends purely on visualization, such as NetMap [8]. Unfortunately, these tools that depend on visualization alone - despite being useful to provide some insight - are insufficient and rely on the user to carry out many tedious and time consuming tasks, many of which could be automated.

In addition to the previous discussion, most of the work on link analysis or network visualization ignores the construction of the hidden hierarchy of covert networks. Uncovering a relationship among or within attributes (connecting the dots) is an important step, but in many domains it is more important to understand how this relationship evolved. Hence, understanding network dynamics and evolution is needed to complete the picture. Once we understand the dynamics and evolution of these relationships and construct the hidden hierarchy, we can search for ways to disconnect the dots if and when needed. This brings about several new tasks:
(i) Subgroup detection (ii) Object classification (iii) Object dependence (iv) Detecting hidden hierarchy (v) Understanding topological characteristics.

In this chapter, we study techniques to detect core members in terrorist networks. Our goal is to understand the structure of these networks in order to assist law enforcement agencies for disrupting them.

In the remainder of this chapter, Section 2 discusses studies on detection of core members. In Section 3, we present our data collection methods and an overview of terrorist network research. In Section 4, we report and discuss our findings from the analysis using a case study. Section 5 concludes the chapter with the an examination of the implications of this research and future research directions.

2. Techniques for Detecting Core Members

In this Section we discuss various techniques to detect the core members in terrorist networks.

2.1. Subgroup Detection

One of the most common interests in analyzing terrorist networks is the search for the substructures that may be present in the network. Subgroups are subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties. In this chapter, we use a bottom-up approach for the detection of subgroups [9].

This approach begins with basic groups, and seeks to see how far this kind of close relationship can be extended. The notion is to build outward from single ties to construct the network. The substructures that can be identified by bottom-up approaches include cliques, n-cliques, n-clans and k-plexes. We discuss each concept briefly:

A *clique* is defined as a maximal sub-graph in which every member of the graph is connected to every other member of the graph. Every member is connected to n-1 others and the distance between every pair is 1. In practice, complete cliques are not very useful. They tend to overlap heavily and are limited in their size. Extensions of this idea include:

n-clique is a sub-graph in which Every person is connected by a path of length *n* or less.

n-clan is a sub-graph like as an n-clique, but all paths must be contained *inside* the group.

k-plex is a sub-graph in which every member connected to at least n-k other people in the graph (recall in a clique everyone is connected to n-1, so this relaxes that condition. For further details refer [9].

2.2. Object Classification

In traditional classification methods, objects are classified on the attributes that describe them. A particular important challenge is to classify in a large network those individuals who play key roles—such as leaders, facilitators, communications "go betweens", and so on. To understand the calculations used to single out the core members in a network, we need to discuss some measures of object classification [10]:

2.2.1. Degree Centrality

A basic measure [19]of social network analysis that turns out to be important in IDM is the degree of a node—that is, the number of other nodes directly connected to it by edges/ links. In a graph (network) describing a terrorist network, nodes of high degree represent "well connected" people, often leaders.

2.2.2. Closeness Centrality

This measure [19] indicates for each node how close it is to other nodes in a graph.
Analysts consider this measure a good indication of how rapidly information can spread through a network from one node to others. This measure relates to the closeness or the distance between nodes. A core member (central actor) can reach other actors through a minimum number of intermediary positions and is therefore dependent on fewer intermediary positions than a peripheral actor.

Suppose a terrorist organization wants to establish a new camp, for example, a human bomb training camp, such that the total distance to all persons interested to kill themselves, for a cause, in the region is minimal. This makes travelling to the camp as convenient as possible for most people who are living in that region and are willing to be used for human bombs in the near future.

2.2.3. Betweenness Centrality

The measure [19] gives each node a score that reflects its role as a stepping-stone along geodesic (shortest) paths between other pairs of nodes. The idea is that if a geodesic path from node A to node B (there may be more than one) goes through node C, then node C gains potential importance. Such nodes—or people they represent in a terrorist network—can have important roles in providing connections (for example, facilitating communications) between sets of nodes that otherwise have few other connections, or perhaps no other connections.

This measure explores an actor's ability (say for example, node C) to be "irreplaceable" in the communication of two random actors (say for example, nodes A and B). It is of particular interest in the study of destabilizing terrorists by network attacks, because at any given time the removal of maximum betweenness actor seems to cause maximum damage in terms of connectivity and average distance in a network.

2.2.4. Efficiency Centrality

Terrorist networks can be understood from a point of view of efficiency, i.e., the efficiency into propagation across the network. The network efficiency [20] *E* (*G*) is a measure to quantify how efficiently the nodes of a network exchange information.

2.2.5. Position Role Centrality

This centrality is a newly introduced measure [10] which highlights a clear distinction between followers and gatekeepers. It depends on the basic definition of efficiency as discussed above. The efficiency of a network in the presence of followers is low, in comparison to their absence in the network. This is because followers are usually less connected nodes and their presence increases the number of low connected nodes in a network, thus decreasing its efficiency

2.2.6. Dependence Centrality

This measure [17] represents how much a node is dependent on other nodes in a network. Consider a network representing a symmetrical relation, "communicates with" for a set of nodes. When a pair of nodes (say, *u* and *v*) is linked by an edge so that they can communicate directly without intermediaries, they are said to be adjacent. Suppose a set of edges links two or more nodes (*u*, *v*, *w*) such that *u* would like to communicate with *w*, using node *v*. The number of time node *u* uses node *v* to reach node *w* along a shortest path can be compared to the overall number of shortest paths between them to measure the dependence of node *u* on node *v* to communicate with node *w*. There can, of course, be more than one geodesic, linking any pair of nodes. This measure can be thought of an index of the degree to which a particular terrorist must depend upon a specific other – as a relayer of messages – in communicating with all others in a network.

2.3. Detecting Hidden Hierarchy

Terrorist networks are known as horizontal networks, i.e., they are different than organizational networks which are known as vertical networks. Detecting hidden hierarchy from terrorist networks is a novel contribution of our research. Discovering hierarchy [16] from a terrorist network is *a process of comparing different centrality values of different nodes to identify which node is more powerful, influential or worthy to neutralize than others*.

From the above definition, it is clear that we may use different centrality measures for finding a corresponding hierarchical view of a terrorist network. Similarly these measures can be used to build hierarchies to detect an organizational view of a corresponding terrorist network/ organization. Currently, experts have agreed that real destabilization is about isolating leaders from enough followers, thus disabling them from executing any terrorism plans. This idea has certainly been a central motive for investigative tools.

2.4. Understanding Topological Properties

Understanding the structure of a terrorist network is very important before applying techniques to destabilize and disrupt the network [11, 12]. If we have good knowledge of a terrorist network today then it will not be operational tomorrow**.** In the literature concerning network theory, some measures for understanding topological properties of networks are available [12]. We discuss in brief:

2.4.1. Clustering Coefficient

A network shows clustering if the probability of a pair of nodes being adjacent is higher when the two nodes have a common neighbor. The clustering coefficient of a network is defined as the average probability that two neighbors of a given node are adjacent.

2.4.2. Average Path Length

The distance between two nodes is defined as the number of edges along the shortest path connecting them. The average path length is a measure of how scattered a network is.

2.4.3. Degree Distribution

This measure shows the probability distribution of degrees in a network. The degree distribution is a function describing the total number of nodes in a network with a given degree.

3. Network Research and Terrorism

After the attacks of 9/11, the academic world has increased the attention paid to network research for terrorism as a result of public interest. The network analysis of terrorist organizations can be divided into two classes: the data collectors and data modelers.

3.1. Data Collectors

Data collection is difficult for any network analysis because it is difficult to create a complete network. It is not easy to gain information on terrorist networks. It is a fact that terrorist organizations do not provide information on their members and the government rarely allows researchers to use their intelligence data [10]. A number of academic researchers focus primarily on data collection on terrorist organizations, analyzing the information through description and straightforward modeling.

By harvesting terrorist information from Web [13], we have also developed a large knowledge base of terrorist attacks that occurred in the past. The focus of the knowledge base we have developed is the agglomeration of publicly available data and the integration of the knowledge base with an investigative data mining software prototype. The main objective is to investigate and analyze terrorist networks to find hidden relations and groups, prune datasets to locate regions of interest, find key players, characterize the structure, trace a point of vulnerability, detect the efficiency of a network and to discover the hidden hierarchy of non-hierarchical networks.

3.2. Data Modelers

Complex models have been created that offer insights into theoretical terrorist networks and looked at how to model the shape of a terrorist network when little information is known through predictive modeling techniques based on inherent network structures.

A common problem for the modelers is the issue of data. Any academic work is only as good as the data, no matter the type of advanced methods used. Modelers often do not have the best data upon which it is based, as they have not collected individual biographies as was done in [21] and do not have access to classified data. Many of the models were created data-free or without complete data, and do not fully consider human and data limitations [10].

On the other hand, in our research we developed mathematical models for further analysis of terrorist networks [10]. These models are implemented in a software prototype, iMiner, which is integrated with a knowledge-base for terrorist events that have occurred in the past.

4. Case Study

Figure 1 shows an example of a terrorist network, which maps the links between terrorists involved in the tragic events of September 11, 2001. This graph was constructed by Valdis Krebs [14] using the public data that were available before, but collected after the event. Even though the information mapped in this network is by no means complete, its analysis may still provide valuable insights into the structure of a terrorist organization. This graph is constructed based on original and adding metadata.

According to Krebs [14] analysis, this network had 62 members in total, of which 19 were kidnapers, and 43 were assistants: organizers, couriers, financiers, scouts, representatives, coordinators, counterfeiters, *etc.* Allen [15] found that successfully functioning large networks typically comprise 25-80 members, with an optimal size between 45 and 50. A close match exists between the results of Allen's analysis of collaborating networked groups and this particular example of a terrorist group.

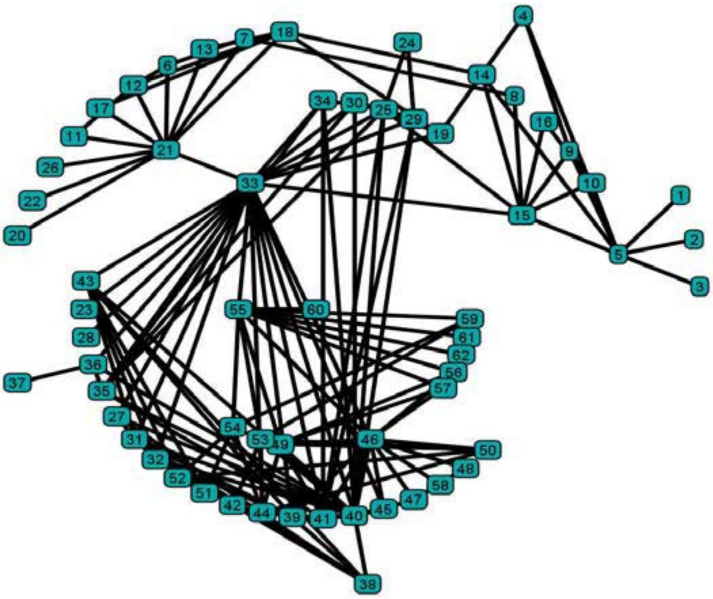


Figure 1: 9/11 Terrorists Network

The results of the measures discussed in Section 2 of this chapter are discussed:

4.1. Subgroup Detection

In this case study we discuss the use of the four concepts: cliques, n-cliques, n-clans, and k-plex. The dataset, which describes the network from Figure 1, has been analyzed for the presence of each them substructure types. The statistics from the results are listed in Table 1.

Each node represents a specific person from the dataset, so the number of nodes should be the same for all concepts. Each substructure concept generated a different number of groups: for example, the *n-clan* concept generates 22 groups while the *k-plex* concept generates 493 groups. The *n-clique* concept generates 38 groups and the *clique* concept generates 41 groups. For each of the concepts the maximum size and minimum size of a group has also been collected and shown in Table 1. The statistics indicate that even with a relatively small dataset a huge number of groups could be generated. The groups generated are analyzed, in order to identify the best candidate nodes for destabilizing the specific network.

Table 1. Statistics from the results

	Groups' Total Number	Groups' Maximum Size	Groups' Minimum Size
Clique	41	6	3
N-Clique	38	23	5
N-Clan	22	23	5
K-Plex	493	7	3

Figure 2 shows how many groups each member participates in, using respectively the *clique, n-clique*, *n-clan* and *k-plex* concepts. As we can see some of the members participate in many groups while other members participate in few groups. We say that a member that participates in many groups, compared to the total number of groups, has a *high participation index*, while a member that participates in few groups, compared to the total number of groups has a *low participation index*. The participation index is defined as participation of a particular member in different the groups generated by the various concepts of subgroup detection.

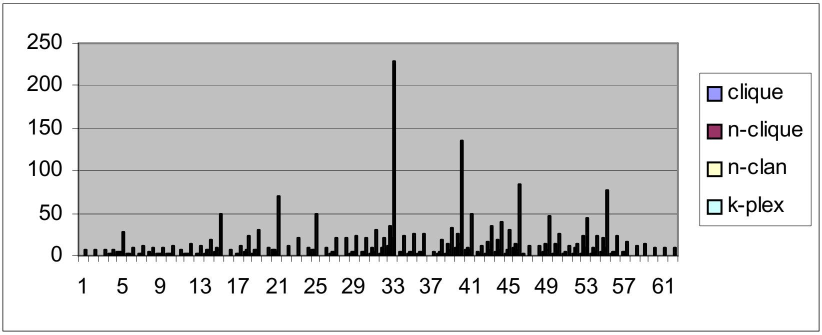


Figure 2: The participation of the members of the 9/11 terrorists network in various groups using the concepts Clique, n-clique, n-clan and k-plex.

For example, consider the member Mohamed Atta (node 33) in the matrix generated using the k-plex concept, has participated in 230 groups and the total number of groups is 493. This gives a participation index equal to 230/493, approximately 0.5. If the participation index is closer to 1, it means that member has participated in most of the groups, and if the participation index is closer to 0, it means that the member's participation is negligible. From the variation seen in the participation index we conclude that the choice of concept has an important influence on the participation index. It seems like using the concepts n-clique and n-clan results in higher participation indices, while the concepts clique and k-plex results in lower participation indices.

The three members described in Table 2, can roughly be seen as a picture of archetypes or roles in the network. In most cases a member is not 100 percent an archetype, but a combination of the three types. What type a member will match best in a specific situation will also be dependent on other factors, e.g. the phase of the operation being conducted by the network.

Table 2: Participation Index

	Member 33	**Member 37**	**Member 55**
Clique	0.293	0	0.098
N-Clique	0.947	0.026	0.553
N-Clan	0.909	0.045	0.455
K-Plex	0.467	0.008	0.152

The archetypes are named brokers (gatekeeper, representative or coordinator), leaders and followers. Brokers encompass members working with logistics, communications, etc. Leaders encompass leaders at all levels, using military terms this means officers. Followers encompass the members that can be compared to the infantry in military terms.

4.2. Object Classification

Inspection of this network by standard measures of network structure reveals firstly its low connectedness. A member of this network holds only 4.9 connections with other members on average (also known as degree centrality), which means that average members were rather isolated from the rest of the network. The density (which is defined as the number of actual links divided by the number of possible links) of this network is only 0.08, meaning that only 8% of all possible connections in the network exist.

In spite of low connectedness, however, the nodes of this network are relatively close. The average closeness of nodes is 0.35. Betweenness as stated above is another important measure in SNA and it indicates a node's importance for communication among other nodes. The average betweenness of this network is 0.032, indicating relatively high average redundancy. However the betweenness of 40 nodes is in fact less than 1% and only 6 nodes have betweenness higher than 10%. These 6 nodes are critical for information flow, especially one in particular with a betweenness of almost 0.589, meaning that almost 60% of communication paths among other nodes pass through this central node. The node represents *Mohamed Atta* (node # 33); the leading organizer of the attack whose central position in the network is confirmed by other centrality indicators as well (For further details see Figure 3).

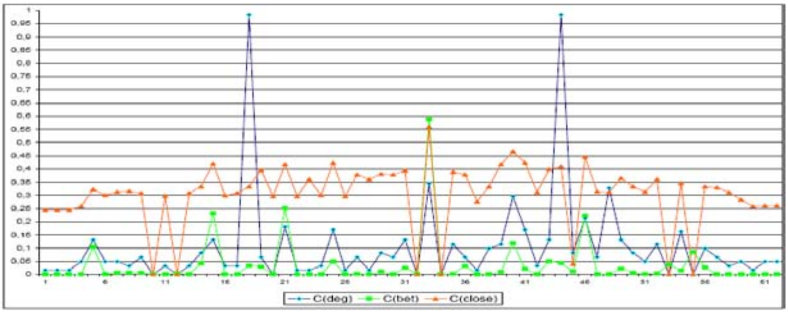


Figure 3: Terrorist Network's Neighbourhood

We applied the above mentioned measures (described in subsection 2.2.4-2.25) in the network of terrorists involved in September 11, 2001 (as shown in Figure 1). The results are depicted in Figure 4. The results show node 33 (*Mohamed Atta*) as a key player in the plot. The position role centrality of this node is higher than all nodes which indicates that this node played an important role in the plot and worked as gatekeeper; and when this node is removed the efficiency of the graph is decreases from *0.395* to *0.32*. This clearly identifies the importance of this node in the network.

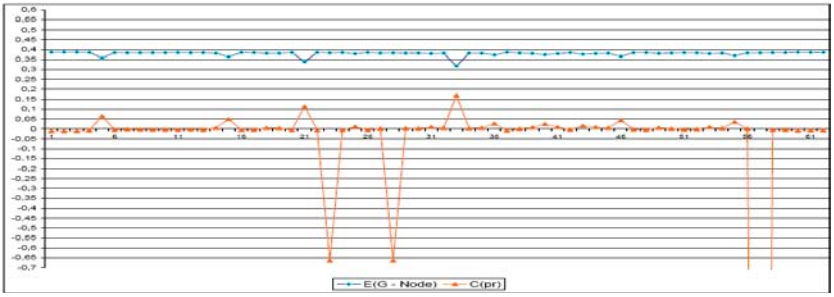


Figure 4: The efficiency of the original network $E\ (G) = 0.395$. The removed node is shown on x-axis. The efficiency of the graph once the node is removed is shown as $E\ (G - v_i)$. The newly introduced measure position role centrality is shown as C(pr).

By examining the 9/11 network, we found that the *dependence centrality* of the node 33 is very low, showing that this person was not depending on the other members of the network. But most of the network members are depending on this node. This is clearly an indication of the importance of node 33 in the network.

4.3. Detecting Hidden Hierarchy

Using the algorithms for detecting hidden hierarchy [16] of non-hierarchical terrorist networks, we tested the network of terrorists involved in the 9/11 tragic events and the results are depicted in Figure 5. Node 33 is found as the commanding authority in the network and shows that the results achieved are in excellent agreement to reality.

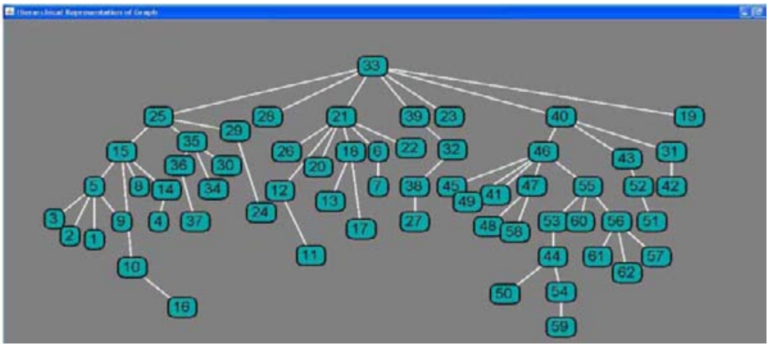


Figure 5: Hidden Hierarchy in 9/11 network.

4.4. Understanding Topological Properties

We found that members in the 9/11 terrorist network are extremely close to their leaders. The terrorists in the network are on average only 1.79 steps away from Mohamed Atta, meaning that Mohamed Atta's command can reach an arbitrary member through only two mediators (approximately). Despite its small size (62), the average path length is 3.01, Information flows quicker, with less distortion, and Mohamed Atta is more involved. The other small-world topology characteristic, a high clustering coefficient, is also present in this network. The clustering coefficient of this network is 0.49, significantly high.

In addition, this network is a scale-free system, i.e., the degree distribution decays much more slowly for small degrees than for that of other types of networks, indicating a higher frequency for small degrees.

The distribution of degrees of nodes is particularly interesting. Degrees of nodes are exponentially distributed: the degree of most of the nodes is small, while few nodes have high degree (see Figure 6). This property characterises the so called scale free networks [11]. Scale free networks form spontaneously, without needing a particular plan or interventions of central authority. Nodes that are members of the network for a longer time, that are better connected with other nodes, and that are more significant for a functioning network, are also more visible to new members, so that the new members spontaneously connect more readily to such nodes than other, relatively marginal ones.

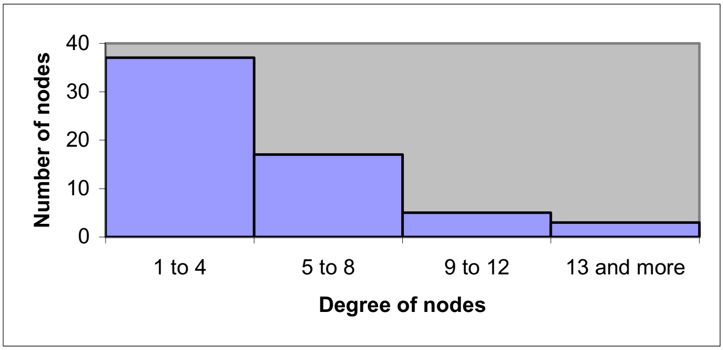


Figure 6. Distribution of degree of nodes in 9/11 network

On the pattern of scale free networks, the Al Qaeda's Training Manual (2001) [18] states: "The cell or cluster methods should be organized in a way that a group is composed of many cells whose members do not know each other, so that if a cell member is caught, other cells would not be affected, and work would proceed normally".

5. Conclusion and Future Recommendations

This chapter sought to describe new theories and measures for detecting core members in terrorist networks. The measures could be useful for law enforcement agencies to disrupt the effective operation and growth of these networks or destroy some terrorist cells entirely. Although these adversaries can be affected in a number of ways, this chapter focuses upon capturing/ eradicating a terrorist network's most influential persons or finding susceptible points of entry and conveying information or influence that contribute to winning the war against terrorism.

There remain a number of research opportunities in this new area of intelligence and security informatics. An important is the refinement in the measures we presented in this chapter to detect different roles in terrorist networks. Also we would like to continue research on the evolution of network structure. It would be interesting to compare structure of multiple terrorist networks to see how they evolve over time.

As mentioned in this chapter we have developed a large knowledge base of terrorist networks by harvesting Web. We are interested to extend this work and to construct a fuzzy knowledge base. Semantic Web languages can be considered for this purpose. The metadata used in homeland security projects are fuzzy by nature, therefore the semantic web could be appropriate for representing fuzzy data.

References

[1] Memon, N., Larsen HL., Practical approaches for analysis, visualization and destabilizing terrorist networks. In the proceedings of ARES 2006: The First International Conference on Availability, Reliability and Security, Vienna, Austria, IEEE Computer Society, pp. 906-913.

[2] Memon, N., Larsen, H. L.(2006) Practical algorithms of destabilizing terrorist networks. In the proceedings of IEEE Intelligence Security Conference, San Diego, Lecture Notes in Computer Science, *Springer-Verlag*, Vol. 3976: pp. 398-411 (2006)

[3] Memon, N., and Larsen, H. L. (2006) Detecting Terrorist Activity Patterns using Investigative Data Mining Tool. *International Journal of Knowledge and System Sciences*, Vol. **3**, No. 01, pp. 43-52.

[4] Memon N., Qureshi, A. R.. (2005). Destabilizing terrorist networks. *In WSEAS Transactions on Computers*, Issue 11, Vol. **4,** pp.1649-1656

[5] Taskar Ben, Pieter Abbeel, Ming-FaiWong, and Daphne Koller. (2003) Label and link prediction. In relational data, in IJCAI Workshop on Learning Statistical Models from Relational Data. http://kdl.cs.umass.edu/srl2003_upload/files/taskar-paper.pdf

[6] M. Barlow, J. Galloway, and H. Abbass. (2002). Mining evolution through visualization. In Proceedings of Workshop on Beyond Fitness: Visualization Evolution at the 8th International Conference on the Simulation and Synthesis of Living Systems, http://www.alife.org/alife8/workshops/15.pdf

[7] Q&A with Professor Karen Stephenson, April 18, 2006 http://www.elearningpost.com/articles/archives/qa_with_professor_karen_stephenson/

[8] DeRosa M., (2004), *Data Mining and Data Analysis for Counterterrorism*, CSIS Report.

[9] Memon, N. et al.: Understanding the structure of covert networks**.** *International Journal of Business Intelligence and Data Mining*. 2007 ; Vol. 2, No. 4, p. 401-425.

[10] Memon, Nasrullah.: Investigative data mining: Mathematical models for analyzing, visualizing and destabilizing Terrorist Networks. 2007. PhD Dissertation, Aalborg University, Denmark.

[11] Memon, N. et al: Small world terrorist networks: A preliminary Investigation**.** In: Applications and Innovations in Intelligent Systems XV, Springer Verlag. Proceedings of SGAI-2007: The Twenty Seventh SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, 2007. p. 339-344

[12] Memon, N. et al.: Topological analysis of terrorist networks**.** In: 3rd ADBIS Workshop on Data Mining and Knowledge Discovery, October 02, 2007 at Varna, Bulgaria. 2007. p. 45-57

[13] Memon, N. et al. Harvesting terrorists information from Web**.** In proc. of 11th International Conference Information Visualization, 2007. IV '07.. IEEE, 2007. s. 664-671.

[14] Krebs, V. E. (2202) Mapping network of terrorist cells. *Connections* 24(3): 43-52

[15] Allen, C. (2004). *The Dunbar Number as a Limit to Group Sizes*. Retrieved May 31, 2006, from http://www.lifewithalacrity.com/2004/03/the_dunbar_numb.html

[16] Memon, N. et al.: Detecting hidden hierarchy in terrorist networks: Some case studies**.** In proc. IEEE International Conference on Intelligence and Security Informatics (ISI 2008) Workshops Springer Lecture Notes in Computer Science (LNCS 5075). 2008. s. 477-489.

[17] Memon, N. et al.: How investigative data mining can help intelligence agencies to discover dependence of nodes in terrorist networks. In Proc. of the Third International Conference, ADMA 2007,. Springer Verlag, 2007. s. 430-441 (Lecture Notes in Computer Science; 4632).

[18] Al Qaeda Training manual www.thetulsan.com/manual.html

[19] Freeman, L. C. (1979). Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3), 15-239.

[20] Latora V., Marchiori, M.,: (2001). Efficient behavior of small world network. Phy. Rev. Let. 87(19)

[21] Sageman, M.: (2004). Understand terror networks. University of Pennsylvania Press.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-357

Geolocalisation in Cellular Telephone Networks

Bruce DENBY[a;b;1], Yacine OUSSAR[b], Iness AHRIZ[b]
Université Pierre et Marie Curie[a] et Laboratoire d'Electronique de l'ESPCI[b], France

Abstract. The paper gives an overview of GPS and radio interface based geolocalisation techniques for cellular telephone networks, including the E911 and E112 initiatives, Location Based Services, and law enforcement/security applications. An example of localisation using the Database Correlation Method is also presented.

Keywords. Cellular telephones, localisation, data mining

Introduction

While a fixed line telephone can always be associated with a street address, cellphones may be used anywhere: indoors or outdoors; in public transport; in a crowd or alone; on top of a building, or in a parking garage. There is significant interest today in technologies which allow associating a position to cellphones as well, by inserting location-specific fields into Call Detail Record, CDR, of a cellphone communications. Currently, a precision of several tens of meters is possible outdoors, and while indoor cellphone localisation is still in the research stages, accuracy of a few meters is probably a good guess for what is achievable there.

In section 1, we outline the different motivations for cellphone localisation. In section 2, after a brief review of how cellphones work, an overview of cellphone localisation techniques is given. An example of the use of the Database Correlation method using machine learning techniques is presented in section 3, and a brief conclusion in section 4.

1. Why Localise Cellphones?

1.1. Emergency Services

The initial motivation for localisation was legislation requiring cellular network operators to provide approximate positions of cellphone users in emergency situations. The E911 mandate launched in the US in 1999 requires a localisation precision of at least 50m for 67% of calls or 150m for 95% using mobile-based localisation. In 2003, the similar E112 initiative was launched in Europe. E911 and E122 require operators,

[1] Corresponding Author: E-mail: denby@ieee.org.

when requested by emergency services, to add a field to call detail records (CDR) permitting to localise the user [1].

1.2. Location Based Services

Operators soon learned that the localisation capability required for emergency services could also be exploited to provide chargeable Location Bases Services, LBS [1] to their clients. Examples are *push advertising*, in which the mobile handset receives unsolicited information on nearby commercial activities (sale, grand opening, etc.); *navigation information*, in which the user may request directions to nearby locations or relevant public transport schedules; and *social and family services* such as a map of current positions of friends or family members, and dating systems.

1.3. Health Care and Family Security

As people live longer, there is greater interest in novel health care and person monitoring systems. Tracking of persons with Alzheimer's disease is a good example. Cellphone localisation can address these problems by providing localisation information for persons of interest. Outdoors, this usually takes the form of *place monitoring*, that is, assuring that the person in question is in one of the places that he or she is known to frequent. Though still experimental, indoor monitoring should also be possible, for example to set an alarm if a person remains in the same place for an unusually long time, or exhibits other non-habitual patterns of movement.

1.4. Customer Behaviour Monitoring

Some operators provide location based billing, allowing customers to enjoy a reduced rate when telephoning from a restricted set of "local" places (home, workplace, etc.) and slightly higher rates in les frequented places, providing an overall savings. It is also possible, for clients who authorize it, for the operator to monitor customer movement and correlate this with other factors for evaluation purposes, perhaps with a view toward developing additional new services. Finally, a sudden change in a user's movement habits could be a possible signal of subscription fraud.

1.5. Law Enforcement and Forensics

Subscription fraud, or illegally sold telecommunications services, is estimated to cost the telecommunications industry $35 billion annually, and as such touches law enforcement agencies in a very real way. Localisation can also play a role in tracking persons or vehicles in cases of kidnapping or other serious crimes. In addition, the use of cellphones in certain areas, such as government installations or prisons, is restricted, and localisation can help determine if unauthorized persons are using radiocommunication devices from such area.

It has become common in recent years to use forensic analysis of cellphone call records as trial evidence in court cases. Localisation technology can contribute by establishing the whereabouts of user at a particular time via the localisation records in CDRs of calls made. It is important, in such cases, to be able to accurately specify the coverage area of a cell tower, to determine if the caller was "in range." Propagation phenomena such as resurgences can make this difficult, particularly in built-up areas.

1.6. National and International Security

Localisation of the user of a cellphone can be of interest in espionage and counterespionage. Although this may seem exotic, there was a recent example of widespread, illegal tapping of cellular telephones in Greece for apparent espionage purposes [2]. There have also been cases of "fake" cellphone towers sending repeated authentication requests to idle phones to obtain enough data to crack encryption security. Localisation is one tool which can be brought to bear in such cases. In cases of suspected terrorist activity, localisation for person tracking is also clearly of interest.

A recent issue has been the United States National Security Administration's request for huge volumes of CDRs of everyday citizens for national security. The European Union's Data Retention Directive of March 2006 is intended for similar purposes. At issue is exactly what information is contained in these files and whether it constitutes a violation of individuals' privacy.

2. Overview of Cellphone Localisation Techniques

2.1. Review of How Cellphones Work

2.1.1. GSM Architecture and Operation

In order to understand some of the methods that will be subsequently presented, it is necessary to have a basic understanding of cellular telephone networks. A functional schematic of a GSM network is given in figure 1. The Base Station Subsystem, BSS, is comprised of the cell towers, called Base Transceiver Stations or BTS, Base Station Controllers, BSC, each of which controls several BTS, and of course the user's cellphone or Mobile Station, MS. The cell in which the MS is currently registered is called the Serving Cell, surrounded by its Neighbor Cells.

Authentication and roaming are handled in the core network by the Home Location register, HLR, containing all subscriber information, and the Visitor Location Register or VLR, which keeps a record of an MS's current location. The Mobile Station Switching Centers, MSC, each handle several BSCs and serve as the link between the BSS and the PSTN, or Public Switched Telephone Network.

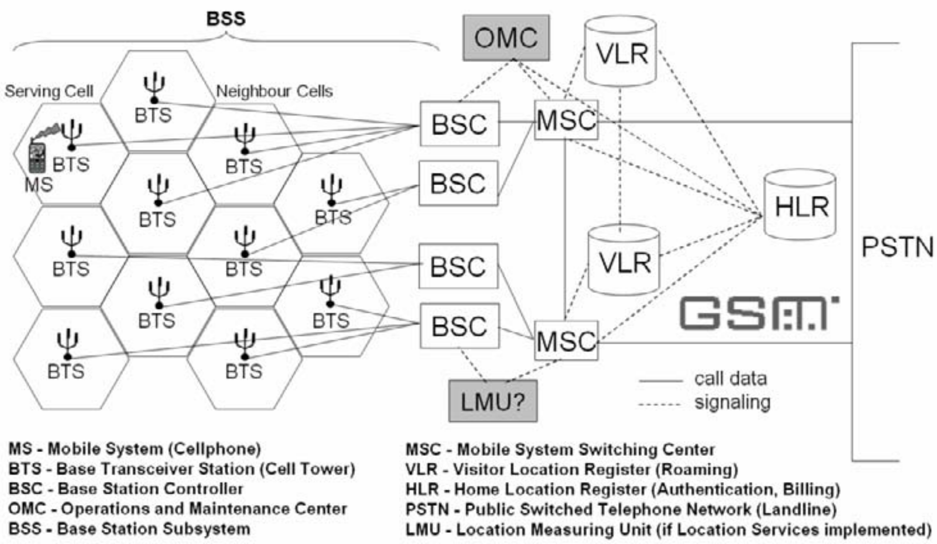

BSS
OMC
VLR
Serving Cell
Neighbour Cells
BTS
MS
BSC
MSC
HLR
VLR
PSTN
GSM
LMU?
call data
signaling
MS - Mobile System (Cellphone)
BTS - Base Transceiver Station (Cell Tower)
BSC - Base Station Controller
OMC - Operations and Maintenance Center
BSS - Base Station Subsystem
MSC - Mobile System Switching Center
VLR - Visitor Location Register (Roaming)
HLR - Home Location Register (Authentication, Billing)
PSTN - Public Switched Telephone Network (Landline)
LMU - Location Measuring Unit (if Location Services implemented)

Figure 1. Functional schematic of a GSM cellular telephone network.

Localisation can enter the picture in one of two ways. The first is via the Operations and Maintenance Center, OMC, which oversees all network operation and serves as the entry point for emergency or law enforcement services needing to localise a particular MS. The second is through Location Measuring Units, LMU, optional in GSM networks, which provide synchronisation and other information to the BSCs to facilitate some of the localisation methods which will be discussed below.

2.1.2. A Word About CDRs

Call Detail Records, CDR, are a billing mechanism first implemented in fixed line communications for keeping track of who called whom and for how long. These initially came out of the exchange hardware on a serial port for printing. Today, CDRs are stored in databases which are saved for 30-60 days depending on operator preference. A baseline CDR consists of: calling and called numbers; channel number used; date and timestamp; elapsed time; call failure class if any; and billing amount.

CDRs in cellular networks contain additional fields due to their additional complexity, including additional call setup information, cell tower linked to, etc. In addition, cellular operators have the freedom to insert new record types, some of which may be proprietary. A major telecom carrier generates hundreds of millions of CDRs per day, corresponding to terabytes of data. These new, richer CDR databases can be analysed to extract information of different types, including localisation. The CDR files must be processed in real time to provide customers immediate access to billing information, as well as for law enforcement and security requests.

2.2. Cellphone Localisation Methods

Localisation is an add-on to cellphone systems, which were initially designed only for communications purposes. As such, solutions adopted for localisation should to the greatest extent possible make use of existing infrastructure or involve only minimal upgrades. The addition of LMUs to a GSM network is one example of such an upgrade. Solutions should also when possible be reverse compatible with the millions of "legacy" phones which may not contain the latest hardware or application software versions. The targeted performance for localisation systems, as mentioned earlier, is a few meters, whether for indoor or outdoor

2.2.1. GPS

It is natural to ask whether GPS is an appropriate technology for cellphone localisation. The GPS network consists of 24 satellites intended for terrestrial navigation applications, which furnishes a position resolution of about 10m in normal usage and less than 1m in differential mode (DGPS). Furthermore, some cellphones already have built-in GPS receivers, and the industry trend is for 100% integration using a new generation of small, light, low power GPS chips.

Of course, the great majority of legacy phones do not have GPS, but what is an even more serious problem is that GPS satellite coverage is poorly adapted to localisation of cellphones carried by pedestrians. Indeed, GPS requires at least 4 locked-in satellites to function correctly, and for full precision even more may be required. "Urban canyon" environments very often do not offer enough visibility to lock in 4 satellites, especially in pedestrian scenarios, where the user is on a sidewalk close to buildings, enters and leaves indoor spaces frequently, etc. GPS of course does not work at all in indoor scenarios, and even when enough satellites are available, lock-in times are considerably longer when visibility is marginal or variable. At the same time, deep urban and indoor environments are of significant interest in emergency situations, for person tracking, law enforcement issues, etc.

2.2.2. Radio Interface Localisation Techniques

Given that GPS is not currently appropriate for a ubiquitous cellphone localisation scheme, it is interesting to examine other techniques, in particular those based on the cellular radio interface itself. These can exploit existing digital location-dependent network variables such as Cell ID (the ID of the tower serving the call), Received Signal Strength (RSS), Timing Advance (TA), etc., or, may make use of characteristics of the received electromagnetic signal itself, such as amplitude, frequency, timing, direction, etc.

2.2.2.1. Cell ID

The simplest method of localising a cellphone is to simply record the position of the serving cell tower. In dense urban environments where picocell sizes can be less than 100m, this can already provide a useful result. However, the maximum GSM cell size is 35 km, and, in general, one would like to do better.

2.2.2.2. Cell ID + TA

Timing Advance, TA, is a GSM network variable used to account for the round-trip propagation time between the MS and the BTS, which, thus, is proportional to the distance between transmitter and receiver. Unfortunately, the position resolution of TA is equal to the speed of light times the bit time, c^*T_{bit} = 500m, which is rather poor. Theoretically, one can do better in spread-spectrum 3G networks where the chip time dominates and c^*T_{chip} = 35m; however, multipath effects make it impossible to achieve this resolution in practice (see section 2.2.2.5).

2.2.2.3. Triangulation

Better resolution is possible by measuring the time delay of a cellphone signal received at 3 BTS simultaneously, for example with the Observed Time Delay of Arrival (O-TDOA) and similar methods. The MS position is taken to lie at the point of intersection of three circles with radii proportional to the measured delay times. This technique, however, requires the BTS to be synchronised, which is not the case in a standard GSM network. Synchronisation can be accomplished at the cost of installing LMUs, but the technique is unfortunately also susceptible to degradation due to multipath effects, which negates much of the advantage gained.

2.2.2.4. Angle of Arrival

In AoA, the MS is pinpointed by detecting the angle of arrival of its signal at two BTS. To do this, however, requires directional antennas at the BTS, which, again, are not part of the GSM norm and would require a costly system upgrade. This technique, as with TA and triangulation, is also compromised by multipath effects.

2.2.2.5. Multipath and Mask Effects

In urban environments the received signal is a superposition of direct, reflected, and diffracted rays, each with its own amplitude, direction, and time delay. In Non-Line of Sight (NLOS) situations, which are the norm in urban propagation channels, the direct ray is “masked” by an obstacle. Position resolution is then degraded because the remaining paths have different delays and angles. Multipath and mask effects are also important in indoor propagation channels

2.2.2.6. Database Correlation Method

Many research and commercial localisation implementations today use the Database Correlation method. The procedure works as follows. In order to always be ready for a handover, the GSM norm requires MS to send regular (~1Hz) Network Measurement Reports (NMR) to the BTS, containing the signal strengths of the serving cell and the 6 strongest neighbours. This seven element vector can be interpreted a “fingerprint” of the local radio environment. Initially, a database of fingerprints that are position-labelled using GPS or some other technique is first acquired. Then, to localise a mobile, a recent NMR is checked against this database using some machine learning technique (k-NN, SVM, etc.). The database correlation method automatically takes into account multipath and mask effects. It uses the standard GSM norm and involves no changes to handsets or infrastructure.

For good precision, however, a fine measurement grid is needed, which requires trace tools and can be expensive and time-consuming to acquire. One solution is to use RSS predictions rather than measurements; however, in addition to giving poorer accuracy, this approach requires operator specific information and special expertise and software tools. A more promising new approach is the use of semi-supervised techniques, in which the amount of labelled data is reduced [3].

There are also a number of other challenges in using the database correlation technique. RSS measurements at a given position follow a Rayleigh distribution, making them quite variable. They also vary with meteorological conditions (dry/rain) and with the season due to foliage effects. Consequently, it is rare to find exactly the same 7 cell towers in two measurements at a given position, making it necessary to use methods which allow for imperfect matches. A database for a big city will contain information on thousands of BTS, only a few of which are important at any location. Although the method could undoubtedly be improved with the knowledge of cell tower locations, the network operator may not want to divulge this information, which, in any case, is undoubtedly not constant over time as the system is maintained and upgraded.

2.2.2.7. Indoor Fingerprint Localisation

GSM has good indoor penetration to allow calls to be made anywhere inside a building. It is clear that as an MS moves about, the absorption of the radio signals from the surrounding BTS by surrounding structural elements will depend heavily upon position. This, however, is precisely what one wants for an RSS-based localisation system. Preliminary studies at our lab (ESPCI) have shown that this phenomenon can be exploited to obtain room classification efficiency in an apartment setting approaching 100% [4].

There is currently substantial interest in indoor localisation using fingerprinting in WiFi networks as well. As some cellphones can connect to WiFi, universal localisation may one day be possible using a single portable device. Home environments will normally not contain large numbers of access points; WiFi localisation may thus be more of interest in the workplace, in airports, etc. The 2007 International Conference on Data Mining sponsored a Data Mining Contest on indoor WiFi localisation using fingerprinting [5], in which only 10% of the data was labelled, which encourages the use of semi-supervised learning methods.

2.2.2.8. Hybrid Method: Assisted GPS

In A-GPS, a GPS-equipped cellphone reporting fewer than 4 visible satellites may relay this partial information to a nearby BTS along with an NMR fingerprint. The network is able to use additional information to resolve the ambiguity in many cases, and subsequently return the correct position information to the mobile in a downlink message. Assisted GPS is already under test by some operators. It may furthermore be of interested for producing labelled fingerprints for use by other localisation systems.

3. Example of Localisation Using the Database Correlation Method

In this section we present an example of the use of the Database Correlation method. It should not be interpreted a research result, rather a pedagogic example on a real dataset which illustrates the various aspects of the technique. Some 100 km of GSM in-car traces were recorded in Paris, France, using the TEMS trace mobile system [6]. This represents a total of about 3 hours of talk time over a 5 day period. Data were labelled with ground coordinates obtained with a GPS device recorded at the same time.

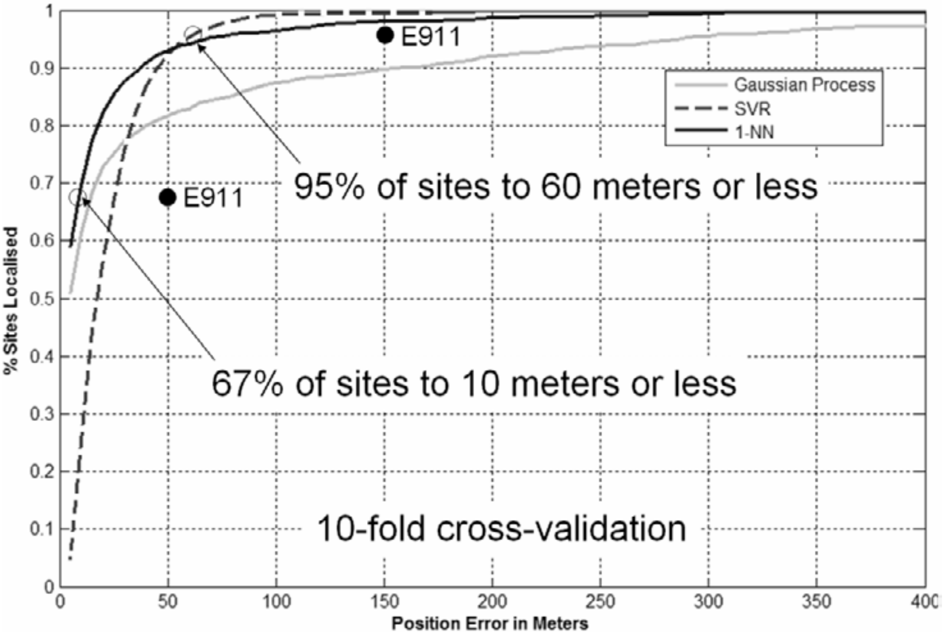


Figure 2. Percentage of sites localised versus position error in meters.

The NMR data were divided into a training set of 4700 elements and 2000 more for testing. The objective is to use the knowledge in the training set to predict the GPS coordinate of an element in the test set, based upon 7 element RSS vectors of the serving cell and 6 strongest neighbouring BTS. The machine learning methods explored were nearest neighbour (k-NN), support vector regression (SVR) [7], and Gaussian Process. This latter method, described, for example, in [8], is a likelihood-based approach which assumes the RSS distributions are Gaussian and allows imperfect matches to be naturally taken into account via a penalty term. For NN and SVR, the problem of imperfect matches was addressed by creating a fixed length input vector containing all BTS ID's encountered in the test set; subsequently, any BTS not present in a particular test set element were set to zero. A ten-fold cross validation was used for all methods.

Results are given in figure 2, which shows percentage of sites versus position accuracy in meters. The figure shows that even with our rather sparse, *ad hoc* data set, it is not difficult to satisfy the E911 requirements, at least when NN or SVR techniques are used. Somewhat surprisingly, Gaussian Process gave much poorer results. Also surprising is that the optimum k value for k-NN was found to be k=1, and that SVR performs worse than 1-NN at small distances. Clearly it would be desirable to combine the classification techniques into a single, optimal classifier.

4. Conclusion

Emergency services legislation has obliged cellular network operators to provide localisation information for its subscribers. This served as an impetus for most to also offer Location Based Services on a paying basis. GPS is only viable for cellular localisation in a restricted set of circumstances; at the same time, many of the radio interface based techniques are expensive to implement and are compromised by multipath effects. The database correlation method has recently emerged as a high performance, easy to implement technique allowing to bypass most of these problems.

Acknowledgements

The author wishes to thank G. Dreyfus for assistance with the analysis of the database correlation example of section 3 and for helpful suggestions.

References

[1] A. Küpper, *Location-Based Services: Fundamentals and Operation*, John Wiley & Sons, 2005.

[2] V. Prevelakis, D. Spinellis, The Greek Cellphone Caper, *IEEE Spectrum,* **44** (July 2007)*,* 26-33.

[3] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, 2006.

[4] B. Denby, Y. Oussar, I. Ahriz, G. Dreyfus, article in preparation.

[5] http://www.cse.ust.hk/~qyang/ICDMDMC07/

[6] Test Mobile System, available at: www.ericsson.com/solutions/tems/.

[7] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, **2** (1998), 121-167.

[8] D. Zimmerman, J. Baumann, M. Layh, F. Landstorfer, R. Hoppe, G. Wölfle, Database Correlation for Positioning of Mobile Terminals in Cellular Networks using Wave Propagation Models, *Proc. IEEE 60th Vehicular Technology Conference*, **7**, (26-29 Sept. 2004), 4682-4686.

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008
© 2008 IOS Press. All rights reserved.
doi:10.3233/978-1-58603-898-4-366

Machine Learning for Intrusion Detection

Pavel LASKOV [a,b,1], Konrad RIECK [a] and Klaus-Robert MÜLLER [a,c]
[a] *Fraunhofer Institute FIRST.IDA*
[b] *University of Tübingen, Wilhelm-Schickard-Institute for Computer Science*
[c] *Technical University of Berlin, Faculty IV*

Abstract

Detection of unknown attacks in network traffic is gaining increasing importance as modern attacks are characterized by high variabilities and mutation rates. Traditional signature-based intrusion detection systems (IDS) are not able to detect unknown attacks due to failing availability of appropriate signatures. We present an alternative approach based on machine learning techniques which enable automatic construction of profiles for normal packet payloads and detection of deviations thereof. Experimental evaluation of our approach showed a remarkable detection accuracy at low false positive rates and a major improvement in comparison to the widely used open-source IDS Snort.

Keywords. Machine learning, Intrusion detection, Anomaly detection

Introduction

Intrusion detection is one of the core computer security technologies. The goal of intrusion detection is to identify malicious activity in a stream of monitored data; the latter can be network traffic, operating system events, log entries, etc. The importance of intrusion detection stems from its ability to provide input to *reactive* security mechanisms which can provider a better tradeoff between restriction and functionality than currently prevalent proactive mechanisms such as encryption, authentication, access control, etc.

The majority of current intrusion detection systems (IDS) is based on signatures, specific pre-defined patterns matching known functionality of attacks. The main limitation of the signature-based approach is its inability to identify novel attacks. Even minor variations of known exploits using polymorphic obfuscation techniques can evade signature-based IDS. Besides, a significant administrative overhead is incurred by the need to maintain extensive signature databases up-to-date.

Machine learning offers a major opportunity to improve quality and to facilitate administration of IDS. A significant body of machine learning research has been devoted to unsupervised learning methods capable of discovering relevant structure in data without human intervention [e.g. 1, 3, 4, 7, 8]. The main goal of this contribution is to develop specialized feature extraction and anomaly detection methods suitable for highly accurate detection of *novel* and *unknown* attacks. Our main assumption – largely supported by the presented experiments – is that attacks exhibit anomalous behavior in some fea-

[1] Corresponding Author: Fraunhofer Institute FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany; E-mail: pavel.laskov@first.fraunhofer.de.

ture space. Hence they can be recognized by anomaly detection methods provided an appropriate features space and the notions of anomality are constructed.

We now focus on the two essential components of our approach: feature extraction and anomaly detection methods. A large body of previous research has concentrated on features extracted from packet headers, see e.g. [5, 6] and references therein. On the other hand, most of exploits use legitimate network-level context in order to reach their victims. The crucial difference lies, however, in the content they are trying to deliver. Due to this inherent limitation of header-based features, which reflect at most the symptomes of an attack, we focus on payload-based features initially explored in [14].

The payload of a packet or connection can be viewed as a stream of bytes following some syntax and bearing some semantics, both defined by an appropriate application-level protocol. Full recovery of syntax and semantics of a packet payload is, in general, only possible by a target application. Even partial protocol analysis is usually considered too slow for high traffic volumes to be processed by a network IDS. Therefore we limit our analysis to simple language models, such as n-grams and words, for which efficient analysis algorithms can be developed. A fundamental operation required by most anomaly detection algorithms is pairwise comparison of events. While this can be easily done for vectorial data, the definition and the efficient computation of similarity measures for sequences is far from being trivial. Unlike the previous ad-hoc methods for comparing packet payloads, e.g. the Mahalanobis distance of byte histograms [14] or the Bloom filter [13], we develop a systematic approach to payload-based feature extraction using the embedding of sequences in a vector space, as proposed in [9]. The particular algorithms are presented in Section 1.

The generality of our approach allows us to apply a wide range of anomaly detection methods using similarity measures between objects. The key feature of our algorithms is their unsupervised character, i.e. they learn to discriminate between normal data and attacks *without training on clean data*. This feature significantly facilitates deployment of IDS in practice since no special care must be taken to sanitize the training data. Details of our algorithms are presented in Section 2.

The effectiveness of the proposed approach combining unsupervised anomaly detection methods and similarity measures over byte sequences is demonstrated on a real dataset created at our institute by a penetration testing expert. As a sample of our results, presented in Section 3, we note that quite remarkable detection accuracy of 77–100% *at zero false alarm rate* has been observed in our experiments.

1. Feature Extraction

Network traffic, as seen by a network IDS, contains heterogenous information the semantics of which is defined by the appropriate network protocols. The network protocol stack defined by OSI contains 7 layers, so that upper layer information is encapsulated within a data frame of a lower layer protocol. The protocol-related control information is contained either in a fixed-format header (typical for lower layer protocols) or in a variable-format body (typical for upper layer protocols). This general property of the network packet structure strongly affects the techniques that are applicable for feature extraction, as shown in Figure 1.

So called “flat” features, containing a vector of parameters and typical for other machine learning applications, can be easily computed for packet headers [see 5]. It is,

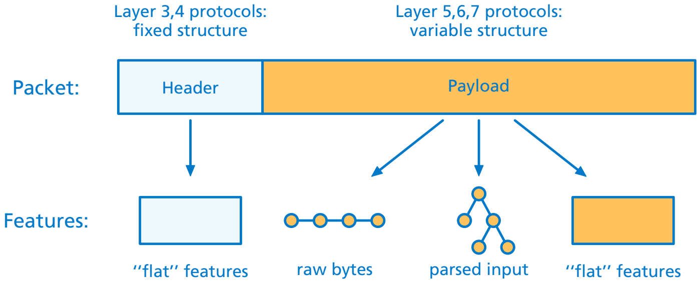


Figure 1. Overview of feature extraction in network data.

however, quite difficult to compute flat features for packet payloads, as this would require extensive expert knowledge of appropriate application layer protocols. On the other hand, as it was mentioned in the introduction, packet header information is insufficient for detection of application-specific exploits. Therefore, we focus on the easiest of the payload-based feature representation in Figure 1, *byte sequences*.

Given a byte sequence $\mathbf{x}$ observed as a payload of a packet[2], we consider it as a sentence generated from some language L over the alphabet $\mathcal{A}$, a set of 256 possible byte values. The language L is a subset of all possible sequences $\mathcal{A}^*$ generated from $\mathcal{A}$. The language encapsulates certain prior knowledge about the nature of information transmitted in a given application layer protocol, e.g. a set of accepted keywords. In the absence of specific protocol knowledge, one can construct generic languages, e.g. a set of all sequences of length n (n-grams), or a set of all consecutive symbols between certain delimiters (a "bag-of-words").

Given a language L, an equivalent representation of a sequence $\mathbf{x}$ in a vector space can be constructed using a language-specific *embedding function* $\phi_w(\mathbf{x})$. This function assigns, for every word $w \in L$ found in $\mathbf{x}$, a certain value based on appearance of w in $\mathbf{x}$. For example, $\phi_w(\mathbf{x})$ can measure the frequency of every word w in $\mathbf{x}$, or simply indicate whether or not w is present in $\mathbf{x}$. The embedding function maps sequences into an equivalent vector space in which similarity between sequences can be efficiently computed.

As an example for the computation of a similarity measure, consider the Manhattan distance defined as:

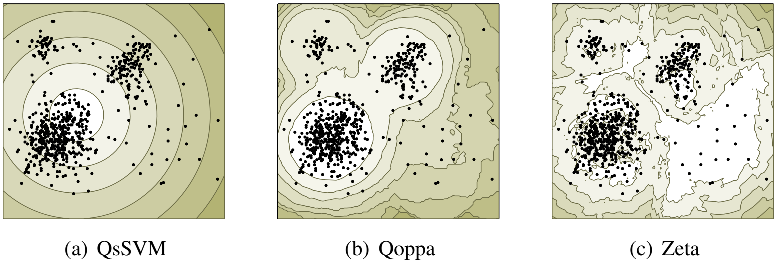
$$d(\mathbf{x}, \mathbf{y}) = \sum_{w \in L} |\phi_w(\mathbf{x}) - \phi_w(\mathbf{y})|. \tag{1}$$

This computation involves an outer loop running over all possible sequences in L and computing an expression involving the "coordinates" $\phi_w(\mathbf{x})$ and $\phi_w(\mathbf{y})$ of the sequences $\mathbf{x}$ and $\mathbf{y}$. In general, the number of words in L can be exponential or even infinite in the length of $\mathcal{A}$, however, only a linear number of features is nonzero in any pair of sequences. Hence computation can be efficiently carried out provided only nonzero entries are accessed. This can be done by using specialized data structures, e.g. lexicographically sorted arrays or suffix trees. The reader is referred to [9] for a detailed description and evaluation of several suitable data structures and algorithms.

[2] Alternatively, byte streams of all packets in a connection can be merged into a single byte stream.

2. Anomaly Detection Methods

The main goal of anomaly detection methods is to detect anomalous events, which can be instances of previously unknown attacks. Anomaly detection can supplement signature-based analysis by constructing a model of normal activity and measuring deviations of incoming events from this model. The goal of machine learning algorithms to be presented in this section is to efficiently construct and evaluate such models in the presence of “malicious noise”. Both of these operations can be expressed geometrically by using pairwise similarities between objects. As an example, Figure 2 depicts three different geometric models of normality which are covered in detail in the following sections.



(a) QsSVM (b) Qoppa (c) Zeta

Figure 2. Anomaly detection methods using a global hypersphere (QsSVM), local hyperspheres (Qoppa) and the local neihgborhood (Zeta) of points. Light shading indicates normal and dark shading anomalous regions.

2.1. Quarter-sphere Support Vector Machine

A simple and intuitive geometric model of normality is a *global hypersphere* centered at the mass of normal data. Deviation of points from this model is measured by the Euclidean distance from the center of the hypersphere [11]. Figure 2(a) shows the Euclidean distances as contour lines from the center of mass of a two-dimensional toy data set.

Mathematically this model can be derived from one-class learning methods [3, 12]. For a novel point $\mathbf{x}$ and a set of previous points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ the deviation is computed by the *Quarter-sphere Support Vector Machine* (QsSVM) defined as

$$s(\mathbf{x}) = \sqrt{k(\mathbf{x}, \mathbf{x}) - \frac{2}{n} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i) + \frac{1}{n^2} \sum_{i,j=1}^{n} k(\mathbf{x}_i, \mathbf{x}_j)}. \tag{2}$$

Note that (2) is solely formulated in terms of pairwise kernels $k(\cdot, \cdot)$ between data points, so that any kernel function over embedded features may be applied for computing the hypersphere and the distance from its center.

2.2. Qoppa Anomaly Score

Measuring deviation as a distance from a global center is difficult if anomalies are locally distributed, i.e. outliers are concentrated in small clusters. This problems can be alleviated by considering anomaly detection methods defining a local model of normality.

A simple model of local normality can be derived from the concept of single linkage clustering [2] and is obtained by putting *local hyperspheres* with fixed radius r on all data points. The deviation of a novel point $\mathbf{x}$ is inversely proportional to the number of previous points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ falling within the radius r from $\mathbf{x}$. It is given by the so called *Qoppa anomaly score* $ϙ$

$$ϙ_r(\mathbf{x}) = -\log \sum_{i=1}^{n} \llbracket r - d(\mathbf{x}, \mathbf{x}_i) > 0 \rrbracket. \tag{3}$$

Note that (3) is expressed solely in terms of a distance function $d(\cdot, \cdot)$, so that any distance between extracted features may be applied for its computation. The intuition underlying this geometric model is that benign points often lie in dense areas, so that several neighboring points fall within the radius r, where for anomalous points only few neighbors lie within the range of r. Figure 2(b) shows the Qoppa anomaly score for a two-dimensional toy data set. In comparison to the global hypersphere, the model reflects distinct groups of benign objects in the data.

2.3. Zeta Anomaly Score

If the normal data is multi-modal, i.e. comprising regions of different density, choosing a fixed radius a priori may not work well. This problem can be addressed by considering the *local neighborhood* of each point $\mathbf{x}$ determined by the set of k nearest neighbor points $\{\mathrm{nn}_1(\mathbf{x}), \ldots, \mathrm{nn}_k(\mathbf{x})\}$. Density-independent deviation of a point $\mathbf{x}$ from this model can be calculated by considering the mean distance of $\mathbf{x}$ to its neighbors and the mean distance between the neighbors [8]. It is given by the *Zeta anomaly score*

$$\zeta_k(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{k} d(\mathbf{x}, \mathrm{nn}_i(\mathbf{x})) - \frac{1}{k^2} \sum_{i,j=1}^{k} d(\mathrm{nn}_i(\mathbf{x}), \mathrm{nn}_j(\mathbf{x})). \tag{4}$$

The first term emphasizes points that lie far away from its neighbors, whereas the second term discounts abnormality of points in wide neighborhood regions. Figure 2(c) depicts Zeta anomaly scores for a two-dimensional toy data set using contour lines. Note that the sparse region in the lower right of Figure 2(c) corresponds to benign objects in comparison to the other models of normality given in Figures 2(a) and (b).

3. Experiments

3.1. Experimental Setup

Experiments were conducted on the *PESIM 2005* dataset, which has been used in previous research [see 8, 9]. The dataset was recorded at our laboratory and comprises normal network traffic of HTTP, FTP and SMTP protocols. Attacks were injected by a security expert using penetration testing tools such as the Metasploit framework.

The feature extraction and anomaly detection methods are evaluated on randomly sampled mixtures of *unseen normal and attack data* containing 2% to 14% malicious

Table 1. Best configuration of similarity measure and anomaly detection method for *n-gram* features.

Protocol	Similarity Measure	Anomaly Detection	$\text{AUC}_{0.01}$
HTTP	Geodesic distance (freq.)	Zeta	0.8520
FTP	Linear kernel (freq.)	QsSVM	0.8653
SMTP	Linear kernel (freq.)	QsSVM	1.0000

Table 2. Best configuration of similarity measure and anomaly detection method for *word* features.

Protocol	Similarity Measure	Anomaly Detection	$\text{AUC}_{0.01}$
HTTP	Kulczynski coefficient (bin.)	Qoppa	0.8406
FTP	Kulczynski coefficient (bin.)	Qoppa	0.8239
SMTP	Linear kernel (freq.)	QsSVM	1.0000

connections. No explicit learning involving labeled attacks is performed. As some algorithms require certain parameters to be set, we precede the evaluation with a validation stage, at which the best parameters are automatically selected based on independent samples of our datasets.

As evaluation criterion for the experiments we choose the *area under the ROC curve* denoted as $\text{AUC}_{0.01}$, which integrates true-positive rates over the fixed interval [0,0.01] of false-positive rates. For statistical significance the results for all experiments are averaged over 30 validation/evaluation runs on randomly drawn samples each comprising 1,000 connections.

3.2. Experiment 1: Best Configuration

The feature extraction techniques and similarity measures introduced in Section 1 induce different geometric properties of embedded objects which are explored in different ways by unsupervised anomaly detection methods. Hence, as a first step, we need to roughly establish what configuration of similarity measures and anomaly detection methods perform best on features extracted from network payloads.

We restrict this evaluation to the language models of *n-grams* and *words*. To avoid fixing an n-gram length a priori, we construct a combined anomaly detector computing the highest anomaly value for multiple n-gram models with n ranging from 1 to 7. As the considered network protocols are all text-based we define the following set of delimiter symbols for the word features:

```
CR LF TAB SPC , . : / & ? = ( ) [ ]
```

Table 1 lists the best configuration for n-gram features on network payloads and Table 2 the best configurations for word features. For all network protocol an $\text{AUC}_{0.01}$ value over 0.82 is obtained, which indicates the high precision achieved by the anomaly detection methods at low false-positive rates – even though unseen attacks were present in the learning data. Different configurations of similarity measures and anomaly detection methods perform best on each network protocol, e.g. for SMTP the QsSVM yields perfect detection performance on n-gram and word features, while for HTTP Zeta (n-gram features) and Qoppa (word features) provide the highest detection accuracy.

3.3. Experiment 2: Detection Performance

We now compare the detection performance achieved by machine learning techniques from the previous experiment against the open-source IDS *Snort* [10] (Snort version 2.4.2, released on 09/2005 and configured with the default rules). Figure 3 depicts ROC curves for the best n-gram and word feature configuration as well as Snort on HTTP, FTP and SMTP traffic of the PESIM 2005 dataset.

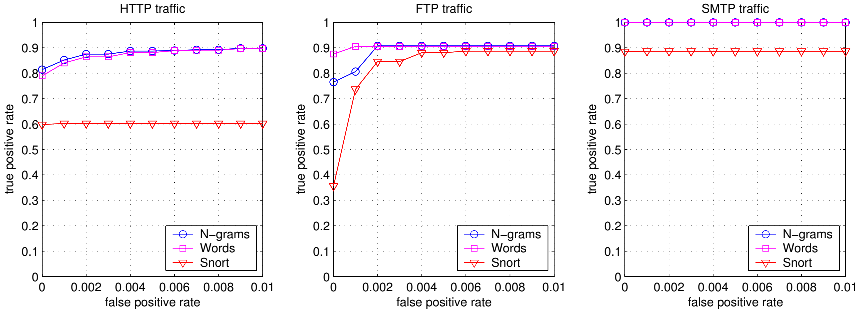


Figure 3. ROC curves for best configuration of n-gram and word features vs. Snort IDS

Both anomaly detection methods significantly outperform Snort reaching an accuracy between 77%-100% with no false-positives. The word-based detector is slightly less accurate on the FTP protocol than the detector combining multiple n-gram lengths, however, the marginal decrease in accuracy can be considered acceptable in comparison to the n times smaller computational load.

The inferior accuracy of Snort – especially for the HTTP and SMTP protocol – is surprising provided that most attacks of the PESIM 2005 data set were known months before the release date of the Snort distribution. This result confirms a misgiving that signature-based IDS may fail to discover "fresh" attacks despite a major effort in the security community to maintain up-to-date signature repositories.

4. Conclusions

The main contribution of the paper is the framework for detection of unknown attacks in network traffic using unsupervised machine learning techniques. Our approach is based on (a) embedding of byte streams from network packets in a high-dimensional vector space induced by some pre-defined language and (b) constructing models of normal activity using similarity measures between byte sequences. The main advantage of this approach over the traditional signature-based network IDS is its ability to reliably detect previously unseen exploits without training on clean network data – a feature of major importance due to increasing variability and mutability of modern exploits.

The experiments carried out on a real dataset created by a penetration testing expert showed a remarkable accuracy of 77–100% at zero false-positive rate. This constitutes not only a major improvement over the de-facto standart Snort IDS, but provides a proof of concept for a wider application of anomaly-based intrusion detection systems. In particular, we can envision that many security-related applications, e.g. malware analysis, compromise detection and Internet early warning, can significantly benefit from the deployment of machine learning techniques.

Acknowledgements

The authors gratefully acknowledge the funding from *Bundesministerium für Bildung und Forschung* under the project MIND (FKZ 01-SC40A) and REMIND (FKZ 01-IS07007A). The authors would like to thank Sören Sonnenburg and Mikio Braun for fruitful discussions and support.

References

[1] D. Barbará and S. Jajodia, editors. *Applications of Data Mining in Computer Security*, chapter A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. Kluwer, 2002.

[2] R. Duda, P.E.Hart, and D.G.Stork. *Pattern classification*. John Wiley & Sons, second edition, 2001.

[3] P. Laskov, C. Schäfer, I. Kotenko, and K.-R. Müller. Intrusion detection in unlabeled data with quarter-sphere support vector machines (extended version). *Praxis der Informationsverarbeitung und Kommunikation*, 27:228–236, 2004.

[4] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proc. of SIAM International Conference on Data Mining (SDM)*, 2003.

[5] W. Lee and S. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information Systems Security*, 3:227–261, 2000.

[6] M. Mahoney and P. Chan. PHAD: Packet header anomaly detection for identifying hostile network traffic. Technical Report CS-2001-2, Florida Institute of Technology, 2001.

[7] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *Proc. of ACM CSS Workshop on Data Mining Applied to Security*, 2001.

[8] K. Rieck and P. Laskov. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology*, 2(4):243–256, 2007.

[9] K. Rieck and P. Laskov. Linear-time computation of similarity measures for sequential data. *Journal of Machine Learning Research*, 9(Jan):23–48, 2008.

[10] M. Roesch. Snort: Lightweight intrusion detection for networks. In *Proc. of USENIX Large Installation System Administration Conference LISA*, pages 229–238, 1999.

[11] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[12] D. Tax and R. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11–13):1191–1199, 1999.

[13] K. Wang, J. Parekh, and S. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Adances in Intrusion Detection (RAID)*, pages 226–248, 2006.

[14] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Adances in Intrusion Detection (RAID)*, pages 203–222, 2004.

This page intentionally left blank

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

Subject Index

Mining Massive Data Sets for Security
F. Fogelman-Soulié et al. (Eds.)
IOS Press, 2008

Author Index

This page intentionally left blank