Petra Perner (Ed.)

LNAI 7988

Machine Learning and Data Mining in Pattern Recognition

9th International Conference, MLDM 2013
New York, NY, USA, July 2013
Proceedings

Springer

Lecture Notes in Artificial Intelligence 7988

Subseries of Lecture Notes in Computer Science

Petra Perner (Ed.)

Machine Learning and Data Mining in Pattern Recognition

9th International Conference, MLDM 2013
New York, NY, USA, July 19-25, 2013
Proceedings

Springer

Volume Editor

Petra Perner
Institute of Computer Vision and Applied Computer Sciences, IBaI
Kohlenstr. 2, 04107 Leipzig, Germany
E-mail: pperner@ibai-institut.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-39711-0 e-ISBN 978-3-642-39712-7
DOI 10.1007/978-3-642-39712-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013943122

CR Subject Classification (1998): I.2.6, H.2.8, I.5, F.2, I.4, F.4, H.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The nigth event of the International Conference on Machine Learning and Data Mining MLDM was held in New York (`www.mldm.de`) running under the umbrella of the World Congress "The Frontiers in Intelligent Data and Signal Analysis, DSA2013" (`www.worldcongressdsa.com`).

For this edition, the Program Committee received 212 submissions. After the peer-review process, we accepted 60 high-quality papers for oral presentation; from these 51 are included in this proceedings book. The subjects range from theoretical topics on classification, clustering, association rule and pattern mining to specific data mining methods for the different multimedia data types such as image mining, text mining, video mining and Web mining. Extended versions of selected papers will appear in the *International Journal of Transactions on Machine Learning and Data Mining* (`www.ibai-publishing.org/journal/mldm`).

Twenty papers were selected for poster presentations and are published in the *MLDM Poster Proceedings* by *ibai-publishing* (`www.ibai-publishing.org`).

A tutorial on Data Mining, a tutorial on Case-Based Reasoning , a tutorial on Intelligent Image Interpreation and Computer Vision in Medicine, Biotechnology, Chemistry and Food Industry, a tutorial on Standardization in Immunofluorescence, and a tutorial on Big Data and Text Analysis were held before the conference.

We were pleased to give out the best paper award for MLDM for the fourth time this year (`www.mldm.de`). The final decision was made by the Best Paper Award Committee based on the presentation by the authors and the discussion with the auditorium. The ceremony took place at the end of the conference. This prize is sponsored by ibai solutions (`www.ibai-solutions.de`), one of the leading companies in data mining for marketing, Web mining, and e-commerce.

The conference was completed by an outlook on new challenging topics in machine learning and data mining before the Best Paper Award Ceremony.

We would like to thank all reviewers for their highly professional work and their effort in reviewing the papers. We also thank the members of the Institute of Applied Computer Sciences, Leipzig, Germany (`www.ibai-institut.de`), who handed the conference as secretariat. We appreciate the help and understanding of the editorial staff at Springer Verlag, and in particular Alfred Hofmann, who supported the publication of these proceedings in the LNAI series.

Last, but not least, we wish to thank all the speakers and participants who contributed to the success of the conference. The next World Congress (www.worldcongressdsa.com), “The Frontiers in Intelligent Data and Signal Analysis, DSA2014,” will be held in St. Petersburg combining the following three events: International Conferences Machine Learning and Data Mining MLDM, the Industrial Conference on Data Mining ICDM, and the International Conference on Mass Data Analysis of Signals and Images in Medicine, Biotechnology, Chemistry and Food Industry MDA.

July 2012 Petra Perner

International Conference on Machine Learning and Data Mining, MLDM 2012

Chair

Petra Perner	IBaI Leipzig, Germany

Committee

Agnar Aamodt	NTNU, Norway
Jacky Baltes	University of Manitoba, Canada
Christoph F. Eick	University of Houston, USA
Ana Fred	Technical University of Lisbon, Portugal
Giorgio Giacinto	University of Cagliari, Italy
Makato Haraguchi	Hokkaido University Sapporo, Japan
Robert J. Hilderman	University of Regina, Canada
Eyke Hüllermeier	University of Marburg, Germany
Atsushi Imiya	Chiba University, Japan
Abraham Kandel	University of South Florida, USA
Dimitrios A. Karras	Chalkis Institute of Technology, Greece
Adam Krzyzak	Concordia University, Montreal, Canada
Brian Lovell	University of Queensland, Australia
Mariofanna Milanova	University of Arkansas at Little Rock,USA
Thang V. Pham	University of Amsterdam, The Netherlands
Maria da Graca Pimentel	Universidade de Sao Paulo, Brazil
Petia Radeva	Universitat Autonoma de Barcelona, Spain
Michael Richter	University of Calgary, Canada
Fabio Roli	University of Cagliari, Italy
Linda Shapiro	University of Washington, USA
Sameer Singh	Loughborough University, UK
Harald Steck	Bell Laboratoris, USA
Francesco Tortorella	Università degli Studi di Cassino, Italy
Patrick Wang	Northeastern University, USA

Additional Reviewers

Pål Sætrom (Paal Saetrom)	NTNU, Norway
Gleb Sizov	NTNU, Norway
Theoharis Theoharis	NTNU, Norway
Luigi Atzori	University of Cagliari, Italy
Davide Ariu	University of Cagliari, Italy
Giuliano Armano	University of Cagliari, Italy

Battista Biggio	University of Cagliari, Italy
Igino Corona	University of Cagliari, Italy
Luca Didaci	University of Cagliari, Italy
Giorgio Fumera	University of Cagliari, Italy
Danilo Pani	University of Cagliari, Italy
Ignazio Pillai	University of Cagliari, Italy
Luca Piras	University of Cagliari, Italy
Riccardo Satta	University of Cagliari, Italy
Roberto Tronci	University of Cagliari, Italy
Eloisa Vargiu	Barcelona Digital Centre of Technolgy, Spain

Table of Contents

The Gapped Spectrum Kernel for Support Vector Machines

Taku Onodera and Tetsuo Shibuya

Human Genome Center, Institute of Medical Science, University of Tokyo
4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
{tk-ono,tshibuya}@hgc.jp

Abstract. We consider the problem of classifying string data faster and more accurately. This problem naturally arises in various fields that involve the analysis of huge amount of strings such as computational biology. Our solution, a new string kernel we call *gapped spectrum kernel*, yields a kind of sequence of kernels that interpolates faster and less accurate string kernels such as the spectrum kernel and slower and more accurate ones such as the wildcard kernel. As a result, we obtain an algorithm to compute the wildcard kernel that is provably faster than the state-of-the-art method. The recently introduced b-suffix array data structure plays an important role here. Another result is a better trade-off between the speed and accuracy of classification, which we demonstrate by protein classification experiment.

Keywords: SVM, string kernels, the b-suffix array, protein classification.

1 Introduction

1.1 Background

One trend that is almost universal in diverse kinds of human activities today is the rapid growth of the amount of data involved. The examples are everywhere from space exploration to online commerce. Originated in the context of artificial intelligence research, statistical machine learning methods are now getting more and more popular as practical tools for the analysis of huge data. In particular, support vector machines (SVMs) [22] and the combination of the kernel trick with them have been studied extensively in the last decade. This research field is a place where continuous optimization algorithms for general data meet discrete algorithms for structural data such as strings, trees, graphs etc. through the link provided by kernels. The key is designing kernels that make it possible to achieve faster computation and accurate classification at the same time. For readers who are not familiar with this topic, Subsection 2.1 provides all the information necessary to read this paper.

Among many kernels for many types of data, string kernels, kernels for strings, have especially wide range of applications in computational biology because some of the fundamental objects in this field such as DNA or protein are naturally

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 1–15, 2013.

Table 1. Summary of the time needed to compute various string kernels

x, y: size of the input strings; k, g, m, h: parameters of kernels (see Section 5)

k-spectrum [11]	$\Theta(x+y)$
(k,m)-mismatch [10]	$O(\alpha_{k,m}(x+y))$ where $\alpha_{k,m} := \sum_{l=0}^{\min(2m,k)} \binom{k}{l}(k-l)$ [9]
k-subsequence [15]	$O(kxy)$
(g,k)-restricted gappy [13]	$O(\beta_{g,k}(x+y))$ where $\beta_{g,k} := (g-k)g^{g-k+1}$
(k,m)-wildcard [13]	$O(k^{m+1}(x+y))$
gapped spectrum [this paper]	$O(h(x+y))$ where $h < k$

represented as strings. For example, string kernels are applied to protein remote homology prediction [11,14,10,8], prediction of protein-protein interactions [2], classification of small molecules [21] and prediction of transcription regulation sites on genome [19] to name a few. Further investigation for faster and more accurate classification methods is still important because the throughput of genome sequencers is rapidly increasing due to technological advances.

1.2 Problem

In the development of various string kernels, the spectrum kernel [11] played a foundational role and one can see most other kernels as elaborations of this kernel. We give a brief survey of existing string kernels at Section 5 and formal descriptions of those kernels that are relevant to this paper at Subsection 2.2 and 2.3. But in short, elaborate kernels are better in terms of classification accuracy yet are more time consuming than the spectrum kernel. We fill this gap.

1.3 Our Solution

We introduce a new string kernel, which we call gapped spectrum kernel (Subsection 3.1). The feature corresponding to this kernel is, like that of the other string kernels, based on the statistics of short strings in the input string. But those short strings are gapped strings instead of contiguous substrings, i.e. they may contain mismatches at predetermined positions.

For example, the 2-specturm kernel characterizes the string "`ababa`" by its substrings of length 2, namely 2 "`ab`"s (at position 1 and 3) and 2 "`ba`"s (at position 2 and 4). On the other hand, the "`101`"-gapped spectrum kernel characterizes the same string by its substrings of length 3 but without caring the second characters, thus it captures 2 "`a*a`"s (at position 1 and 3) and 1 "`b*b`" (at position 2) where `*` represents an arbitrary character.

Although the efficient computation of the corresponding kernel function is not trivial, the b-suffix array [20], a recently introduced variant of the suffix array, provides a sufficiently fast algorithm. In particular, the time complexity is exponential terms-free unlike the cases for other more complex kernel functions like that of the mismatch kernel [10] or the wildcard kernel [13]. In that sense, the gapped spectrum kernel is more similar to the spectrum kernel rather than to the advanced kernels like the wildcard kernel.

We then propose an way to bridge faster but less accurate kernels and more accurate but slower kernels (Subsection 3.2). As the above example indicates, the gapped spectrum kernel has the set of predetermined positions of the gaps as its parameter and different gap positions yield different instances of the kernel. We propose picking instances one by one from a certain set randomly and merging them. What we get is the sequence of kernels converging to the wildcard kernel.

1.4 Our Contributions

This work's contributions are the followings:

1. an $O(k^m m(|x|+|y|))$-time algorithm to compute the (k, m)-wildcard kernel;
2. a better trade-off between the time and accuracy of string classification by string kernels.

As for the algorithm, it computes the wildcard kernel simply by completing the merge process mentioned in the previous subsection.[1] The bound above is m/k (≤ 1) times smaller than the state-of-the-art method. This efficiency originates from the gapped spectrum kernel computation using the b-suffix array.

In terms of the second contribution, we demonstrate the trade-off between time and accuracy through experiments on protein classification (Section 4). The key observation is that the convergence of the sequence of the kernels toward the wildcard kernel mentioned in the previous subsection is much faster at the early part than it is at the later part (a glance at Figure (a) in Subsection 4.3 might help), which means that the composite of the first few kernels is likely to be good enough. Furthermore, the random order of merger gives as accurate results as a seemingly more powerful order, which incorporates the knowledge gained by an extra work in the training phase, does. This result justifies the introduction of randomness in the merge of kernels.

2 Preliminaries

2.1 Support Vector Machines and the Kernel Method

SVMs [22] are widely used supervised learning methods to classify data into two groups. Because there are many good surveys about SVMs, e.g., [1], we just briefly summarize very fundamental notions, which suffices for our purpose. In the training phase, a SVM takes a set of points in Euclidean space, each of which is labeled either positive or negative. Then, it outputs the hyperplane dividing the positive labeled points and the negative labeled ones with the maximum margin if any. If the two groups are not linearly separable, you can allow errors with penalty to guarantee the existence of a solution. In the test phase, a SVM

[1] Randomness is not necessary to achieve this gain itself.

takes a set of points and judges one to be positive or negative according to which side of the hyperplane it resides. Thus, a support vector machine is formalized as follows:

$$\begin{aligned} &\text{given } \{\boldsymbol{x}_i, y_i\}_i \subset \mathbb{R}^n \times \{1, -1\} \text{ and } C > 0\,, \\ &\text{minimize } |\boldsymbol{w}|^2 + C \sum_i \zeta_i \\ &\text{subject to } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \zeta_i, \zeta_i \geq 0\,. \end{aligned} \tag{1}$$

Here, ζ_i is the error corresponding to $\boldsymbol{x}_i$ and C is the weight of error. This is an instance of the quadratic optimization problem and there are many software packages that can solve this problem sufficiently fast, e.g., LIBSVM [4].

The above method has two drawbacks:

a) it captures the boundary between the classes only by a hyperplane and cannot, for example, separate intertwined input points by a curve;
b) it assumes the inputs to be points in Euclidean space.

A natural way to get around these is to prepare a map ϕ and apply SVM to $\{\phi(x_i), y_i\}_i$. The map ϕ is called the feature map and the range of ϕ, which is a Euclidean space of some dimension, is called the feature space. As for a), you can realize non-linear classification of inputs by mapping them by non-linear feature map and linearly classifying in the feature space. On the other hand, you can fix b) just by letting the domain of ϕ to be an arbitrary set S. If S is a general set, the notion of linearity is no longer clear but high dimensionality of the feature space corresponds to a kind of generalization of non-linearity for those cases. One can intuitively see this by observing the way a polynomial of degree greater than 1, a typical instance of non-linear map, increases the number of variables, e.g., x_1, x_2 to x_1^2, x_1x_2, x_2^2 by a quadratic map. But if one naïvely increases the dimension of the feature space, the cost of optimization, which, for example, involves the computation of $\boldsymbol{w}^\top \boldsymbol{x}_i$, will get more expensive accordingly. To avoid this, you can convert the SVM formulation (1) into the following dual form:

$$\begin{aligned} &\text{given } \{x_i, y_i\}_i \subset S \times \{1, -1\} \text{ and } C > 0\,, \\ &\text{maximize } \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j y_i y_j \alpha_i \alpha_j \phi(x_i)^\top \phi(x_j) \\ &\text{subject to } \sum_i y_i \alpha_i = 0, 0 \leq \alpha_i \leq C\,. \end{aligned} \tag{2}$$

Again, conventional software packages can solve this optimization problem in a reasonable amount of time and it is easy to convert the optimal solution for this problem into that for (1). The dual representation has the nice property that the objective function depends on $\{x_i\}_i$ only through the inner products of their feature vectors. Because of this property, you do not need to maintain feature vectors $\{\phi(x_i)\}_i$ explicitly as long as you can some how compute $K(x_i, x_j) := \phi(x_i)^\top \phi(x_j)$ efficiently. The map K is called the kernel function. We sometimes refer a kernel function as just kernel. A feature map's relevance for classification and the efficiency of kernel function computation are independent concepts.

2.2 Spectrum Kernel

Let Σ be the set of characters, i.e. a totally ordered set of cardinality $\sigma < \infty$. A string is an element of $\Sigma^* := \cup_{i=0}^{\infty} \Sigma^i$. We denote the length of a string x as $|x|$.

Below, we assume a fixed bijection between $\{1, 2, \ldots, \sigma^k\}$ and k-mers, and call the k-mer corresponding to i as the i-th k-mer. The k-spectrum kernel [11] is a kernel whose corresponding feature map maps a string x as

$$\phi_k : \Sigma^* \ni x \mapsto \sum_{s:k\text{-mer in } x} \mathbf{1}_s \in \mathbb{Z}^{\sigma^k}$$

where $\mathbf{1}_s$ is the vector whose i-th entry is 1 if s is the i-th k-mer or 0 otherwise. In other words, the i-th entry of $\phi_k(x)$ is the number of the occurrences of the i-th k-mer in x.

You can compute the kernel function $K_k(x, y) := \phi_k(x)^\top \phi_k(y)$ in optimal $\Theta(|x| + |y|)$-time by traversing the generalized suffix tree for x and y. [23,5,11] In practice, it is more desirable to use the suffix array [16] than the suffix tree because it is several times smaller. The suffix array is a data structure closely related to the suffix tree and in most cases, it is easy to translate an algorithms based on one of these data structures into the version based on the other with no or little overhead. In this case, you can trivially translate the traversal over the suffix tree into the scan on the suffix array with the use of an auxiliary data structure called the height array, which you can also compute in $\Theta(|x|+|y|)$-time [7]. Not only does this translation work without any overhead, it even makes the algorithm run faster in practice because the memory access pattern of array scan is optimal in terms of cache efficiency.

2.3 Wildcard Kernel

Let $*$ denotes the wildcard, a special character that can match any other character. If a k-mer $s \in \Sigma^k$ and a k-mer possibly including wildcards $t \in (\Sigma \cup \{*\})^k$ satisfy $s[i] = t[i]$ for $\forall i$ s.t. $t[i] \in \Sigma$, we denote $s \approx t$.

The (k, m)-wildcard kernel [13] is the kernel defined by the following feature map:

$$\phi_{(k,m)} \Sigma^* \ni x \mapsto \sum_{s:k\text{-mer in } x} \sum_{\substack{t \in (\Sigma \cup \{*\})^k \\ \#\{i : t[i] = *\} \leq m \\ s \approx t}} \mathbf{1}_t \in \mathbb{Z}^{(\sigma+1)^k}.$$

While this kernel achieves better learning performance than the k-spectrum kernel, the best bound obtained so far for the computation of the corresponding kernel function $K_{k,m}$ is $O(k^{m+1}(|x| + |y|))$ for inputs x and y.

3 Gapped Spectrum Kernel

3.1 Definition and Computation

Let w and k be integers s.t. $1 \leq w \leq k$. Let b be a binary sequence of length k containing w $\mathtt{1}$'s in it and i_j be the index of the j-th $\mathtt{1}$ in b, i.e. $\{i_j\}_{j=1}^{w}$ is the subsequence of $\{1, \ldots, k\}$ s.t. $b[i] = \mathtt{1}$ iff $\exists l$ with $i = i_l$. Let $mask_b$ be the map $\Sigma^k \ni x_1 x_2 \ldots x_k \mapsto x_{i_1} x_{i_2} \ldots x_{i_w} \in \Sigma^w$.

The b-gapped spectrum kernel is the kernel corresponding to the following feature map:

$$\phi_b : \Sigma^* \ni x \mapsto \sum_{i=1}^{|x|-k+1} \mathbf{1}_{mask_b(x_i x_{i+1} \ldots x_{i+k-1})} \in \mathbb{Z}^{\sigma^w}$$

where, like in Subsection 2.2, we are assuming a fixed bijection between $\{1, 2, \ldots, \sigma^w\}$ and w-mers.

This feature map is a natural generalization of that of the k-spectrum kernel, and the corresponding kernel function $K_b : (x, y) \mapsto \phi_b(x)\phi_b(y)$ can be calculated analogously by using the b-suffix array data structure [20]. Let T be a string of length n and T_i denotes the i-th suffix of T, i.e. $T[i]T[i+1] \ldots T[n]$. The b-suffix array $b\text{-}SA[1, \ldots, n]$ of T is the unique permutation of $\{1, \ldots, n\}$ s.t. $\{mask_b(T_{b\text{-}SA[i]})\}_i$ is lexicographically increasing breaking the tie by $b\text{-}SA[i]$. In $b\text{-}SA$, for $1 \leq \forall l \leq w$, the suffixes sharing the i_k-th characters for $1 \leq \forall k \leq l$ are laid contiguously. The b-height array $b\text{-}Hgt[1, \ldots, n]$ of T is an array s.t. $b\text{-}Hgt[i]$ is the length of the longest common prefix of $mask_b(T_{b\text{-}SA[i]})$ and $mask_b(T_{b\text{-}SA[i-1]})$ for $1 < \forall i \leq n$. $b\text{-}SA$ and $b\text{-}Hgt$ array contain information about suffixes of a string T and one can extend it for multiple numbers of strings $\{T_i\}_i$ just by replacing the T in the definitions above with the concatenation of T_i's with a special character $\mathtt{\$}$, which does not appear in any $\{T_i\}_i$, inserted between each other. In the following, we assume there are only two strings T_1 and T_2 for brevity.

For any w-mer y, one way to compute $\mathbf{1}_y$ is to find those i's that satisfy $mask_b(x_i x_{i+1} \ldots x_{i+k-1}) = y$ and these i's are forming contiguous regions in $b\text{-}SA$.[2] The boundaries between different regions correspond to those i's s.t. $b\text{-}Hgt[i] < w$. But this calculation involves enumeration of all w-mers. In most cases, one should not pay such an exponential cost because typically most w-mers never occur either in T_1 or T_2. Instead, we can enumerate only those w-mers that appear at least once in the inputs by directly scanning $b\text{-}SA$. And as already mentioned, those i's that contribute to $\mathbf{1}_y$ for some y are already collected in $b\text{-}SA$. One can judge which of T_1 ond T_2 a suffix $b\text{-}SA[i]$ belongs to just by comparing it with $|T_1|$. Therefore, scanning once on $b\text{-}SA$ is enough to compute all the ϕ_b, and thus K_b.

The running time is dominated by the time to build the b-suffix array and the b-height array, each of which is bounded by $O(h(|x| + |y|))$ where h is the

[2] Except, of course, that i's corresponding to the suffixes in T_1 and T_2 are mixed.

number of the consecutive `1`'s in b. In particular, $h \leq \min\{w, k-w\}$, which is, in practice, a good bound because the number of `0`'s $k-w$ is often much smaller than w.

3.2 Multiple Gapped Spectrum Kernel

The following equation links the gapped spectrum kernel and the wildcard kernel:

$$K_{k,m}(x,y) = \sum_{\substack{b \text{ of length } k \\ k-w \leq m}} K_b(x,y). \tag{3}$$

Note that adding the kernel functions for different b's is equivalent to concatenating the feature vectors charging the approximate matches with different positions of mismatch independently, which is exactly what the wildcard kernel does.

From this we define the multiple gapped spectrum kernels as follows. k, m and some order on $B_{k,m} := \{b \text{ of length } k : k-w \leq m\}$ are assumed as parameters and let b_i be the i-th element of $B_{k,m}$. We call $\Sigma_{i=1}^{j} K_{b_i}$ as the j-th multiple gapped spectrum kernel. From equation (3), the last element of the sequence matches the wildcard kernel. Because the $i+1$-th multiple gapped spectrum kernel can be computed in $O(m(|x|+|y|))$-time from the i-th as mentioned in Subsection 3.1, this gives $O(k^m m(|x|+|y|))$ bound for the wildcard kernel computation, which is k/m-factor smaller than the best existing one.

4 Experiments

4.1 Protein Classification

We applied the multiple gapped spectrum kernels to automatic classification of proteins, one of the most important problems in computational biology today.

We took the data set from the SCOP database [18,3]. SCOP had been used in many studies of string kernels [6,11,12,13] and is the de facto standard data set in this area. SCOP is a hierarchical database, namely, all proteins in it are first classified into Classes, then members of each Class are further classified into Folds, then into Superfamilies, Families and so on. Previous studies focused on the prediction of membership for Superfamilies. In other words, they predicted weather a given protein sequence is a member of a Superfamily or not. Recognizing the boundaries of Superfamilies is indeed important because it is essential to find a new Family under a known Superfamily. But on the other hand, one should also care about the recognition of boundaries of Families because, with the growth of the database, the probability that a given unknown protein is a member of some known Family is increasing. This is especially true for large Families. Thus, we predicted the membership for large Families here.

4.2 Experimental Design

We were interested in if it is possible to apply the multiple gapped spectrum kernels to classify proteins as accurate as, say, the wildcard kernel but faster. In particular, we wanted to know how randomness can help it because we do not have enough time to perform a heavy pre-computation such as learning 'the best' parameter if any. Having considered this requirement, we applied the following procedures for each Family F of the 13 Families including at least 50 protein sequences:

1. split the whole database into two groups, namely, the training set R and the test set E;
2. train the SVM with $R \cap F$ as the positive data set and $R \setminus F$ as the negative data set;
   (a) for each binary string b of length 5 with at most 2 `0`'s, calculate the b-gapped spectrum kernel;
   (b) generate 10 independently random permutations of binary strings of length 5 with at most 2 `0`'s;
   (c) continuously add gapped spectrum kernels according to the permutation giving rise to a multiple gapped spectrum kernels for each permutation;
3. test the SVM with $E \cap F$ as the positive data set and $E \setminus F$ as the negative data set trying every kernel of every sequence.

Note that each set of the multiple gapped spectrum kernels converge to the $(5, 2)$-wildcard kernel because of the equation (3). Thus, this experiment yields a kind of interpolations of the gapped spectrum kernel and the wildcard kernel.

As for the performance evaluation, we used the area under the receiver operation characteristic (ROC) curve [17]. ROC curve is the curve drawn by plotting the false positive rate as the x-coordinate and the true positive rate as the y-coordinate shifting the threshold of classification as a parameter. As we relax the threshold and become more likely to judge one data positive both the true positive rate and the true negative rate increase, producing an upper trend ROC curve. Because the increase in terms of the y-coordinate is a good thing while that of the x-coordinate is a bad thing, the more ROC curve extends to the upper-left the better. Thus, the goodness is quantized as the area under ROC curve (AUC), which is some value between 0 and 1. Note that the expectation of AUC of classification by uniformly random guess is 0.5, not 0.

In order to assess the effectiveness of the random choice, we also tried a seemingly better yet time expensive method and compared the result with the results of the above experiment. At this time, we conduct 2-fold cross validation in the training set R. Then, we obtained multiple gapped spectrum kernels by greedily adding the gapped spectrum kernels in the order of the score from the highest to the lowest.

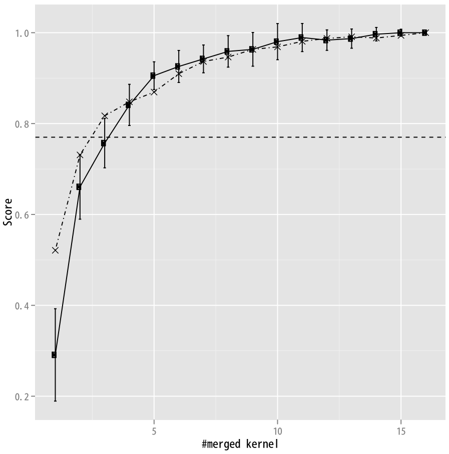
Also, we tried the k-spectrum kernels for $3 \leq k \leq 5$ because if the k-spectrum kernel was better than the gapped spectrum kernel or any other kernel, there would be no reason to use that kernel in the first place.

4.3 Results

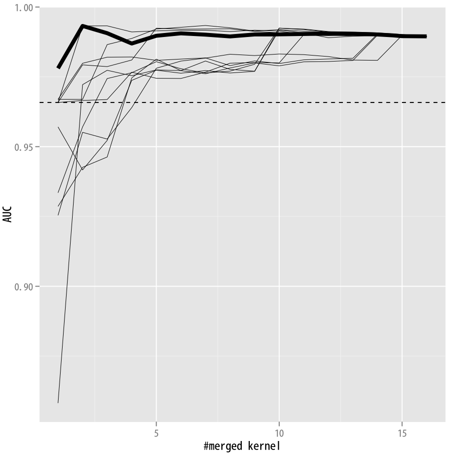
Due to the limit of the space, we show, at Figure (a) and Figure (b), the summary and, as an example of the result for each Family, the result for Family b.1.1.1 while putting the rests in the appendix.

Before explanations and discussions, let us mention that the Figure (a) is meant to convey the gist of the results as quickly as possible to the readers and is not suitable for careful analyses. In particular, it went through some post-processes, about which we will explain shortly. Please refer to the result for each Family for serious examination.

The original result we obtained for each of the 13 Families looks like Figure (b). In each of these figures, the horizontal axis corresponds to the number of the merged kernels and the vertical axis corresponds to the AUC. The thin lines, the thick line and the dashed horizontal line, represent the performances of the random order merges, the greedy order merge and the best k-spectrum kernel respectively. The summary shown in Figure (a) is a kind of 'average' of the results over all Families. The solid line, the dot-dashed line and the horizontal dashed line each corresponds to the 'average' performance of the random order merge, the greedy order merge and the best k-spectrum kernel respectively. Besides its construction, Figure (a) clearly demonstrates a) the fast convergence of the sequence of the random order merged multiple gapped spectrum kernels to the wildcard kernel, the rightmost dot; b) good competitiveness of random order strategy against greedy order strategy. Therefore, we claim that we can randomly merge a small number, say s, of gapped spectrum kernels and use the resulting multiple gapped spectrum kernel instead of the wildcard kernel, saving the computation time by the factor of $\sum_{i=0}^{m} \binom{k}{i}/s$ in expectation and still be able to expect comparably accurate results. There are several Families for which the k-spectrum kernel performs very well, e.g., c.2.1.2 (Figure (g)), c.37.1.8



(a) Summary of the Results



(b) Result for Family b.1.1.1

(Figure (h)). This is not a problem because the computation of the k-spectrum kernel is cheap and we can just try them first if we want.

Last, we describe how Figure (a) was derived. For each result for a Family, the horizontal axis was normalized so that the score for the wildcard kernel would be 1 and the average score of the k-spectrum kernels would be 0. Then, thin lines were averaged over all the trials. After that, all the components were averaged over all the Families. The thick lines were changed to the dot-dashed line for visibility. The vertical bars stabbing circles represent the average standard deviations.

5 Related Works

In this section, we briefly summarize existing string kernels relevant to our results. See also Table 1.

Leslie et al. [11] introduced the spectrum kernel in the context of protein remote homology search problem. This kernel has an integer parameter $k > 0$ and called the k-spectrum kernel when k should be specified explicitly. The feature it captures is the number of each k-mer included in the string. As explained in Subsection 2.2, the corresponding kernel function can be computed in the optimal $\Theta(x+y)$-time. But the accuracy of classification by the spectrum kernel is not high. Thus, many similar[3] kernels were developed since then in the context of achieving both fast computation and high learning performance at the same time. The (k,m)-mismatch kernel [10] considers the number of each k-mer that appears in the string with at most $m(< k)$ mismatches as the feature. The computation of the kernel function takes $O(\alpha_{k,m}(x+y))$-time where $\alpha_{k,m} := \sum_{l=0}^{\min(2m,k)} \binom{k}{l}(k-l)$ [9]. The k-subsequence kernel [15] considers the number of each k-mer that appears as subsequences in the string as the feature.[4] It takes $O(kxy)$-time to compute the corresponding kernel function. With an additional assumption that the region in the original sequence spanned by each subsequence occurrence must be at most g, the (g,k)-gappy string kernel is obtained and the time needed to compute the kernel function drops to $O(\beta_{g,k}(x+y))$-time where $\beta_{g,k} := (g-k)g^{g-k+1}$ [13]. The (k,m)-wildcard kernel [13], like the (k,m)-mismatch kernel, is based on the counts of occurrences of k-mers in the input allowing at most m mismatches. The difference is that only the wildcard kernel is conscious of where the mismatches occur (see Section 2 for the detail). The kernel function computation takes $O(k^{m+1}(x+y))$-time. As for the learning performance, these kernels are as good as each other [13].

Kuksa et al. [9] also gave bounds for the (g,k)-restricted gappy kernel and (k,m)-wildcard kernel that are, except for some minor terms and truncations in analyses, the same as the ones above.

[3] The common property is that they are based on the statistics of some substructure.

[4] In the original paper [15], it is called 'string subsequence kernel' with k implicitly assumed.

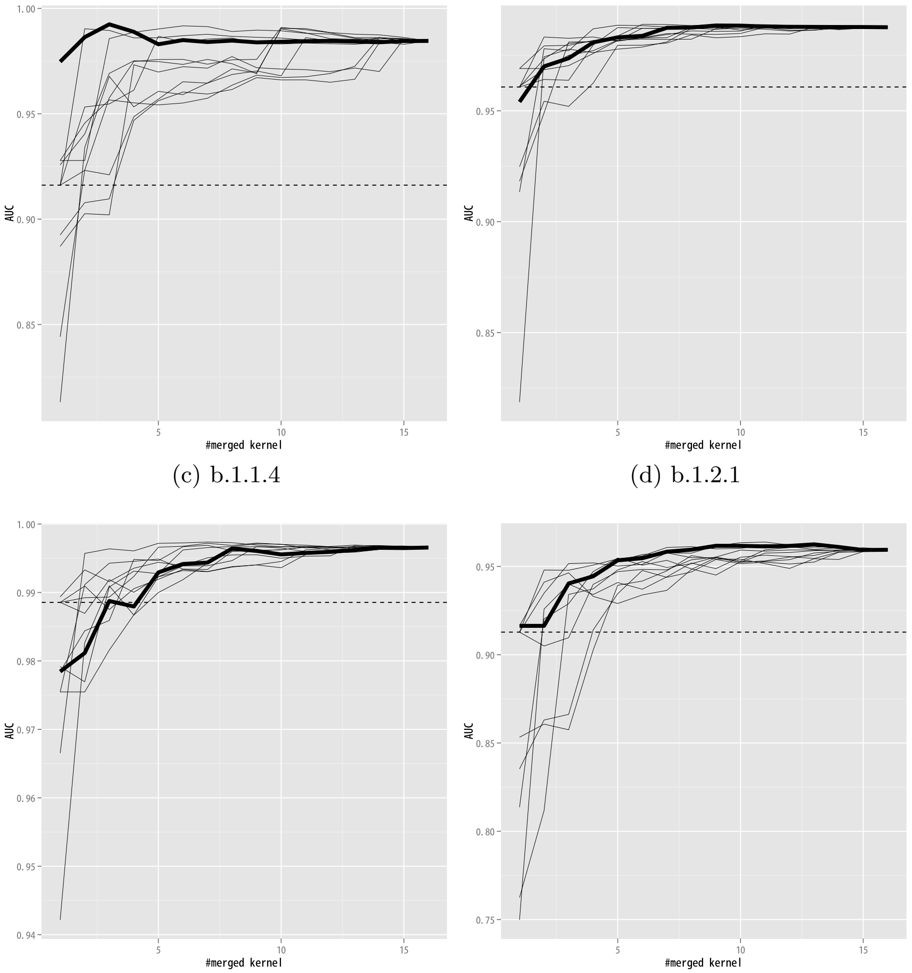
Acknowledgement. This work was partially supported by the Grant-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan.This research was also supported by JST, ERATO, Kawarabayashi Large Graph Project. The SVM computation in the experiment was done by LIBSVM [4] software. SCOP [18,3] ver1.75a was retrieved from `http://scop.berkeley.edu`. The super-computing resource was provided by Human Genome Center, Institute of Medical Science, University of Tokyo.

References

1. Ben-Hur, A., Ong, C.S., Sonnenburg, S., Schölkopf, B., Rätsch, G.: Support vector machines and kernels for computational biology. PLoS Computational Biology 4(10), e1000173 (2008)
2. Asa, B.-H., Noble, W.S.: Kernel methods for predicting protein-protein interactions. In: ISMB (Supplement of Bioinformatics), pp. 38–46 (2005)
3. Chandonia, J.-M., Hon, G., Walker, N.S., Conte, L.L., Koehl, P., Levitt, M., Brenner, S.E.: The ASTRAL Compendium in 2004. Nucleic Acids Research 32(Database-Issue), 189–192 (2004)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM TIST 2(3), 27 (2011)
5. Farach, M.: Optimal Suffix Tree Construction with Large Alphabets. In: FOCS, pp. 137–143. IEEE Computer Society (1997)
6. Jaakkola, T., Diekhans, M., Haussler, D.: Using the Fisher Kernel Method to Detect Remote Protein Homologies. In: Lengauer, T., Schneider, R., Bork, P., Brutlag, D.L., Glasgow, J.I., Mewes, H.-W., Zimmer, R. (eds.) ISMB, pp. 149–158. AAAI (1999)
7. Kasai, T., Lee, G., Arimura, H., Arikawa, S., Park, K.: Linear-Time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications. In: Amir, A., Landau, G.M. (eds.) CPM 2001. LNCS, vol. 2089, pp. 181–192. Springer, Heidelberg (2001)
8. Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., Leslie, C.S.: Profile-Based String Kernels for Remote Homology Detection and Motif Extraction. In: CSB, pp. 152–160. IEEE Computer Society (2004)
9. Kuksa, P.P., Huang, P.-H., Pavlovic, V.: Scalable Algorithms for String Kernels with Inexact Matching. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) NIPS, pp. 881–888. Curran Associates, Inc. (2008)
10. Leslie, C.S., Eskin, E., Cohen, A., Weston, J., Noble, W.S.: Mismatch string kernels for discriminative protein classification. Bioinformatics 20(4), 467–476 (2004)
11. Leslie, C.S., Eskin, E., Noble, W.S.: The Spectrum Kernel: A String Kernel for SVM Protein Classification. In: Pacific Symposium on Biocomputing, pp. 566–575 (2002)
12. Leslie, C.S., Eskin, E., Weston, J., Noble, W.S.: Mismatch String Kernels for SVM Protein Classification. In: Becker, S., Thrun, S., Obermayer, K. (eds.) NIPS, pp. 1417–1424. MIT Press (2002)
13. Leslie, C.S., Kuang, R.: Fast Kernels for Inexact String Matching. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 114–128. Springer, Heidelberg (2003)
14. Liao, L., Noble, W.S.: Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In: RECOMB, pp. 225–232 (2002)

15. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.J.C.H.: Text Classification using String Kernels. Journal of Machine Learning Research 2, 419–444 (2002)
16. Manber, U., Myers, G.: Suffix Arrays: A New Method for On-Line String Searches. In: Johnson, D.S. (ed.) SODA, pp. 319–327. SIAM (1990)
17. Metz, C.E.: Basic principles of ROC analysis. Seminars in Nuclear Medicine 8(4), 283–298 (1978)
18. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: A structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology 247(4), 536–540 (1995)
19. Noble, W.S., Kuehn, S., Thurman, R.E., Yu, M., Stamatoyannopoulos, J.A.: Predicting the in vivo signature of human gene regulatory sequence. In: ISMB (Supplement of Bioinformatics), pp. 328–343 (2005)
20. Onodera, T., Shibuya, T.: An Index Structure for Spaced Seed Search. In: Asano, T., Nakano, S., Okamoto, Y., Watanabe, O. (eds.) ISAAC 2011. LNCS, vol. 7074, pp. 764–772. Springer, Heidelberg (2011)
21. Swamidass, S.J., Chen, J.H., Bruand, J., Phung, P., Ralaivola, L., Baldi, P.: Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. In: ISMB (Supplement of Bioinformatics), pp. 359–368 (2005)
22. Vapnik, V.: Statistical learning theory (1998)
23. Weiner, P.: Linear Pattern Matching Algorithms. In: SWAT (FOCS), pp. 1–11. IEEE Computer Society (1973)

Appendix: Full Results of the Experiment

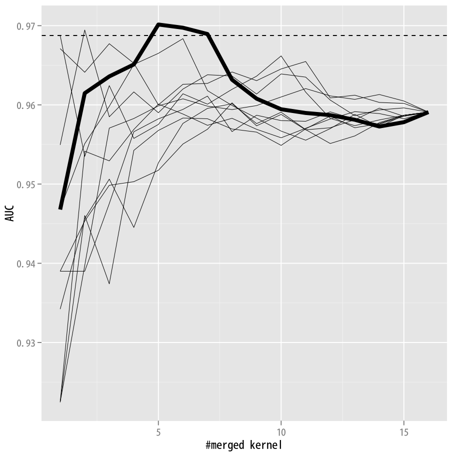


(c) b.1.1.4

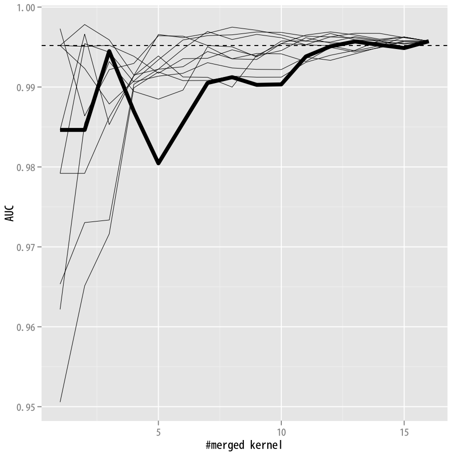
(d) b.1.2.1

(e) b.36.1.1

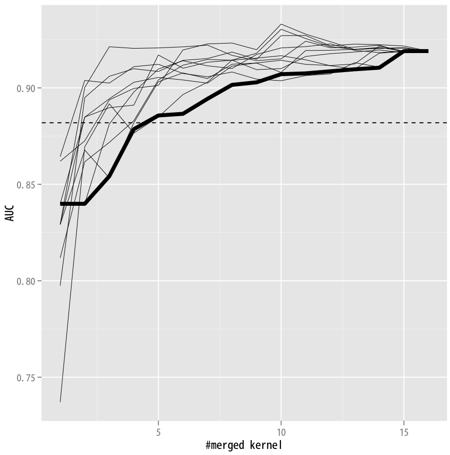
(f) b.40.4.5



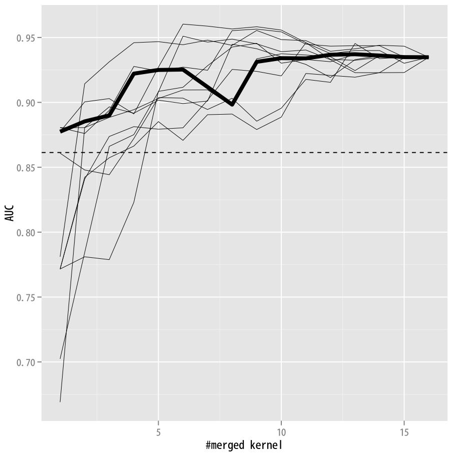
(g) c.2.1.2



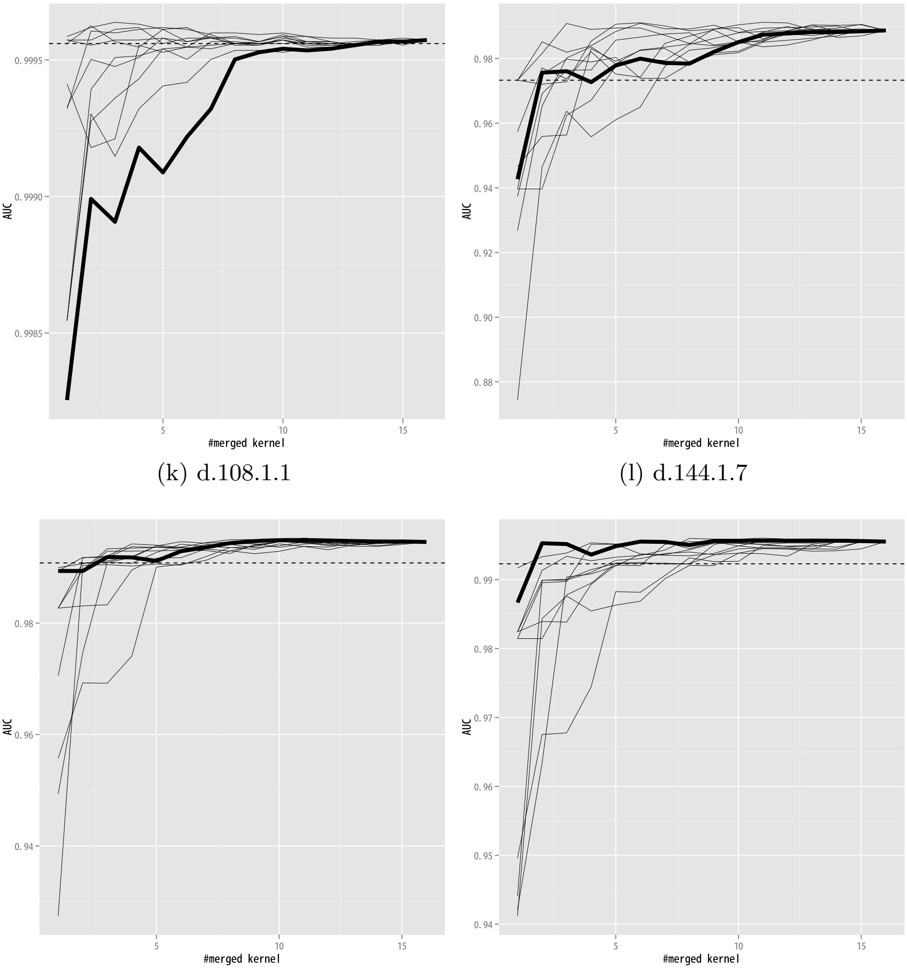
(h) c.37.1.8



(i) c.94.1.1



(j) d.58.7.1



(k) d.108.1.1

(l) d.144.1.7

(m) g.37.1.1

(n) g.39.1.3

Typhoon Damage Scale Forecasting with Self-Organizing Maps Trained by Selective Presentation Learning

Kazuhiro Kohara and Isao Sugiyama

Chiba Institute of Technology
2-17-1 Tsudanuma, Narashino, Chiba 275-0016, Japan
kohara.kazuhiro@it-chiba.ac.jp

Abstract. We previously proposed a new typhoon warning system which forecasts the likely extent of damage associated with a typhoon towards humans and buildings. The relation between typhoon data and damage data was learned by self-organizing maps (SOM) and typhoon damage scale (small, middle or large) was forecast by the SOM using typhoon data. Although average accuracy for actually small scale damage data was comparatively high (96.2%), average accuracy for actually large scale damage data was comparatively low (65.2%). Thus, we apply a selective presentation learning technique for improving the predictability of large scale damage by SOM. Learning data corresponding to middle and large scale damage are presented more often. Average accuracy for actually large scale damage data was increased by about 9%. The accuracy for actually large scale of numbers of fatalities and houses under water was increased by 25% and 20%, respectively.

Keywords: typhoon damage scale forecasting, self-organizing maps, selective presentation learning, multiple regression analysis, decision trees.

1 Introduction

Intelligent techniques such as back-propagation neural networks (BPNN) [1], self-organizing maps (SOM) [2], decision trees [3] and Bayesian networks [4] have been extensively investigated, and various attempts have been made to apply them to identification, prediction, control and so on [e.g., 1-12]. Harada et al. applied BPNN to forecasting typhoon course [9], Takada et al. applied BPNN to forecasting typhoon damage of electric power systems [10] and Udagawa et al. applied Bayesian networks to rain prediction [11]. We previously applied intelligent techniques to forecasting typhoon damage to human and buildings [12].

Damage caused by typhoons to both people and structures has decreased in Japan due to improvements of countermeasures against natural disasters, however, such damage still occurs [13, 14]. A typhoon warning that represents typhoon menace with high accuracy should be issued appropriately. A typical typhoon warning currently issued in Japan may be "This typhoon is large and very strong." We proposed a new typhoon warning which forecasts the risk of damage scale to both human and

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 16–26, 2013.

buildings as follows: “We forecast that the coming typhoon has a risk of causing both large scale human and building damage. Please take care.”

We investigate relation between typhoon data and damage data and forecast typhoon damage using typhoon data. The typhoon data includes the month when the typhoon was born, latitude and longitude where the typhoon was born, lowest atmospheric pressure, maximum wind speed and total precipitation. Damage data includes human damage data such as number of fatalities and injured persons and building damage data such as number of completely destroyed houses and number of houses under water. The types of typhoon and damage data are shown in Table 1. There are nine types of typhoon data and nine types of damage data, divided into three types of human damage and six types of building damage.

Our previous work and its problem are described in Chapter 2. Typhoon damage scale forecasting with equal presentation (conventional) learning and its problem are described in Chapter 3. Typhoon damage scale forecasting with selective presentation (proposed) learning and its effectiveness are described in Chapter 4.

2 Previous Work and Its Problem

2.1 Forecasting Damage Data Using Typhoon Data

Our previous work [12] is reviewed here briefly. We used SOM [2], multiple regression analysis and decision trees [3] for typhoon damage forecasting. *Viscovery SOMine 4.0* was used as SOM software and *See5 release 1.19* was used as decision tree software with default parameter values. 139 data records of typhoon data and damage data from June 1981 to September 1999 were collected from the typhoon database [15, 16]. We used 111 data records (to September 1995) for learning and 28

Table 1. Types of typhoon data and damage data used in this study

Typhoon data	Month when the typhoon was born, Latitude and longitude where the typhoon was born, Lowest atmospheric pressure, Maximum wind speed, Total, one-hour and twenty-four-hour precipitation, Life span
Damage data	Number of fatalities, Number of injured persons, Number of dead and injured persons
	Number of completely destroyed houses, Number of half destroyed houses, Number of partially destroyed houses, Total number of damaged houses, Number of houses under water, Total number of destroyed non-house structures

data records (from July 1996) for testing. The average and maximum of every type of damage data in learning data are shown in Table 2. The minimum of every damage type was zero. We considered two types of typhoon damage forecasting: two-class (*yes* or *no*) and three-class (*small*, *middle* or *large* scale) damage forecasting.

Table 2. Average and maximum of every type of damage data

Data type	Average	Maximum
Number of fatalities	6.0	100
Number of injured persons	34.2	1499
Number of dead and injured persons	40.2	1561
Number of completely destroyed houses	21.7	541
Number of half destroyed houses	2260.4	169877
Number of partially destroyed houses	343.3	20303
Total number of damaged houses	2625.4	170418
Number of houses under water	8319.8	174124
Total no. destroyed non-house structures	60.6	3029

2.2 Two-Class *(yes* or *no*) Damage Forecasting

In two-class damage forecasting, a predictor is trained by two values (0 and 1). In this case, 0 means that the damage is zero (*no*) and 1 means that the damage is not zero (*yes*). Experiments were made with nine types of continuous (or raw) typhoon data as inputs and one damage data (two values) as an output. Here, we expect that typhoon data such as lowest atmospheric pressure, maximum wind speed, precipitation and life span can be forecast with high accuracy by a weather forecasting system such as Japanese SYNFOS [17] and hence actual typhoon data was used as inputs.

In testing with SOM, we input nine types of continuous typhoon data to SOM and recall one damage data using the map. In other words, we find the neuron which has the nearest weight vector to nine types of continuous typhoon data and obtain one damage data from the nearest neuron. The average accuracy of two-class (*yes* or *no*) damage forecasting for the three intelligent methods is shown in Table 3. Here, average accuracy means the average of the accuracy of nine damage data. This experiment confirmed that damage data are well related with typhoon data and that SOM learned the nonlinear relation very well. The accuracy with SOM was much better than that with MR and DT.

Table 3. Average accuracy of two-class (*yes* or *no*) damage forecasting

Method	Learning data	Test data
SOM	100%	93.3%
Multiple regression (MR)	70.9%	70.2%
Decision trees (DT)	77.7%	63.9%

2.3 Three-Class (*small*, *middle* or *large* scale) Damage Forecasting

In three-class damage forecasting, a predictor was trained by three values (0, 1 and 2). In the learning data, 0 means that the damage is *small scale*, 1 means the damage is *middle scale* and 2 means the damage is *large scale*. The average of each damage data was calculated as shown in Table 2. *Small scale* corresponded to under half of the average, *medium scale* corresponded to between half of the average and the average, and *large scale* corresponded to over the average, respectively. The prediction was considered accurate when the predicted scale was equal to the actual scale. The average accuracy of three-class damage forecasting was shown in Table 4. The accuracy with SOM was also much better than that with MR and DT.

Table 4. Average accuracy of three-class (*small*, *middle* or *large* scale) damage forecasting

Method	Learning data	Test data
SOM	100%	93.3%
Multiple regression (MR)	76.7%	62.7%
Decision trees (DT)	88.6%	78.2%

2.4 Problem

In previous work [12], the percentage of small scale damage was comparatively high (83.2% of learning data) and the percentage of middle scale damage was comparatively low (5.2% of learning data). The percentage of large scale damage was 11.6%. Therefore, we modified the range of scales as follows: *small scale* corresponds to under the 25% point of learning data, *middle scale* corresponds to between 25% and the 75% points, and *large scale* corresponds to over the 75% point. When calculating the 25% and 75% points, zero damage data was excluded. The 25% and 75% points of every damage type are shown in Table 5. After modification of the range of scales, percentages of *small*, *middle* and *large scale* damage are 65.0%, 23.2% and 11.8%, respectively. The numbers of each scale of every damage data are shown in Table 6.

Table 5. 25% and 75% points of damage data

Data type	25% point	75% point
Number of fatalities	2	17
Number of injured persons	4	63
Number of dead and injured persons	5	74
Number of completely destroyed houses	3	31
Number of half destroyed houses	5	520
Number of partially destroyed houses	7	92
Total number of damaged houses	18	399
Number of houses under water	8	3773
Total number of destroyed non-house structures	7	61

Table 6. Number of every damage data in learning data

Data type	Small	Middle	Large
Number of fatalities	84	17	10
Number of injured persons	76	23	12
Number of dead and injured persons	74	25	12
Number of completely destroyed houses	76	23	12
Number of half destroyed houses	75	24	12
Number of partially destroyed houses	70	28	13
Total number of damaged houses	63	32	16
Number of houses under water	57	36	18
Total number of destroyed non-house structures	74	24	13
Total number	649	232	118
Percentage	65.0%	23.2%	11.8%

3 Typhoon Damage Scale Forecasting with Equal Presentation and Its Problem

We used 111 data records (from June 1981 to September 1995) for learning in the same way as previous work [12] and used 86 data records (from July 1996 to September 2008) for testing. 58 recent data records (from September 2000 to September 2008) were added to testing. All training data are presented equally to SOM. Average accuracy of two-class (yes or no) damage forecasting with SOM for 86 test data was 96.0%. Average accuracy of three-class damage scale forecasting was 86.8%, as shown in Table 7. Although average accuracy for actually small scale damage data was comparatively high (96.2%), average accuracy for actually large scale damage data was comparatively low (65.2%), as shown in Table 8.

Table 7. Accuracy of three-class damage scale forecasting for 86 test data with equal presentation

Data type	SOM
Number of fatalities	86.0%
Number of injured persons	87.2%
Number of dead and injured persons	86.0%
Number of completely destroyed houses	87.2%
Number of half destroyed houses	90.7%
Number of partially destroyed houses	84.9%
Total number of damaged houses	88.4%
Number of houses under water	84.9%
Total number of destroyed non-house structures	86.0%
Average	86.8%

Table 8. Accuracy of each damage scale forecasting for 86 test data with equal presentation

Data type	Small	Middle	Large
Number of fatalities	98.2%	66.7%	50.0%
Number of injured persons	97.6%	77.8%	76.5%
Number of dead and injured persons	93.3%	87.0%	61.1%
Number of completely destroyed houses	92.6%	89.5%	61.5%
Number of half destroyed houses	94.3%	88.5%	71.4%
Number of partially destroyed houses	100%	83.3%	63.0%
Total number of damaged houses	100%	79.3%	77.8%
Number of houses under water	93.3%	85.4%	66.7%
Total number of destroyed non-house structures	96.0%	84.2%	58.8%
Average	96.2%	82.4%	65.2%

4 Selective Presentation Learning for Typhoon Damage Scale Forecasting

Generally, the predictability of large scale damage is more important than that of small scale damage. When all training data are presented equally as in the conventional approach, SOM will learn the small and large scale damage equally well and cannot learn the large scale damage more effectively. We previously proposed the selective-presentation and selective-learning-rate approaches for improving the ability of BPNN and SOM to predict large changes and applied them into stock market prediction [18-20] and foreign exchange rate prediction [21]. In the selective-presentation approach, the training data corresponding to large changes in the prediction-target time series are presented more often [18, 21]. In the selective-learning-rate approach, the learning rate for training data corresponding to small changes is reduced [19-21].

We apply the selective presentation learning into typhoon damage scale forecasting by SOM. To allow SOM to learn about large scale damage more effectively, we separate the learning data into large scale and small scale damage data. As shown in Table 6, the number of small scale damage data is comparatively large (65.0%) and the numbers of middle and large scale data are comparatively small (23.2% and 11.8%, respectively). Therefore, the number of presentations of middle and large scale data was increased in order to decrease the difference of number of presentations for each scale damage data, as shown in Table 9. For example, "* 5" means five times presentation per one learning cycle. The percentage of total number of presentations per one learning cycle for small, middle and large scale damage data was almost the same (33.9%, 33.2% and 32.9%, respectively), as also shown in Table 9.

Table 9. Number of presentations of every damage data in selective presentation learning

Data type	Small	Middle	Large
Number of fatalities	84 * 1 = 84	17 * 5 = 85	10 * 8 = 80
Number of injured persons	76 * 1 = 76	23 * 3 = 69	12 * 6 = 72
Number of dead and injured persons	74 * 1 = 74	25 * 3 = 75	12 * 6 = 72
Number of completely destroyed houses	76 * 1 = 76	23 * 3 = 69	12 * 6 = 72
Number of half destroyed houses	75 * 1 = 75	24 * 3 = 72	12 * 6 = 72
Number of partially destroyed houses	70 * 1 = 70	28 * 2 = 56	13 * 5 = 65
Total number of damaged houses	63 * 1 = 63	32 * 2 = 64	16 * 4 = 64
Number of houses under water	57 * 1 = 57	36 * 2 = 72	18 * 3 = 54
Total number of destroyed non-house structures	74 * 1 = 74	24 * 3 = 72	13 * 6 = 78
Total number of presentations per cycle	649	634	629
Percentage	33.9%	33.2%	32.9%

5 Typhoon Damage Scale Forecasting with Selective Presentation and Its Effectiveness

Forecasting results with selective presentation are shown in Tables 10 and 11. Although average accuracy of three-class damage scale forecasting was decreased (79.6%, as shown in Table 10), average accuracy for actually large scale damage data was increased (74.0%, as shown in Table 11).

We evaluated two-class (*large* or *not large*) damage scale forecasting with equal and selective presentation. Forecasting results are shown in Tables 12 to 14. Average accuracy of two-class damage scale forecasting with equal presentation was 93.4%, as shown in Table 12. Although average accuracy for actually small or middle (not large) scale damage data with equal presentation was comparatively high (99.5%), average accuracy for actually large scale damage data was comparatively low (65.2%), as shown in Table 13. Although average accuracy of two-class damage scale forecasting with selective presentation was decreased by 1.2% (92.2%, as shown in Table 14), average accuracy for actually large scale damage data was increased by 8.8% (74.0%, as shown in Table 15). Average accuracy for actually small or middle (*not large*) scale damage data with selective presentation was still high (96.3%, as shown in Table 15).

Table 10. Accuracy of three-class damage scale forecasting for 86 test data with selective presentation

Data type	SOM
Number of fatalities	86.0%
Number of injured persons	82.6%
Number of dead and injured persons	74.4%
Number of completely destroyed houses	82.6%
Number of half destroyed houses	77.9%
Number of partially destroyed houses	73.3%
Total number of damaged houses	76.7%
Number of houses under water	77.9%
Total number of destroyed non-house structures	84.9%
Average	79.6%

Table 11. Accuracy of each damage scale forecasting for 86 test data with selective presentation

Data type	Small	Middle	Large
Number of fatalities	93.0%	71.4%	75.0%
Number of injured persons	92.9%	70.4%	76.5%
Number of dead and injured persons	82.2%	60.9%	72.2%
Number of completely destroyed houses	92.6%	68.4%	61.5%
Number of half destroyed houses	83.0%	69.2%	71.4%
Number of partially destroyed houses	85.4%	61.1%	63.0%
Total number of damaged houses	87.2%	58.6%	83.3%
Number of houses under water	86.7%	68.3%	86.7%
Total number of destroyed non-house structures	94.0%	68.4%	76.5%
Average	88.5%	66.3%	74.0%

Table 12. Accuracy of two-class (*large* or *not large*) damage scale forecasting for 86 test data with equal presentation

Data type	SOM
Number of fatalities	95.3%
Number of injured persons	95.3%
Number of dead and injured persons	91.9%
Number of completely destroyed houses	94.2%
Number of half destroyed houses	97.7%
Number of partially destroyed houses	88.4%
Total number of damaged houses	95.3%
Number of houses under water	90.7%
Total number of destroyed non-house structures	91.9%
Average	93.4%

Table 13. Accuracy of each damage scale forecasting for 86 test data with equal presentation

Data type	Not large (small or middle)	Large
Number of fatalities	100%	50.0%
Number of injured persons	100%	76.5%
Number of dead and injured persons	100%	61.1%
Number of completely destroyed houses	100%	61.5%
Number of half destroyed houses	100%	71.4%
Number of partially destroyed houses	100%	63.0%
Total number of damaged houses	100%	77.8%
Number of houses under water	95.8%	66.7%
Total number of destroyed non-house structures	100%	58.8%
Average	99.5%	65.2%

Table 14. Accuracy of two-class (*large* or *not large*) damage scale forecasting for 86 test data with selective presentation

Data type	SOM
Number of fatalities	97.7%
Number of injured persons	94.2%
Number of dead and injured persons	91.9%
Number of completely destroyed houses	93.0%
Number of half destroyed houses	94.2%
Number of partially destroyed houses	86.0%
Total number of damaged houses	90.7%
Number of houses under water	88.4%
Total number of destroyed non-house structures	94.2%
Average	92.2%

Table 15. Accuracy of two-class (*large* or *not large*) damage scale forecasting for 86 test data with selective presentation

Data type	Not large (small or middle)	Large
Number of fatalities	100%	75.0%
Number of injured persons	98.6%	76.5%
Number of dead and injured persons	97.1%	72.2%
Number of completely destroyed houses	98.6%	61.5%
Number of half destroyed houses	96.2%	71.4%
Number of partially destroyed houses	96.6%	63.0%
Total number of damaged houses	92.6%	83.3%
Number of houses under water	88.7%	86.7%
Total number of destroyed non-house structures	98.6%	76.5%
Average	96.3%	74.0%

Results of each damage scale forecasting were as follows: 1) the accuracy for actually large scale of numbers of fatalities was increased by 25%, 2) the accuracy for actually large scale of numbers of dead and injured persons was increased by 11.1%, 3) the accuracy for actually large scale of numbers of houses under water was increased by 20%, and 4) the accuracy for actually large scale of numbers of each damage was not decreased.

6 Conclusion

We investigated typhoon damage scale forecasting with SOM trained by selective presentation learning. Using nine types of typhoon data as inputs to SOM, learning data corresponding to middle and large scale damage were presented more often. Average accuracy for actually large scale damage data was increased by about 9%. The accuracy for actually large scale of numbers of fatalities and houses under water was increased 25% and 20%, respectively. As a result, a typhoon forecasting method is proposed as follows: 1) Evaluate two-class (*yes* or *no*) damage forecasting (average accuracy: 96.0%), 2) When two-class (*yes* or *no*) forecasting result is *yes,* evaluate two-class (*large* or *not large*) damage scale forecasting (average accuracy: 92.2%), 3) Issue a typhoon warning based on above two class (*large* or *not large*) damage scale forecasting. For example, such a warning may be issued, "According to the Japanese typhoon database, we forecast that the coming typhoon has a risk of causing both large scale human and building damage. Please take care." In further research, we will consider more detailed damage forecasting and use other predictors such as Bayesian networks and support vector machines. We will also investigate the theoretical aspect of the proposed method.

References

1. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation. In: Rumelhart, D., McClelland, J., the PDP Research Group (eds.) Parallel Distributed Processing, vol. 1. MIT Press, Cambridge (1986)
2. Kohonen, T.: Self-Organizing Maps. Springer (1995)
3. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
4. Jensen, F.: Bayesian Networks and Decision Graphs. Springer (2001)
5. Pham, D., Liu, X.: Neural Networks for Identification, Prediction and Control. Springer (1995)
6. Kohara, K.: Neural networks for economic forecasting problems. In: Leondes, C.T. (ed.) Expert Systems - The Technology of Knowledge Management and Decision Making for 21st Century. Academic Press, San Diego (2002)
7. Kohara, K., Aoki, K., Isomae, M.: Forecasting, Diagnosis and Decision Making with Neural Networks and Self-Organizing Maps. In: Rodic, A.D. (ed.) Automation and Control, Theory and Practice. In-Tech Publishing, Vienna (2009)
8. Kohara, K., Tsuda, T.: Creating Product Maps with Self-Organizing Maps for Purchase Decision Making. Transactions on Machine Learning and Data Mining 3(2), 51–66 (2010)

9. Harada, H., Momma, E., Ishii, H., Ono, T.: Forecast of typhoon course using multi-layered neural network (III). In: Proceedings of National Convention of the Institute of Electrical Engineers of Japan, Toyama, vol. 3, p. 111 (2007)
10. Takata, H., Kawaji, S., Ha, T.: Study on a Prediction Method of Typhoon Damage of Electric Power Systems in each District on the Main Island in Kagoshima Prefecture. Technical Report 48, Faculty of Engineering, Kagoshima University (2006)
11. Udagawa, S., Nishio, S., Kimura, M.: Rain prediction by the Bayesian network. In: Proceedings of National Convention of the Information Processing Society of Japan, vol. (3), pp. 237–238 (2005)
12. Kohara, K., Hasegawa, R.: Typhoon Damage Forecasting with Self-Organizing Maps, Multiple Regression Analysis and Decision Trees. In: Proceedings of 5th International Workshop on Artificial Neural Networks and Intelligent Information Processing, Milan, pp. 106–111 (2009)
13. Murayama, K.: Introduction to Typhoon Study. Yama-Kei Publishers (2006)
14. Nyoumura, Y.: Weather Damage Prediction and Countermeasure. Ohmsha (2002)
15. National Research Institute for Earth Science and Disaster Prevention (2008), `http://www.bosai.go.jp/index.html`
16. National Institute of Informatics (2008), `http://agora.ex.nii.ac.jp/digital-typhoon`
17. Japan Weather Association (2008), `http://www.jwa.or.jp/synfos/`
18. Kohara, K., Fukuhara, Y., Nakamura, Y.: Selective presentation learning for neural network forecasting of stock markets. Neural Computing & Applications 4(3), 143–148 (1996)
19. Kohara, K., Fukuhara, Y., Nakamura, Y.: Selectively intensive learning to improve large-change prediction by neural networks. In: Proceedings of International Conference on Engineering Applications of Neural Networks, pp. 463–466 (1996)
20. Kohara, K.: Selective-learning-rate approach for stock market prediction by simple recurrent neural networks. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2773, pp. 141–147. Springer, Heidelberg (2003)
21. Kohara, K.: Foreign Exchange Rate Prediction with Selective Learning BPNNs and SOMs. In: Proceedings of World Multi-Conference on Systemics, Cybernetics and Informatics, Orland, FL, pp. 350–354 (2005)

Dynamic-Radius Species-Conserving Genetic Algorithm for the Financial Forecasting of Dow Jones Index Stocks

Michael Scott Brown, Michael J. Pelosi, and Henry Dirska

University of Maryland University College, Adelphi Maryland, USA
michaels.brown@faculty.umuc.edu,
mpelosi@maui.net,
dirska@nova.edu

Abstract. This research uses a Niche Genetic Algorithm (NGA) called Dynamic-radius Species-conserving Genetic Algorithm (DSGA) to select stocks to purchase from the Dow Jones Index. DSGA uses a set of training data to produce a set of rules. These rules are then used to predict stock prices. DSGA is an NGA that uses a clustering algorithm enhanced by a tabu list and radial variations. DSGA also uses a shared fitness algorithm to investigate different areas of the domain. This research applies the DSGA algorithm to training data which produces a set of rules. The rules are applied to a set of testing data to obtain results. The DSGA algorithm did very well in predicting stock movement.

Keywords: Niche Genetic Algorithm, Genetic Algorithm, stock forecasting, financial forecasting, classification, black-box investing.

1 Introduction

Forecasting the price movements of stocks is a difficult task. The possible financial reward of picking the correct direction that a stock will move has created much interest in developing systems to predict such behavior. Early work on formal financial forecasting began in the early 1900's [1] and continues to this day [2]. In the last 20 years a variety of techniques have been used to predict stock movement. These include Genetic Algorithms (GAs), Neural Networks and other artificial intelligence techniques.

A variety of methods use GAs and Genetic Programming (GP) to predict stock and security movements [3-5]. These methods take different approaches. Some research uses GAs and GPs to develop classification rules, while others use GAs in hybrid approaches [2]. Much research has been done using these evolutionary approaches to perform black-box investing.

This paper presents a new system for financial forecasting using a Niche Genetic Algorithm (NGA). The presented research used an NGA and a set of financial data to derive a set of classification rules that the research later applied to another set of data. The training and test data each comes from a full quarter of stock prices from the

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 27–41, 2013.

Dow Jones Index. These stocks represent 30 of some of the largest companies in the United States.

2 Literature Review

Over the last few decades, hundreds of papers have been written on GAs. But only a small number of these studies use GAs to classify and forecast stock market data. While sparse, research that uses genetic algorithms for this purpose validates its effectiveness and value for future research. Medical research also supports using classifiers based on GAs.

The following literature review discusses these points and lists significant papers supporting them. The literature review also provides a brief explanation of genetic algorithms, lists a few significant early studies, and suggests that NGA's can be effective in a variety of domains.

2.1 Genetic Algorithms

GAs are very useful algorithms that can locate optima within very complex domains. Early research in the subject began the 1950's [6]. The 1970's was another period of important research in the area [7-9]. Research in the subject has been going on continuously since the 1950's and with more powerful computing power we are beginning to see more applied uses for GAs.

A GA is a specific type of search technique that models biological systems. *Individuals* within a population are modeled after values in the domain. Each individual has *genes* that represent different traits. A *Fitness Function* is used to determine an individual's *Fitness*, which is how well an individual copes with the environment or domain. A GA begins with an initial population, normally randomly generated. Each generation goes through three biological operations: *Selection*, *Cross-ove*r and *Mutation*. The fitness of each individual in the population is used to determine which individuals will reproduce. Individuals with higher fitness have a greater chance of reproducing. The process of selecting two individuals to reproduce is called selection. Cross-over is the process of taking some genes from each parent and producing new offspring. To encourage exploration within the domain, some genes are changed or mutated based upon a *mutation rate*. Selection, cross-over and mutation produce a new generation. Over numerous generations the population converges to the optimum within the domain.

2.2 Genetic Algorithms as Classifiers

Because GAs are search techniques they are not normally thought of first as being tools that classify. However, GAs can be used for classification. GAs have been used as data classifiers in the diagnosis of cancer [10]. They have been used in financial security forecasting [3-5]. While not the most common use of a GA, they can be used to classify data.

The challenge of using a GA for classification is how to represent the search space. Individuals in GAs represent domain values. But classification rules are more complex than simple domain values. Rules have conditions and conclusions. Conditions can be very complex. Mapping these complex rules into strings allows GAs to seek out optimal rules.

De Jong, Spears and Gordon [11] used a GA to develop an algorithm called GABIL. GABIL represents classification rules as individuals in disjunctive normal form. Each individual within the population is a classification rule. The left-most genes describe a condition and the right-most genes indicate the class that data matching the description should be placed into. In their research they present an example [11]. The example assumes that the rule is attempting to classify an object as a *widget* or *gadget.* There are two characteristics of the objects: size and shape. For size there are values of {*small*, *medium*, *large*} and for shape there is {*sphere*, *cube*, *brick, tube*}. Each value for the characteristics and conclusions are represented by a binary value in the Individual's genes. An Individual containing the following gene sequence, 11110000, would represent the following rule:

	111		1000		0
If	object is *small, medium* or *large*	And	object is *sphere*	Then	object is *widget*

GABIL uses variable-length rule sets meaning that multiple rules could be joined together to form an individual. This is unusual for GAs that normally have fixed length individuals. A set of training data was used to determine the best rules. GABIL is used to classify medical data on patients to determine if they have breast cancer.

A second GA classification method is from Booker, Goldberg and Holland [12]. This is a hybrid algorithm that combines a GA with a bucket brigade algorithm. This algorithm used a GA to discover rules. The bucket brigade algorithm is used to evaluate and assign credit to rules generated by the GA. This algorithm was designed for domains that are very fluid, in which new information is constantly coming in. An example given is to guide a robot that is to locate certain objects in the environment while trying to avoid other objects. In the example the robot moves and the object moves giving an ever changing domain space to optimize.

The Booker, Goldberg, Holland [12] algorithm defined individuals much differently than GABIL. This algorithm assigns meaning to binary values of 0 and 1, but also introduces a new symbol #. This new symbol is used when the value of the corresponding position is not included in the rule. If the genes of an individual represent movement, size and color respectively, then 1#0 would represent the following condition:

1		#		0
Object is moving	AND	object size is irrelevant	AND	object is black

These conditions then get associated with an action that the robot can take, which typically is some type of movement. As stated earlier these results of the GA are passed to another algorithm, bucket brigade. This algorithm determines the effectiveness of each rule.

The two GAs presented in this literature review represent two types of domains to be optimized. GABIL represents optimization problems in which the domain is unchanged. The Booker, Goldberg, Holland algorithm contains an ever changing domain. The problem of classifying stocks exhibit characteristics of both types of optimization.

2.3 Genetic Algorithms as Financial Forecasters

There have been hundreds of systems developed to forecast financial data [13]. Most use some type of artificial intelligence technique from neural networks [20] to GP. Atsalakis and Valavanis [13] state that a goal of the research community is to produce the best results using the least amount of information about the stocks and by developing the least complex model. Many financial forecasting algorithms, including the one presented in this paper, attempt to meet these two conditions.

Mahfoud and Mani [3] developed a GA that classifies stocks by having each individual within a population represent a classification rule. They used a clustering NGA to develop classification rules that were applied to a large number of stocks. The goal of the research was to make a prediction about each stock by classifying them into groups to buy or sell.

The Mahfoud and Mani [3] research provides a novel way to represent stock classification rules for GAs. Each individual represents a classification rule. Each piece of data used in the experiment is given a number of genes in the individual. The first two bits represent the numerical condition: 00 is >; 01 is <; 10 is = and 11 is !=. The remaining genes store the value for the rule that corresponds to the data about the stock. This can be any number of genes based upon the precision needed. Finally a bit used to represent if the stock meets the condition should be bought or sold. Now rules can be created like:

If Price < 15 and EPS > 1 Then Buy [3]

The Mahfound and Mani [3] research uses 15 attributes of the stocks used and covers a 12 week period. The GA searches for optimal rules. The evaluations rules are then applied to each stock. Results show that the algorithm correctly predicts stock movement 47.6% of the time. It makes no prediction 45.8% of the time. And it incorrectly predicts the direction of a stock only 6.6% of the time [3]. The research presented in this paper is very similar to Mahfoud and Mani with a few exceptions. The method defined in this research attempts to select a single stock to purchase each week. It also uses a specialized NGA.

A variety of other research uses GAs for financial forecasting. Tsang, Markose and Er [4] use a GA to attempt to locate temporary misalignment between options and futures. When these conditions happen within a market investors can position themselves to profit. Wagman [5] uses a GA to evaluate entire portfolios. Other methods create hybrid approaches [14] by combining GAs with other methods. GAs are probably not the most popular method to predict stock prices, however results for many methods show promise.

2.4 Niche Genetic Algorithms

NGAs are specialized GAs that attempt to find multiple optima within a domain. NGAs are often used for finding local maximums and minimums of functional optimization problems, while traditional GAs are used for global maximums and minimums. NGAs generally fall into one of two categories: Fitness Sharing and Crowding. NGA Fitness Sharing methods alter the fitness function to prevent global convergence. This is done through adjusting an individual's fitness based upon how close it is to other individuals [15]. NGA Crowding methods replace individuals from one generation with ones from a previous generation [8, 16, 17]. Most of these methods forum groups, called clusters, of individuals that are within a predetermined radius. Typically the strongest member of each cluster is moved into the new generation. This promotes genetic diversity. NGAs are useful in searching domains in which multiple optima exist.

The Dynamic-radius Species-conserving Genetic Algorithm (DSGA) is a recently develop NGA framework that performs very well in a variety of domains [18]. DSGA enhances a traditional crowding NGA, SCGA [16], with a tabu list [19] that is used to encourage exploration. The tabu list stores investigated areas of the domain. Exploratory techniques are used within the algorithm to encourage the population to investigate other areas of the domain. Previous results of DSGA show that it competes very well against other NGAs [18].

3 DSGA and Financial Forecasting

The DSGA is a clustering algorithm framework that uses a tabu list and varies the radius during execution. The tabu list stores optimal areas of the domain that have already been investigated. Future generations are encouraged to explore other areas of the domain. As in most clustering algorithms a radius parameter is used to determine the area of a cluster in the domain. DSGA varies the radius at fixed intervals as the algorithm runs. This helps mitigate poor radius choices.

3.1 Algorithm Overview

This research uses two sets of data. DSGA is run against training data and produces a set of rules. The rules are then applied to the second set of data, the test data. The set of rules are used to select a stock to purchase in the future. By applying rules to data from week x, a single stock is selected to be purchased in week $x + 1$. This is done by summing up the number of rules that indicate to buy the stock and subtracting the number of rules that indicate to sell the stock.

$$\text{Purchase\_Indicator}(\textit{stock}) = \text{\#\_buy\_rules}(\textit{stock}) - \text{\#\_sell\_rules}(\textit{stock}) \quad (1)$$

This is referred to as the *purchase indicator*. This can be seen in the Data Flow Diagram in Figure 1.

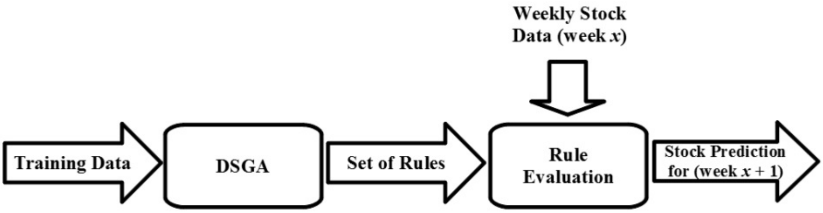

Weekly Stock Data (week *x*)
Training Data
DSGA
Set of Rules
Rule Evaluation
Stock Prediction for (week *x* + 1)

Fig. 1. The DSGA Financial Forecasting algorithm accepts training data and uses DSGA to locate a set of rules based upon the training data. The rules are used in a Rule Evaluation component that takes weekly stock data and predicts a stock to purchase the following week.

The algorithm maintains a list of rules, which is the tabu list, as the NGA runs. Periodically the algorithm analyzes the current generation and looks for convergence. Areas in the domain that the generation converges to are included on the list of rules. During this phase of the algorithm the seeds are also placed on the tabu list. This is discussed in a future section. The algorithm goes through multiple phases with each phase having the opportunity to add rules to the tabu list through convergence or seeds.

3.2 Individuals and Genes

The genes used to create individuals form a rule that can be applied to certain stocks. The first bit represents the decision to buy the stock or sell the stock. Arbitrarily, 1 represents a decision to buy and 0 represents a decision to sell. The remaining genes are divided into sections for each characteristic in the rule. This research uses four stock characteristics which are described in Table 1.

Table 1. Characteristic of stocks

Characteristic	Description
Percent stock changed	The percent change in the stock price for week *x*, which is the week prior to the week that the algorithm attempts to predict the stock for.
Percent volume changed	The percent change in volume during week *x* compared to week *x - 1*. Volume is the number of shares of a stock sold.
Days to next dividend	The number of days until the next dividend. If the stock does not have a dividend, this value is the maximum integer.
Percent return of next dividend	The percent return based upon the stock price of week *x* of the amount of the dividend. If the company does not give out dividends, this value is 0.

Each section begins with a bit to indicate if the characteristic is used in the rule. This allows rules to be made up of different combinations of characteristics. The next two bits indicate the condition for the characteristics (<=, <, > or >=). The remaining bits in the section determine the value for the rule. These are binary numbers and have implied decimal places where appropriate. In some cases there is a sign bit. When there is a sign bit it is always the first digit. All numbers are stored in little-endian format, with least significant digits to the left. Table 2 shows the actual 43 genes.

The following is an example of the gene sequence 1111001001001001010110111111011010101010100. This individual corresponds to the following rule:

1	1	11	0100100	1	00	10101101	1	11	1101101	0	10	1010100
Buy	Use in Rule	>=	-1.8	Use in Rule	<	90	Use in Rule	>=	91	Do Not Use in Rule	>	0.21

This rule states the following: Buy stock if all of the following conditions are met: percent change of stock price during week *x* >= -1.8 and percent change volume of throughout week *x* < 90 and days to the next dividend >= 91

Table 2. Positions of genes within individual

Position(s)	Description
1	1 = buy, 0 = sell
2	1 = include percent change price in the rule, 0 = don't include it
3-4	00 <; 01 <=; 10 >; 11 >= for percent change price
5-12	number for percent change price rule with one implied decimal place. First bit is sign bit 0 -; 1 +
13	1 = include percent volume change in the rule, 0 = don't include it
14-15	00 <; 01 <=; 10 >; 11 >= for percent volume change
16-23	number for percent change volume rule. First bit is sign bit 0 -; 1 +
24	1 = include days to next dividend in the rule, 0 = don't include it
25-26	00 <; 01 <=; 10 >; 11 => for days to next dividend
27-33	number for days to next dividend (max 127 days)
34	1 = include percent return on dividend in the rule, 0 = don't include it
35-36	00 <; 01 <=; 10 >; 11 => percent return on dividend
37-43	number for percent return with 2 implied decimal places.

3.3 Distance

There are a number of ways to determine distance in a GA. There is genetic difference, where the number of genes that have different values. There is also Euclidean distance. When rules for this algorithm are applied to data sets, they return order pairs consisting of a stock and week. A rule may return zero or more (stock, week) pairs. This research uses the intersection of the (stock, week) pairs between two individuals. If the rules for two individuals retrieve many of the same (stock, week) pairs, the distance between the individuals is very small. This research uses the intersection of the two sets of (stock, week) pairs to determine distance. The distance function is defined as:

$$\text{distance}(i_1, i_2) = 1 \,/\, |(\text{sw}(i_1) \cap \text{sw}(i_2))| \tag{2}$$

The function sw returns the (stock, week) pairs for the conditions encoded in the individual. A distance function is needed for all GAs that use fitness sharing.

3.4 Fitness Function

In order to choose individuals for the selection process, each individual's fitness must be computed. The fitness function should correspond to the goal of the GA. In this case the goal is the highest percent return in the following week, week $x + 1$. The fitness function retrieves all data for (stock, week) combinations that match the rule. The fitness is the average of the stock price percent returns for the following week. This fitness function encourages the GA to locate stocks that should produce favorable returns in the following week.

The first bit of the individual indicates if the rule suggests buying or selling the stock. If the individual corresponds to a rule to sell, then high fitness would have a negative return for the following week. So, for the rules that indicate selling the stock, the fitness function returns -1 times the average of the price percent return for the following week.

There were some minor manipulations for the fitness function. It is possible that by the fitness function defined above, there could be a negative fitness. This can cause issues with selection, so all fitness values were increased by 10. Also, having rules that only correspond to a small amount of data in the training data set is not very useful. So a parameter, w, was introduced. If a rule does not retrieve at least w (stock, week) pairs in the training data set, the fitness is 0. This encourages the GA to locate rules that apply to larger sets of data and not focus in on global optima. These modifications to the standard fitness function allow the algorithm to locate better rules.

Unlike other GAs DSGA does not use the fitness function for selection. It uses a shared fitness function. The fitness function value is altered to encourage exploration in new areas of the domain. The tabu list stored potential optimal areas of the domain that have already been explored. The shared fitness function will decrease the fitness of individuals close to these areas. The shared fitness for an individual i is, sf(i) = fitness(i) / m_i. The value for m_i is calculated as follows. The variable TLj is the jth individual on the tabu list.

$$m_i = \sum_{j=1}^{TabuSize}\left(1-\left(\frac{distance(i,TLj}{0.1}\right)\right) \tag{3}$$

Shared fitness encourages the algorithm to explore areas of the domain not on the tabu list.

3.5 Parameters

There are a variety of parameters used in DSGA. Some are found in all GAs like population size and mutation rate. Some are found in other NGAs like radius. Some are unique to DSGA. Table 3 shows the parameters used in DSGA along with a description of their purpose.

Table 3. DSGA parameters

Parameter	Variable	Description
Population Size	N	The number of individuals in each generation.
Number of Generations	NG	The number of generations before the GA terminates.
Mutation Rate	M	The probability that a gene will be mutated.
Seed Radius	IS	The size of the radius at initialization.
Radius Delta	SD	The size of the change of the radius.
Reevaluation Loop Count	RLC	The number of generations in the intervals for deciding candidates for the tabu list.
Convergence Limit	CL	The number of identical individuals to be placed on the tabu list.
Weight	W	The number of (stock, week) pairs that must be returned by a rule in order to have a fitness other than 0.

3.6 **DSGA Algorithm**

DSGA is a typical clustering GA. Within the basic selection, crossover and mutation steps of a GA, DSGA incorporates additional steps of seed selection and seed conservation. A *seed* is the strongest individual within a population for some area of the domain. Seeds are selected by sorting the population by fitness. In the case of DSGA the sorting is done by the shared fitness as described in the section above. Individuals are evaluated for seed selection from the fittest individual to the least fit. If no other individuals within a predefined radius are seeds, then the individual is a seed. When a new generation is created, seeds from the previous generation replace individuals in the next generation. Each seed replaces the weakest individual within the predefined radius. If no individuals are within the radius, the seed replaces the globally weakest individual. This ensures that seeds are carried into the next generation. In each generation seeds are evaluated, so a seed in one generation may not be a seed in the next generation. Many clustering GAs work in this way [16].

After *RLC* number of generations, DSGA evaluates the current generation. DSGA puts all current seeds on the tabu list. It also analyzes the current generation for convergence. If there are *CL* or more identical individuals within the generation, one of them is placed on the tabu list also. Then all individuals placed on the tabu list are replaced within the population with randomly generated individuals. The radius is then changed by the Sigma Delta, *SD*. In this research the radius is always increased by *SD*. Changing the radius helps the algorithm locate other optima. A known limitation to clustering algorithms is poor selection of the radius parameter [21]. Varying the radius helps mitigate this limitation. The algorithm then generates another *RCL* number of generations before it performs the evaluation again. Table 4 shows pseudocode for the DSGA algorithm.

DSGA has shown promise in locating local optima [18]. Tests against other NGAs show that it is very competitive. DSGA is especially good at locating arbitrarily close optima.

Table 4. DSGA pseudocode

```
Pseudocode
Initialization
While not termination condition
    For (int $r$ = 1; $r$ < $RLC$; $r$++)
            Seed Selection
             Selection
            Crossover
            Mutation
            Seed Conservation
    End for loop
    If there exists an individual $d$ with $CL$ or more identical individuals then
        Add $d$ to tabu_list
        Replace all individuals identical to $d$ with randomly generated individual
        End if
        Add the seeds of the current generation to the tabu_list
        Alter radius by $SD$
End while loop
Output the tabu_list – these are the generated classification rules
```

4 Results

Data for this research was obtained from the Dow Jones Index stocks, which are 30 of some of the largest American companies. The training data came from the first quarter of 2011 and the test data came from the second quarter of 2011. In the testing data, one stock was purchased each week based upon data from the previous week and the rules generated from DSGA. It was assumed that a constant amount of money was used to purchase each stock. Calculations in this section are based upon the stock being purchased at the opening price at the beginning of the week, which is usually Monday. The calculations are based upon the stock being sold at the closing price on the last day of the week, usually Friday.

As mentioned in the previous section, it is possible in a week for two stocks to have the same purchase indicator. In situations like this a stock characteristic is used to break the tie and resolve the conflict. The characteristic of percent change in price was used in this research. If multiple stocks have the same purchase indicator for determining to purchase the stock in week $x + 1$, whichever stock had the greatest price percent gain in week x would be selected. Because two stocks rarely have the exact same price percent increase for a given week, this is a good characteristic to use

4.1 Parameter Settings

Table 5 shows the parameter settings for this research. Their definitions can be found in Table 3. The parameter values were determined through experimentation prior to running the eight trials.

Table 5. Parameter settings

Parameter	Variable	Value
Population Size	N	25
Number of Generations	NG	80
Mutation Rate	M	0.01
Seed Radius	IS	0.05
Radius Delta	SD	0.01
Reevaluation Loop Count	RLC	20
Convergence Limit	CL	2
Weight	W	10

4.2 Selected Stocks

Tables 6 shows the stocks selected for one of the trials. A stock was selected for the 13 weeks in the experiment. The second column for each trial shows the percent return of the stock in the following week. A total return and weekly return for each trial is shown in Figure 2.

Table 6. Stocks select for trial 4

Week	Stock	Return	Week	Stock	Return
1	T	-0.10%	8	PFE	2.15%
2	MRK	2.46%	9	PFE	-0.76%
3	MRK	-0.47%	10	DIS	-1.74%
4	DIS	1.79%	11	VZ	0.79%
5	MRK	-0.14%	12	T	-0.72%
6	T	0.67%	13	VZ	5.00%
7	DIS	0.58%	Total		9.51%

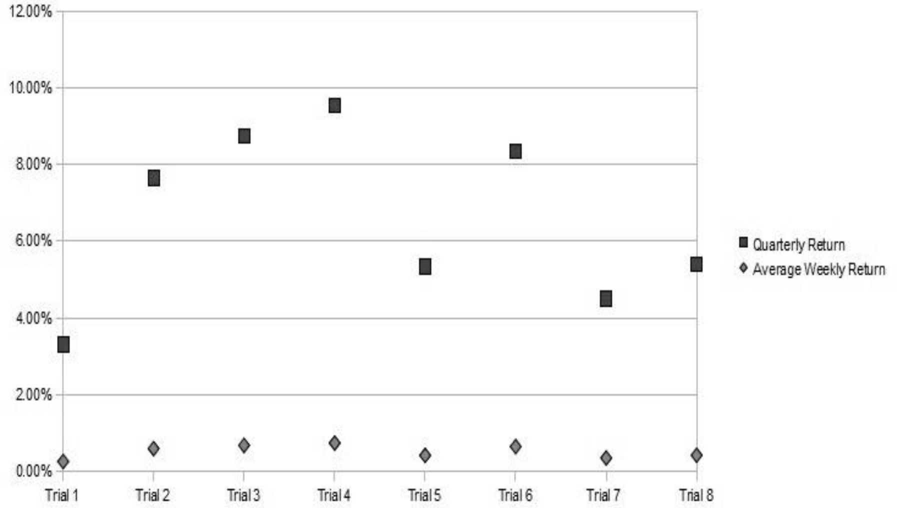


Fig. 2. Percent return per trial

4.3 Rates of Return

The results for this research show the rate of return for the quarter and average weekly rate of return. Since a constant amount of money is invested each week, the rate of return for the quarter is the sum of each weekly return. The following graph shows the percent return for each of the 8 trials.

The results for DSGA are compared to other indicators during this period. The data shown in the table below for DSGA is the average of 8 runs. *Maximum* is the true optima. This is the rate of return assuming the best performing stock is selected each week. *Minimum* is the rate of return for selecting the worst performing stock each week. The *Dow Jones Index* shows the rate of return by investing an equal amount of money in each of the 30 stocks. Finally, the *Average of Stocks* shows the rate of return for averaging all stocks in the index. The Dow Index and Average of Stocks are realistic returns that many investors obtain. Table 7 shows the results for DSGA and the indicators.

Table 7. DSGA results compared against other indicators

Methods/Indicators	Quarter	Week
DSGA	7.075%	0.54%
Maximum	55.90%	4.30%
Minimum	-57.02%	-4.39%
Dow Jones Index	1.66%	0.13%
Average of Stocks	2.50%	0.19%

Of the 8 runs of the DSGA algorithm the average rate of return for the quarter was 7.075% or 28.3% a year. This corresponds to a 0.54% return per week. The standard deviation was 2.166.

4.4 Discussion of Results

While the return of DSGA was not optimal, 55.9%, it did do very well against the Dow Jones Index. It outperformed the Dow Jones Index by more than a factor of three. Even one standard deviation away from the average still beat the Dow Jones Index by almost 4%. All of the trials outperformed the Dow Jones Index. The minimum return was 3.3% which outperformed the Dow Jones Index 1.64%. The maximum return of the trials was 9.5% which outperformed the Dow Jones Index by 7.84%. These results show that the DSGA algorithm did very well against the Dow Jones Index.

To see if the results could be a coincidence the T-test was performed. The T-test was performed with the following values, $\mu = 1.66$, $n = 8$, average = 7.075, $s = 2.166$ and $\alpha = 0.001$. In the T-test μ is the expected value if the algorithm had no ability to select stocks that would increase in value. This comes from the performance of the Dow Jones Index. The value n is the number of trials. 7.075 is the average of the results of the experiment. The value s is the standard deviation of the results. Finally, α is the confidence level being tested. When calculated the t value comes out to be 7.1 which is well within the 0.001 confidence value. The T-test shows that within a 0.001 confidence level the results of this research were not a coincidence. The T-test is a good statistical analysis test for experiments with a low number of trials, normally considered $n < 30$.

5 Conclusion and Future Work

The DSGA algorithm did very well in predicting single stock selection for a week of the 30 Dow Jones Index stocks. It produced returns many times greater than the Dow Jones Index, which is often considered a safe and lucrative investment selection. The Dow Jones Index stocks make a great set of stock for forecasting systems because if the system predicts a stock incorrectly losses are normally minimal. DSGA produces these results by examining only four stock characteristics: change in stock price, change in stock volume, days until the next dividend and return of next dividend.

The results of this research suggest many other areas of future work. Future research could be to use the DSGA algorithm on other stock data sets, like the S&P 500, NASDAQ 1,000 and FTSE Eurotop 100. This researched used a three month timeframe for training data and test data. This was an arbitrary timeframe. Research in other timeframes could be another area of future research. Although the parameters for the algorithm produced very good results there may be better parameter values. The DSGA algorithm could also be used for other data classification problems. Future research in these areas could produce better returns and expand our understanding of stock forecasting.

Portfolio management is another possible area for future research. Instead of using the algorithm to predict a stock to purchase, the algorithm could be used to evaluate portfolios of already purchased stock. That algorithm could then be used to recommend changes to the portfolio.

Stock forecasting is a very complex problem because it is based upon many factors. Some of these factors are human bias which cannot be represented by mathematical models. NGAs seem capable of locating rules that produce very attractive returns. The DSGA algorithm presented in this paper produced returns many times greater than the stock index that it was based upon. Results indicate that it is suitable for stock forecasting.

References

1. Graham, B., Dodd, D.: Security Analysis. McGraw-Hill, New York (1934)
2. Wang, J.J., Wang, J.Z., Zhang, Z.G., Guo, S.P.: Stock Index Forecasting Based on a Hybrid Model. Omega 40(6), 758–766 (2012)
3. Mahfoud, S., Mani, G.: Financial Forecasting Using Genetic Algorithms. Applied Artificial Intelligence 10, 543–565 (1996)
4. Tsang, E., Markose, S., Er, H.: Chance Discovery in Stock Index Option and Future Arbitrage. New Mathematics and Natural Computation 1(3), 435–477 (2005)
5. Wagman, L.: Stock Portfolio Evaluation: An Application of Genetic-Programming-Based Technical Analysis (2003)
6. Bremermann, H.J.: The Evolution of Intelligence. The Nervous System as a Model of its Environment (Technical Report, No.1, Contract No. 477, Issue 17). Seattle WA: Department of Mathematics, University of Washington (1958)
7. Cavicchio, D.J.: Adaptive Search Using Simulated Evolution. Unpublished doctoral dissertation. University of Michigan, Ann Arbor (1970)

8. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems (Doctoral dissertation, University of Michigan). Dissertation Abstracts International, 36(10), 5140B (University Microfilms No. 76-9381) (1975)
9. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
10. Dolled-Filhert, M., Ryden, L., Cregger, M., Jirstrom, K., Harigopal, M., Camp, R.L., Rimm, D.L.: Classification of breast cancer using genetic algorithms and tissue microarrays. Clinical Cancer Research 12, 6459–6468 (2006)
11. De Jong, K.A., Spears, W.M., Gordon, D.F.: Using Genetic Algorithms for Concept Learning. Machine Learning 13(2-3), 161–188 (1993)
12. Booker, L.B., Goldberg, D.E., Holland, J.H.: Classifier Systems and Genetic Algorithms. Artificial Intelligence 40, 235–282 (1989)
13. Atsalakis, G.S., Valavanis, K.P.: Survey Stock Market Forecasting Techniques - Part II: Soft Computing Methods. Expert Systems with Applications 36, 5932–5941 (2009)
14. Armano, G., Marchesi, M., Murru, A.: A Hybrid Genetic-Neural Architecture for Stock Indexes Forecasting. Information Sciences 170(1), 3–33 (2005)
15. Goldberg, D.E., Richardson, J.: Genetic Algorithms with Sharing for Multimodal Functional Optimization. In: Proceedings of the Second International Conference on Genetic Algorithms and their Application, Cambridge Massachusetts, pp. 41–49 (1987)
16. Li, J.P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A Species Conserving Genetic Algorithm for Multimodal Function Optimization. Evolutionary Computation 10(3), 207–234 (2002)
17. Ling, Q., Wa, G., Yang, Z., Wang, Q.: Crowding Clustering Genetic Algorithm for Multimodal Function Optimization. Applied Soft Computing 8, 88–95 (2008)
18. Brown, M.S.: A Species-Conserving Genetic Algorithm for Multimodal Optimization (Doctoral dissertation). Available from Dissertations and Theses database, UMI No. 3433233 (2010)
19. Glover, F.: Tabu Search – Part I. ORSA Journal on Computing 1(3), 190–206 (1989)
20. Cao, Q., Parry, M.E.: Neural Network Earnings Per Share Forecasting Models: A Comparison of Backward Propagation and the Genetic Algorithm. Decisions Support Systems 47(1), 32–41 (2009)
21. Ando, S., Kobayashi, S.: Fitness-based Neighbor Selection for Multimodal Function Optimization. In: Proceeding of the 2005 Conference on Genetic and Evolutionary Computation, Washington DC, pp. 1573–1574 (2005)

Multi Model Transfer Learning with RULES Family

Hebah ElGibreen[1] and Mehmet Sabih Aksoy[2]

[1] IT Department,
[2] IS Department,
College of Computer and Information Sciences, King Saud University
{hjibreen,msaksoy}@ksu.edu.sa

Abstract. Due to the rapid growth of computer technologies and the extensive changes in human needs, expertise and digital information were used to induce general conclusions. Such conclusions can be used to deal with future activates and make the life of humans easier. One active filed of machine learning that was developed for this purpose is inductive learning, and several families have emerged from this field. Specifically, RULES family was discovered as covering algorithm that directly induces good and general conclusions in the shape of simple rules. However, it was found that RULES suffer from two major deficiencies. It needs to tradeoff between time and accuracy when inducing the best rule and it did not appropriately handle incomplete data. As a result, this paper will present a new RULES algorithm, which takes advantage of previous versions of RULES family in addition to other advance methods of machine learning, specifically Transfer learning. Moreover, multi-modeling is also merged to transfer the knowledge of a different classification model and further improve the original algorithm. At the end, an empirical test is applied to compare the proposed algorithm with different single-model algorithms to prove that using the past knowledge of other agents in different domains improves specialization accuracy, whether the data is complete or incomplete.

Keywords: Rules induction, transfer learning, RULES family, covering algorithms, multi-model classification.

1 Introduction

In order to predict future activities and induce general conclusions a field of machine learning, called inductive learning, was introduced. It is a form of data analysis that uses the knowledge gained through training to conclude general conclusion in the shape of rules and identify new objects using a classifier. Different methods have been proposed to induce classification rules. These methods were divided into two main types: Decision Tree and Covering algorithms. Decision Tree algorithms, such as ID3 [1] and C4.5 [2], discover rules using a tree that is used to represent the rules [3]. This type of algorithm drew a lot of attention in the past few years because of the simplicity of deriving a rule from a tree. However, handling of decision trees created other problems that badly affected the induction process.

In covering algorithms, however, the properties of inducing rules directly from the dataset made it more preferable than the decision tree structure, as described in [4].

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 42–56, 2013.

First, representing the rules as "IF ... THEN" statement makes it more readable than trees. It has been empirically proven that rule learning sometimes better than decision trees, which make it impossible to say that decision tree methods are absolutely better than covering algorithms. Additionally, the resulting rules can be easily translated into any expert system language and can be directly stored in any knowledge-based systems. Finally, it is easy to analyze and modify the induced rules due to its independency. Hence, any rule can be understood and validated without the need to reference other rules in the repository.

As a result, researchers have recently tried to improve covering algorithms to be comparative or even better than the decision tree methods. Thus, different families have been born for this purpose. Specifically, RULe Extraction System (RULES) [5] has been developed as a simple covering algorithm family. It allows for the discovery of inconsistent rules to automatically obtain partial immunity to noise. It is differentiated from the other algorithms because of its preservation of the covered examples, to avoid repetitive computation while generalizing the results. Hence, it will resist fragmentation and small combination problems.

Different versions of RULES family have been proposed by different authors. Nevertheless, even though each version of RULES family was proposed for a certain purpose but most of them have one common property. Specifically, when searching for the best rule, a rule is induced starting from an empty one then a specialization process begins to create more specialized rules from the seed examples. This process was proposed to improve the accuracy of rule induction, but it consumes a lot of time. It needs special pruning procedures to stop the search and reduce its space. It is difficult to decide what to specialize and when to stop. In addition, as stated in [6], searching by specialization is usually considered as one of the most time-consuming methods. Hence, it causes the needs to tradeoff between the accuracy and speed of the algorithm.

Moreover, another problem that was sought in RULES family is the way it handles incomplete data. Examples that have a missing class are either neglected or filled based on other examples in the training set. When the example is neglected it is possible to lose important information and decrease the performance of the algorithm. On the other hand, when the current examples are only considered then the available classes are only reflected while, in reality, other labels might also be missing from the all the training set. Hence, not only the current data is important but also future cases should be considered to increase the resistance to noise.

Moreover, in general, Kotsiantis [7] clarified that the question that should be asked when evaluating an inductive learning algorithm is not whether the algorithm is the best, but instead "*under which conditions a particular method can significantly outperform others on a given application problem.*" However, testing and choosing the best classifier is a difficult task, especially for novice developers. Moreover, inductive learning algorithms should be general enough to cover as many problems as possible. One active research area that was proposed to solve this problem is learning by multiple models, like ensemble learning [8]. However, as indicated in [9], most of the existing methods of multi-model learning are confusing; the more accuracy it results the less comprehensible it becomes.

Consequently, the purpose of this paper is to propose a new RULES version that improves the searching procedure and incorporates a new scheme to generalize the algorithm and efficiently handle incomplete data. Specifically, the proposed algorithm is an extended version of RULES-TL [10]; but different inductive models are used instead of only one. In addition, Transfer Learning [11] is also applied to transfer rules discovered, by other agents, based on different but related tasks. The transferred rules are used as the base knowledge to accurately fill incomplete data with all possible labels, whether it occurs in the training set or in the knowledge base, and to reduce the time of specialization and guide the searching process by using past knowledge instead of pure pruning. On the other hand, multi-modeling is applied in a simple way to improve the accuracy without affecting the comprehensibility.

This paper is organized as follows. First, the background needed to understand the proposed algorithm is presented. Then, the related work is discussed. After that, the proposed method is presented, and its details are explained. Following that, the algorithm is tested and its empirical result that compares it with other covering algorithms is explained and discussed. Finally, the paper is concluded and future work is presented.

2 Background

This section explains the background needed to understand the proposed algorithm, including RULES family and Transfer Learning.

2.1 RULe Extraction System - RULES

RULES family is one of the covering algorithms that directly induces rules from the training set based on the concept of separate and conquer. It is considered as one of the simplest and precise families that induce one rule at a time based on a seed example and then apply specialization process to find the best rule. The rule that covers the most positive and least negative examples are chosen as the best rule of the current seed example. Hence, RULES family does not require finding of consistent rules. It allows the best rule to cover some negative examples in order to handle noisy data, reduce over-fitting problem, and increase the flexibility of rule induction. After that, examples that are positively covered by the discovered rules are marked as covered without removal. This way, repeating the discovery of the same rule is prevented while coverage accuracy and generality of new rules will be preserved. At the end, the algorithm is repeated until all examples are covered.

Nevertheless, the most important part of RULES algorithm is rule forming process [12]; which is the searching strategy that aims to create the best rule to cover a certain example. In specific, it searches for the best conjunction of conditions and measures its strength using a certain heuristics. To reduce the cost of this process, RULES family order the search based on the concept of specialization (general-to-specific) process. The main idea of specialization is to start from the most general rule, i.e. null rule, and then specialize it by adding one attribute at a time as an additional condition.

However, this process still consumes a lot of time and greatly affects the algorithm performance. It needs to tradeoff between the speed of the search and its accuracy.

Consequently, it can be concluded, from all above, that RULES family has a good future concerning the rule discovery. However, it needs further improvement. One way to improve it is by applying advance machine learning methods, such as Transfer learning, as will be explained next.

2.2 Transfer Learning

Transfer Learning (TL) is one area of machine learning that makes the agent learn by connecting different but related environments [13]. Thus, instead of starting from scratch, the agent will be able to direct its learning rather than randomly exploring the problem [14]. For example, a person can learn how to read a French text using his own knowledge of English characters. Even though French and English are different languages, but they have related context that a person can re-use to minimize the time needed for the learning process. Hence, it imitates how human could apply previously learned knowledge to improve the lifelong learning methods of machine learning [15].

TL approaches, in general, have been divided into different types depending on the state of the class labels, as explained by Sinno Jialin and Qiang [11]. Moreover, different types of transfer can occur depending on what to transfer. It can transfer instances, feature representation, parameters, or even relationship knowledge between the source and target tasks depending on the problem at hand [16].

3 Related Work

In RULES family, different problems have been targeted, and different methods have been proposed along with it. The early versions of RULES family, specifically RULES-1 [17], 2 [18], and 3 [19], were basically proposed to enhance the performance of covering algorithms and induce better rules using a simple method. However, RULES-3+ [20] and RULES-EXT [21] was proposed, afterwards, to improve the performance of RULES-3. Nevertheless, RUELS-3+ became very famous and was used as the base for many new algorithms. It is different from the previous version in the way it searches for the best rule and in sorting and selecting the candidate rules, where specialization and H measure were applied.

After that, RULES-4 [22], which is based on RULES3+, and RULES-IS [23] were developed for incremental learning. In addition, RULES-5 [12], RULES-F [24], and RULES-5F [25] was also developed based on RULES3+ to handle continuous values during the learning process. After that, scalability became a concern so RULES-6 [26], 7 [27], and SRI [28] were developed to enhance RULES family performance and increase its speed.

Nevertheless, even though these methods served different purposes and were designed by different authors, but all these versions worked based on the concept of specialization, which caused the need to tradeoff between accuracy and time. Nevertheless, RULES-IS did not have this property since it was built based on the immune system network. However, this version has lost the simplicity property that makes

RULES family more appealing than the others. Moreover, in all visions of RULES algorithm, incomplete data were inappropriately handled. Examples that contain missing classes were either neglected, removed, or current examples were used to fill it without considering future cases or the volatility of data.

Hence, it can be concluded that RULES family is still lacking in different areas. In general, two main deficiencies can be sought when searching for the best rule and dealing with incomplete data. Specialization is important to surpass the problem of irrelevant conditions. However, it consumes a lot of time and highly affects the algorithm performance. Additionally, incomplete data was inappropriately handled without considering future cases. Therefore, past knowledge of other agents can be used to solve these problems.

Moreover, it was found that past knowledge can be learned from other tasks to be used as the base knowledge of the current problem using TL. Hence, TL can be applied with rule induction. In general, inductive TL has been used in different machine learning techniques. Instances were transferred to improve the classification accuracy, as in [29]. Features' representation were transferred to create a model to handle unreliable data, especially in images and text mining, as in [30]. Parameters were also transferred to guide the classification process and, finally, relationship knowledge has been transferred to simplify dealing with a complex target problem, as in [31-33].

Nevertheless, these techniques were mostly targeting certain type of data. There target was either to improve the accuracy or replace the heuristics, and it was usually used to deal with a scarce target task that is complex to deal with. Thus, most of these algorithms actually increased the space of searching or reduced its accuracy, which can even cause greater problems with scalability. However, it was concluded in [34] that transferring the whole rule from the source to the target task would tremendously improve the induction process performance. Hence, it would be better to take advantage of the good properties that can be gained from transferring rules while avoiding its drawback. Moreover, transferring rules from different models can also be applied to improve the algorithm performance without affecting its complexity.

In our knowledge, no one yet has used Rule Transfer in covering algorithms and simple multi-model mechanism has not been applied in RULES family. Consequently, the goal of this paper is to use TL and transfer the rules gathered by different agents in different domains to solve the problems of RULES family, as will be explained next.

4 RULES with Multi-Model Transfer Learning

In general, the proposed algorithm is an extended vision of RULES-TL with rule induction by multiple models. It is an algorithm that is originally built based on RULES-6, and developed to improve the performance of RULES family and scale it further over incomplete data. It integrates relational knowledge transfer (in the shape of rules) with RULES-6. In particular, Inductive TL is applied, which require that the rules transferred from the source contains the class label, and no missing information is transferred. These transferred rules are then used in a way to reduce the searching space before going through the induction process.

The proposed algorithm discovers the rules from the source task using any inductive model. Then, it maps these rules and transfers it to the target tasks. These rules are then used to fill missing classes and reduce the search space by marking the examples covered by these rules so that no further searching is needed in the induction rule procedure, as explained in Fig. 1.

```
Main()
  Input: Tset = Target set, mN = minimum negative, mP = minimum
  positive, w = beam width
  Output: RuleSet = induced rules

Trules = induce source rules using any inductive model
MappedRules = MapRules (Trules, Tset)
//Mark covered examples, store unseen rules, and return induced rues
RuleSet = Mark_covered_examples (Tset, mP, mN, w, MappedRules)
//update Tset and fill its missing classes
Fill_miss_class (Tset, MappedRules)
//Call RULES-6 as in [25]
RulesSet = RulesSet + RULES-6(Tset, mP, mN, w )
---------------------------------------------------------------
MapRules (Trules, Tset)
  Input: Trules, Tset
  Output: Transferred rules

MappedRules = ϕ
For every rule(i) ∊ Trules
MappedR(i) = change the format of rules(i) to the same format
as Tset
MappedRules = MappedRules + MappedR(i)

Return MappedRules
---------------------------------------------------------------
Mark_covered_examples (MappedRules, Tset, mP, mN, w)
  Input: MappedRules, Tset, mP, mN, w
  Output: Transferred rules

RuleSet = ∅
Urules = ∅
For every rule(i) ∊ MappedRules
For every uncovered example ∊ Tset and match rule(i)
Select a seed example (s) from uncovered example
//Call RULES-6 Induce procedure BUT initial BestRule = rule(i)
```

Fig. 1. Pseudo code of the new algorithm

```
NOT empty rule
R = Induce_One_Rule (s, rule(i), Tset, mP, mN, w)
If R≠ ϕ THEN
//mark the rule as seen since it covers an example
 rule(i).Unseen = False
Mark example (s) as covered
RuleSet = RuleSet + R
If rule(i).Unseen = True THEN
Urules = Urules + rule(i)
Store Urules for future use
Return RuleSet
---------------------------------------------------------------
Fill_miss_class (Tset, MappedRules)
  Input: MappedRules, Tset
  Output: Only update the target set by filling its missing
  classes

BestScore = -1
BestLabel = ϕ
For every example (s) with missing class
//get the best label based on Mapped rules
For every rule(i) ∊ MappedRules
Score = Compute S-Score for rule(i)
If (score > Threshold) & (score > BestScore) THEN
BestScore = score
BestLabel = rule(i).label
//If no rule cover the example with missing class
If BestLabel = ϕ THEN
BestLabel = Label of most similar example in Tset
Update s with Label value
```

Fig. 1. *(Continued)*

In particular, it starts by applying any inductive method, such as PRISM or RULES-SRI, over the source tasks to induce the best rules. Then, the resulting rules are mapped to the target task representation. The mapping process first removes any unrelated rules, where an attribute or value in the rule is not found in the target header. Then, its format is transformed to the format of the target task. After that, every mapped rule is taken as the ground base to cover existing examples. If the rule covers an example, then the algorithm induces the best rule based on RULES-6 starting from the mapped rule instead of empty one. Hence, mapped rule is specialized to induce better ones based on the seed example. This will reduce the specialization time and increase the accuracy because the specialization process did not start from scratch and previous knowledge of different models is used.

Following that, covered examples are marked as covered, rules that cover at least one example are marked as seen rule, and rules that do not cover any example are marked as unseen rule for future use. Thus, past knowledge of other agents in different task is used as the base to discover and induce good rules instead of always starting from an empty rule which decreases the need of time/accuracy tradeoff.

After that, incomplete examples, which miss a class, are filled using the mapped rules. It starts to measure the strength of the mapped rules that matches the incomplete example using S measure [35], computed using (1); where P and N is the total number of positive and negative examples, respectively, and p and n is the number of positive and negative examples covered by the new rule.

$$S = \frac{p}{p+n} . \frac{p}{P\_unclassified} . (1 - \frac{n}{N}) \tag{1}$$

Afterwards, the score is compared to a threshold, computed using (2), so that if the rule exceeds this threshold then the example is considered as covered. This threshold is called T threshold [22]. It considers the noise level and the ratio of positive examples in the whole training set. In this threshold, NL is the allowed noise specified by the user, Ei is the number of example that belongs to a class i, and E is the total number of examples in the training set.

$$T = 2 - 2\sqrt{(1 - NL)\frac{E_i}{E}} - 2\sqrt{NL(1 - \frac{E_i}{E})} \tag{2}$$

However, if the algorithm was not able to find any matching rule, then the most similar example in the training set will be used to fill the missing class. The similarity measure is computed using equation (3), where V^{i}_{E1} and V^{i}_{E2} are the values of the continuous attribute (i) in example E1,E2 respectively, V^{i}_{max} and V^{i}_{min} are the maximum and minimum values of attribute (i), and the distance between discrete attributes is computed using (4). D-distance, as explained by Pham, Bigot, and Dimov [12], is a distance measure that can compare any type of examples together. In this measure, distance between continuous attributes is considered in addition to the difference between the discrete ones.

$$D(E1, E2) = \sqrt{\sum_{All\ cont.Attr}(\frac{V^{i}_{E1} - V^{i}_{E2}}{V^{i}_{max} - V^{i}_{min}})^2 + \sum_{All\ disc.Attr} d(A1, A2)} \tag{3}$$

$$d(A1, A2) = \begin{cases} 0 & if\ A1 = A2 \\ 1 & if\ A1 \neq A2 \end{cases} \tag{4}$$

After that, RULES-6 algorithm is applied over the uncovered examples to make sure that all the examples are marked as covered. Consequently, it can be concluded that the accuracy can be improved without trading the induction processing time. Actually, it is anticipated that it would further reduce the time since part of the training set is covered in advance using the transferred rules. Hence, transferred rules can reduce the time of specialization and improve the accuracy due to the use of past knowledge. Moreover, incomplete data are accurately handled using past knowledge instead of neglect important information just because it is incomplete. Thus, it is anticipated that the proposed algorithm will be more scalable and accurate than the original RULES-6, as will be proven in the next section.

5 Experiment

In order to test the performance of the proposed algorithm, different experiments were conducted. The proposed algorithm was first implemented using Java language in JBuilder environment. The experiments were conducted on a PC with Intel®Core™ i7 CPU, 2.67 GHz processes, and 6GB RAM. In addition, to build the experiments and decide on its property, KEEL (Knowledge Extraction based on Evolutionary Learning) tool [36, 37] was used. It can compare the performance of the proposed algorithm and existing methods. Moreover, it has Statistical Analysis Tools (SAT) that can be used to design an experiment and perform a complete analysis on the algorithms' performance using a simple graphical user interface. Hence, three key elements were defined in order to build the desired experiment, as follows.

5.1 Dataset

In order to show how reliable the proposed algorithm is, ten dataset were used to test the proposed algorithm. These data set were taken from KEEL dataset repository [37] and the property of each dataset is shown in Table 1.

Table 1. Experiment dataset property

Dataset	#Examples	#Attributes	#Labels
ecoli	336	7	8
yeast	1484	8	10
Australian Crd.	690	14	2
Crd. Approval	690	15	2
Cleveland	303	13	5
Statlog	270	13	2
Bupa	345	6	2
Hepatitis	155	19	2
Red Wine	1599	11	11
White Wine	4898	11	11

Each dataset is partitioned to five-pair partition using the hold-out approach [38]. Each pair includes a test and training data. Training data was used to train the algorithm, and the test data was used to test its result. In specific, for large dataset (>1000 example), the test set include one-third of the data while the training set include the remaining two third. However, with small data (≤1000 example), the partitioning was repeated five times, and the result was averaged.

5.2 Predecessors

In the training set, the attributes values sometimes need further refinement before starting with the learning process. In specific, predecessor procedures need to be applied over all data to deal with missing and continuous attributes.

In order to deal with continuous attributes, an offline discretization method was applied before inducing the rules. In specific, entropy-based discretization method of Fayyad and Irani [39] were applied to convert the continuous values into discrete ones. This discretization method was chosen based on the study conducted in [40, 41], where it was indicated that the choice of offline discretization method depends on the data and algorithm used. Hence, since it was empirically proven in RULES-6 and RULES-SRI that Fayyad and Irani discretization is the most appropriate discretization method for RULES algorithms then it was decided to use it.

On the other hand, when it comes to missing attributes' values, in [42] different methods that handle missing attributes have been presented and empirically tested over different rule induction methods. As a result of this test, it was found that imputation methods, especially Fuzzy K-means (FK-means) [43], are the most suitable methods to handle missing values in rule induction. Hence, FK-means method was applied over the data to fill missing attributes before inducing the rules.

5.3 Postprocessor

After conducting the experiment, and to visualize the result, different statistical analysis tests were applied. Specifically, two statistical tests were recorded to measure the accuracy of each algorithm and the rule set complexity. As for the accuracy, it was measured based on the classification accuracy of applying the resulting rules over the test data file. However, the rule set complexity was measured by recoding the average number of rules.

5.4 Evaluation Results

In order to prove that the performance of rule induction has improved by using the proposed algorithm, other covering algorithms are compared with it; including Ripper [44], DataSqueezer [4], PRISM [45, 46], RULES-6, and RULES-SRI. Using the resulting statistics of KEEL analysis tool, it was possible to record the algorithms' performance.

Table 2 presents the accuracy of the covering algorithms (single model) that have been applied over the five target data set. However, Table 3 shows the accuracy result of applying the proposed algorithm with three source models, whether the dataset is complete or has missing classes. The source models were applied over the five source tasks in order to transfer the resulting rules and apply it over the related five target tasks. As it can be noticed from these two tables, the proposed algorithm is comparable with the other covering algorithms, whether the data is complete or not. In the case of using RULES-6 or PRISM with the proposed algorithm, it was found that the accuracy has improved, in comparison to RULES family methods and DataSqueezer. Moreover, comparing with RIPPER, it can be seen that the proposed algorithm is comparable, where the accuracy difference has became only 1% to 2%. However, this is not the case with PRISM algorithm. The proposed algorithm is still lacking when compared to PRISM algorithm. Nevertheless, since it can efficiently handle missing classes to be applied over incomplete data, and it reduced the error difference between PRISM and other RULES family algorithms (RULES-6 and RULES-SRI), then the proposed algorithm is still appealing.

Table 2. Accuracy result of covering algorithm

Target Dataset	Ripper	DataSqueezer	PRISM	RULES-6	RULES-SRI
yeast	0.30	0.28	0.62	0.31	0.16
Credit Approval	0.91	0.68	0.97	0.75	0.65
Statlog	0.79	0.55	0.92	0.75	0.66
Hepatitis	0.63	0.83	0.92	0.92	0.67
White Wine	0.70	0.37	0.78	0.46	0.05
Average	**0.67**	**0.54**	**0.84**	**0.62**	**0.44**

Table 3. Accuracy result of the new algorithm

Dataset		RULES-6 à New Method		PRISM à New Method		RULES-SRI à New Method	
Source	Target	Miss Class	No Miss Class	Miss Class	No Miss Class	Miss Class	No Miss Class
ecoli	yeast	0.32	0.31	0.31	0.31	0.31	0.33
Australian Crd.	Crd. Approval	0.80	0.82	0.67	0.77	0.65	0.74
Cleveland	Statlog	0.80	0.81	0.82	0.83	0.69	0.70
Bupa	Hepatitis	0.91	0.93	0.91	0.93	0.91	0.93
Red Wine	White Wine	0.40	0.40	0.47	0.47	0.23	0.23
Average		0.65	0.65	0.64	0.66	0.56	0.59

Alternatively, when comparing the different implementation of the proposed algorithm, illustrated in Table 3, it can be noticed that the accuracy of transferring rules from PRISM and RULES-6 is comparable while transferring it from RULES-SRI tremendously reduced the accuracy. This result is normal because the accuracy of the source model RULES-SRI is very low in comparison to RULES-6 and PRISM. Hence, transferring inaccurate rules will naturally give worse result than the accurate one. Nevertheless, it must be noted that the multi-model of the proposed algorithm when using RULES-SRI is still better than the single model of RULES-SRI.

Conversely, when comparing PRISM with RULES-6, the first has better performance than the second. However, when using the two models as the source of the proposed algorithm the result is very similar, as illustrated in Table 3. This is because of the rule complexity which is measured using the rule set size. As explained in [28], the performance of an algorithm should not only be measured by the accuracy but rule set complexity should also be considered. This complexity can be measured by the size of the rule set, where small size indicates less complexity. Hence, more resulting rules reduce the performance because the rule set will not be general enough for future cases and it will be highly affected by noise.

Consequently, the average number of rules resulting from each algorithm was recorded in Table 4. As it can be noticed, the rule set size of PRISM is very large in comparison to RULES-6. Hence, transferring PRISM rules will badly affect the other domain since it is very specialized over the source domain. Thus, even through PRISM is more accurate than RULES-6 but it has grater rule set complexity. Accordingly, the result of transferring the rules from PRISM and RULES-6 was very similar. However, even though the number of rules discovered using RULES-SRI is less than the rest, but it resulted in worse performance because the single model performance is very bad.

Table 4. Average number of transferred rules

Source Dataset	PRISM	RULES-6	RULES-SRI
ecoli	42	30	25
Australian Crd.	106	31	18
Cleveland	65	27	20
Bupa	2	2	2
Red Wine	212	54	39
Average	**85**	**29**	**21**

Ultimately, it can be said that the proposed algorithm has improved the performance of RULES family, in general. It efficiently handled incomplete data and considered future cases that are not available in the training set using Transfer learning. Moreover, it improved the performance of covering algorithms where it usually resulted in better or comparable performance with other covering algorithms. Hence, it can be concluded from all above that using multi-modeling usually improve the performance of the original algorithm. However, transferring low quality or large number of rules might not greatly improve it. Thus, the models used with the proposed algorithm should be carefully chosen, where not only the accuracy of the rules is considered but also its complexity. The proposed algorithm is a good option for large and incomplete data but developers must take extra care when choosing the source model, where the general performance must be considered.

6 Conclusion

RULES is a covering algorithm family that induces general rules based on training set. It is an interesting field of covering algorithm where it gives good results in a simple way. This family needs to trade between time and accuracy during the specialization process, and incomplete data were inappropriately handled. Thus, this paper has proposed a new algorithm that takes advantage of advance machine learning methods, specifically TL, in order to transfer other agents' knowledge. This algorithm improves the performance while reducing the induction time, where transferred rules are used to fill missing classes, cover part of the dataset in advance, and use past knowledge of different models to improve the performance.

The proposed algorithm performance was tested over ten dataset and compared with other single models of RULES family and covering algorithms. It was found that, whether the dataset have missing classes or not, the proposed algorithm always outperforms its predecessors, RULES-6 and RULES-SRI. Hence, it can be stated that the proposed algorithm has improved the scalability of RULES family on complete and incomplete data. However, the type of the source model also affects such improvement; models with simple and accurate rule sets give the best result. In the future, further improvement can be made over the way test examples are classified, in order to improve the algorithm performance and outperform PRISM algorithm.

References

1. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
2. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
3. Stahl, F., Bramer, M.: Computationally efficient induction of classification rules with the PMCRI and J-PMCRI frameworks. Knowledge-Based Systems (2012)
4. Kurgan, L.A., Cios, K.J., Dick, S.: Highly Scalable and Robust Rule Learner: Performance Evaluation and Comparison. IEEE Systems, Man, and Cybernetics—Part B: Cybernetics 36, 32–53 (2006)
5. Aksoy, M.: A review of rules family of algorithms. Mathematical and Computational Applications 13, 51–60 (2008)
6. Theron, H.: An Empirical Evaluation of Beam Search and Pruning in BEXA. In: Fifth International Conference on Tools with Artificial Intelligence (TAI 1993), Boston, MA, pp. 132–139 (1993)
7. Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. Presented at the Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies (2007)
8. Maclin, R., Opitz, D.W.: Popular Ensemble Methods: An Empirical Study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
9. Pham, D., Afify, A.: Machine-learning techniques and their applications in manufacturing. Proceedings of the I MECH E Part B Journal of Engineering Manufacture 219, 395–412 (2005)
10. ElGibreen, H., Aksoy, M.: RULES – TL: A simple and Improved RULES Algorithm for Incomplete and Large Data. Journal of Theoretical and Applied Information Technology 47 (2013)
11. Sinno Jialin, P., Qiang, Y.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359 (2010)
12. Pham, D., Bigot, S., Dimov, S.: RULES-5: a rule induction algorithm for classification problems involving continuous attributes. In: Institution of Mechanical Engineers, pp. 1273–1286 (2003)
13. Ramon, J., Driessens, K., Croonenborghs, T.: Transfer Learning in Reinforcement Learning Problems Through Partial Policy Recycling. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 699–707. Springer, Heidelberg (2007)

14. Taylor, M., Suay, H.B., Chernova, S.: Integrating Reinforcement Learning with Human Demonstrations of Varying Ability. In: International Conferance of Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan (2011)
15. Mahmud, M.M.H.: On Universal Transfer Learning. In: Hutter, M., Servedio, R.A., Takimoto, E. (eds.) ALT 2007. LNCS (LNAI), vol. 4754, pp. 135–149. Springer, Heidelberg (2007)
16. Yongqiang, L.: A Review About Transfer Learning Methods and Applications. In: International Conference on Information and Network Technology IPCSIT, Singapore, pp. 7–11 (2011)
17. Pham, D.T., Aksoy, M.S.: RULES: A simple rule extraction system. Expert Systems with Applications 8, 59–65 (1995)
18. Pham, D.T., Aksoy, M.S.: An algorithm for automatic rule induction. Artificial Intelligence in Engineering 8, 277–282 (1993)
19. Pham, D.T., Aksoy, M.S.: A new algorithm for inductive learning. Journal of Systems Engenering 5, 115–122 (1995)
20. Pham, D.T., Dimov, S.S.: The RULES-3 Plus inductive learning algorithm. In: Proceedings of the Third World Congress on Expert Systems, Seoul, Korea, pp. 917–924 (1996)
21. Aksoy, M.S., Mathkour, H.: CSLS: Connectionist Symbolic Learning System. Mathematical and Computational Applications 14, 177–186 (2009)
22. Pham, D.T., Dimov, S.S.: An algorithm for incremental inductive learning. Journal of Engineering Manufacture 211, 239–249 (1997)
23. Pham, D.T., Soroka, A.J.: An Immune-network inspired rule generation algorithm (RULES-IS). In: Third Virtual International Conference on Innovative Production Machines and Systems. Whittles, Dunbeath (2007)
24. Pham, D.T., Bigot, S., Dimov, S.S.: RULES-F: A fuzzy inductive learning algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 220, 1433–1447 (2006)
25. Bigot, S.: A new rule space representation scheme for rule induction in classification and control applications. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering (2011)
26. Pham, D.T., Afify, A.A.: RULES-6: A Simple Rule Induction Algorithm for Supporting Decision Making. Presented at the 31st Annual Conference of IEEE Industrial Electronics Society (IECON 2005) (2005)
27. Shehzad, K.: EDISC: A Class-tailored Discretization Technique for Rule-based Classification. IEEE Transactions on Knowledge and Data Engineering 24, 1435–1447 (2012)
28. Afify, A.A., Pham, D.T.: SRI: A Scalable Rule Induction Algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 220, 537–552 (2006)
29. Pan, W., Zhong, E., Yang, Q.: Transfer Learning for Text Mining, pp. 223–257 (2012)
30. Xie, Y.-F., Su, S.-Z., Li, S.-Z.: A Pedestrian Classification Method Based on Transfer Learning. Presented at the International Conference on Image Analysis and Signal Processing, IASP, Zhejiang (2010)
31. Estévez, J.I., Toledo, P.A., Alayón, S.: Using an Induced Relational Decision Tree for Rule Injection in a Learning Classifier System. Presented at the IEEE Congress on Evolutionary Computation New Orleans, LA (2011)
32. Boström, H.: Induction of Recursive Transfer Rules. In: Cussens, J., Džeroski, S. (eds.) LLL 1999. LNCS (LNAI), vol. 1925, pp. 237–450. Springer, Heidelberg (2000)

33. Reid, M.D.: DEFT Guessing: Using Inductive Transfer to Improve Rule Evaluation from Limited Data. In: Doctor of Philosophy, School of Computer Science and Engineering. University of New South Wales, Sydney (2007)
34. Ganchev, P., Malehorn, D., Bigbee, W.L., Gopalakrishnan, V.: Transfer learning of classification rules for biomarker discovery and verification from molecular profiling studies. Journal of Biomedical Informatics 44(suppl. 1), S17–S23 (2011)
35. Bigot, S.: A study of specialisation and classification heuristics used in covering algorithms. Presented at the IPROM2009 Innovative Production Machines and Systems Fifth I*PROMS Virtual Conference, Cardiff, UK (2009)
36. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. Soft Computing 13, 307–318 (2009)
37. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17, 255–287 (2011)
38. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, USA (1993)
39. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuousvalued attributes for classification learning. Presented at the 13th International Joint Conference of Artificial Intelligence (1993)
40. Cai, Z.: Technical Aspects of Data Mining. PhD, University of Wales Cardiff, Cardiff, UK (2001)
41. Pham, D.T., Afify, A.A.: Online Discretization of Continuous-Valued Attributes in Rule Induction. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 219, 829–842 (2005)
42. Luengo, J., García, S., Herrera, F.: On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowledge and Information Systems 32, 77–108 (2012)
43. Li, D., Deogun, J., Spaulding, W., Shuart, B.: Towards Missing Data Imputation: A Study of Fuzzy K-means Clustering Method. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 573–579. Springer, Heidelberg (2004)
44. Cohen, W.W.: Fast Effective Rule Induction. In: Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
45. Stahl, F., Bramer, M., Adda, M.: P-Prism: A Computationally Efficient Approach to Scaling up Classification Rule Induction. In: Bramer, M. (ed.) Artificial Intelligence in Theory and Practice II. IFIP, vol. 276, pp. 77–86. Springer, Heidelberg (2008)
46. Stahl, F., Bramer, M.: Induction of Modular Classification Rules: Using Jmax-pruning, pp. 79–92 (2011)

3D Geovisualisation Techniques Applied in Spatial Data Mining

Carlos Roberto Valêncio, Thatiane Kawabata, Camila Alves de Medeiros,
Rogéria Cristiane Gratão de Souza, and José Márcio Machado

São Paulo State University
Departamento de Ciências de Computação e Estatística
Rua Cristóvão Colombo, 2265, São José do Rio Preto, São Paulo, Brazil
{valencio,rogeria,jmarcio}@ibilce.unesp.br,
{thatianekawabata,camila.alves.medeiros}@gmail.com

Abstract. The increase in the number of spatial data collected has motivated the development of geovisualisation techniques, aiming to provide an important resource to support the extraction of knowledge and decision making. One of these techniques are 3D graphs, which provides a dynamic and flexible increase of the results analysis obtained by the spatial data mining algorithms, principally when there are incidences of georeferenced objects in a same local. This work presented as an original contribution the potentialisation of visual resources in a computational environment of spatial data mining and, afterwards, the efficiency of these techniques is demonstrated with the use of a real database. The application has shown to be very interesting in interpreting obtained results, such as patterns that occurred in a same locality and to provide support for activities which could be done as from the visualisation of results.

Keywords: Geovisualisation, Spatial Data Mining, Database, Geographic Information System.

1 Introduction

The expressive amount of stored spatial data in the system has awakened the interest of various areas of study. As the data is of high complexity, the task of extracting knowledge became very costly so, therefore, computational spatial data mining systems appeared [2], [6], [10], [16].

Nevertheless, results were many times not easily visible nor understood, principally in the case of spatial data in the same geographic coordinates [5]. To overcome this obstacle, 3D graphs were developed following geovisualisation guidelines together with an analysis of the region of the map.

The approach by means of a 3D visualisation technique made possible interactivity and flexibility of the spatial data mining results [4]. The implementation of that technique is described in this article, as well as its application with a knowledge extraction system. The experimental results were done based on the work accidents real data, which validates the efficiency of this work.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 57–68, 2013.

This article is organised in the following manner: section 2 presents the theoretical substantiation; section 3 describes the development of the work; section 4 shows the experimental results from the developed work and section 5 the conclusions.

2 Theoretical Substantiation

The notorious increase of georeferenced data collected by new technological systems such as GPS, remote sensing, spatial localising systems among others, has intensified studies in this area [7], [13]. A new concept has emerged, the spatial data mining or Knowledge Discovery in Spatial Databases (KDSD) to discover implicit patterns in the correlation of spatial and non-spatial attributes [12]. Among these techniques is the geovisualisation that has, as a principal, to facilitate the understanding and decision making from the results of the spatial data mining stage and to emphasise the interactivity for exploring knowledge [5], [10].

Geovisualisation has shown itself to be a very important technique for the analysis of results, as it is not necessary to have an extensive knowledge about the technique to be able to use it. The technique addresses three important points: data processing, due to the large amount and variety of the spatial data; the movement of diverse variables to discover implicit patterns; and an interface that is easy to use and understand [8], [10].

Various spatial data mining systems can be found in literature together with their geovisualisation guidelines to extract knowledge, such as: 3D mono-colour graph for the analysis of outliers [3] that are discrepant points in a set of data; 3D multicolour graphs to show different types of rocks and to analyse the wearing out in certain regions [1]; among other relevant works related to this area.

This work covers both the 3D multicolour graphs for each attribute chosen in the data analysis, as well as the manipulation of these resources from any desired angle, besides the relationship of the locality to the graph as from a highlighted geographical point on the map and the description of the information about the selected georeferenced data.

The implemented techniques can be taken as an effective contribution, since no works were found in literature that adopt these techniques for the analysis of results obtained from spatial data mining algorithms.

3 Developed Work

The developed set of visual resources is coupled to a spatial clustering system, which makes for a better understanding of results obtained with knowledge extraction process of this system [9], [14].

With the potentialisation of visual resources, it is possible to obtain implicit information from the results of a cluster, since a large quantity of georeferenced objects may be in a same spatial locality. In Fig. 1, is possible to see the registers contained in a cluster of a determined region, which implies a strong correlation between the spatial and non-spatial data, since these clusters are formed from specific criteria, such as: lost work time (time absent from work due to accident), sex and branch of activity. Each square of the map refers to a geographic position that may have many georeferenced registers in that locality, not visualised in that perspective.

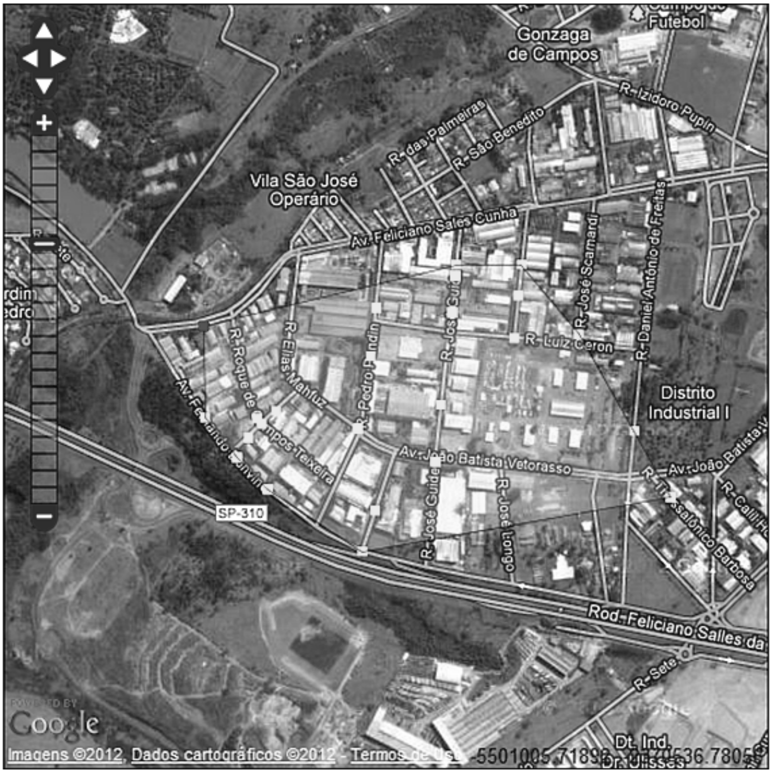


Fig. 1. Spatial cluster

In view of that occurrence, a three dimension and multicolour graph resource was implemented which favours the visualisation of different characteristics. The 3D graph provided a rapid visualisation of the distribution of georeferenced points in the cluster, as well as a quantity in certain regions. This is relevant in an analysis, because the visualisation of a map in two dimensions makes it impossible to see the number of georeferenced points in a same locality and to make a comparison to its neighbours.

The 3D graph has interactivity with the user, that is, the graph can be seen from various angles, its focus increased or decreased, making it easier to discover implicit patterns as shown in Fig. 2.

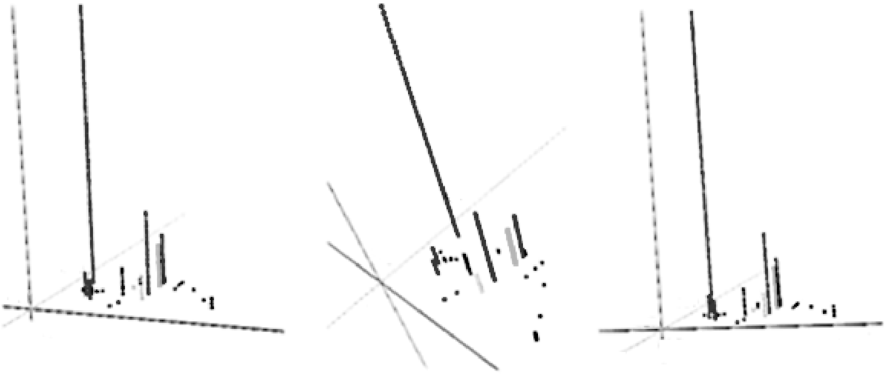
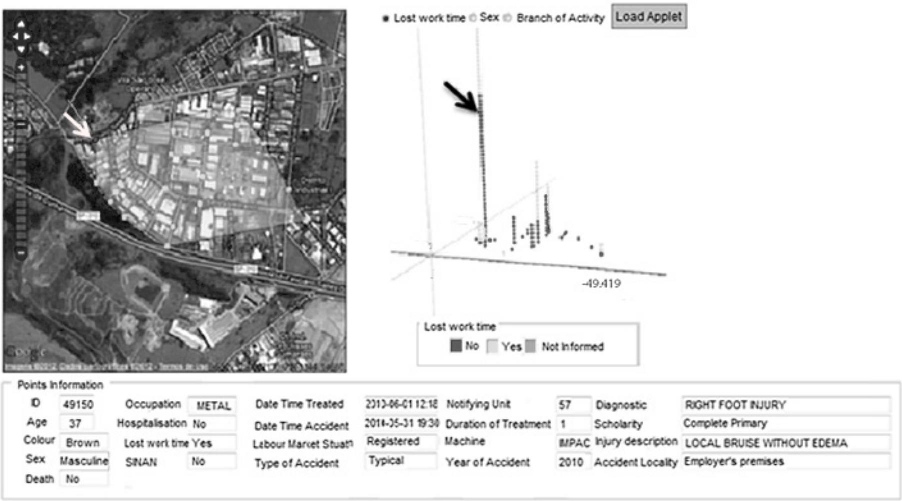


Fig. 2. 3D graph in different angles

Finally, there is a relationship between the map and the graph, in that, on pressing on a point in the 3D graph, its location on the map is shown and it is also possible to verify all its characteristics. Fig. 3 shows an example of the application of the visualisation techniques and implemented resources.


● Lost work time ○ Sex ○ Branch of Activity
Load Applet
-49.419
Lost work time
■ No □ Yes ▨ Not Informed
Google

Points Information

ID	49150	Occupation	METAL	Date Time Treated	2010-06-01 12:18	Notifying Unit	57	Diagnostic	RIGHT FOOT INJURY
Age	37	Hospitalisation	No	Date Time Accident	2010-05-31 19:30	Duration of Treatment	1	Scholarity	Complete Primary
Colour	Brown	Lost work time	Yes	Labour Market Stuatn	Registered	Machine	IMPAC	Injury description	LOCAL BRUISE WITHOUT EDEMA
Sex	Masculine	SINAN	No	Type of Accident	Typical	Year of Accident	2010	Accident Locality	Employer's premises
Death	No								

Fig. 3. Overall vision of the system

4 Experimental Results

A system namely SIVAT - *Sistema de Informação e Vigilância de Acidentes do Trabalho* (Work Accidents Vigilance System) catalogues all work accident and contains the georeferenced data about the places where those accidents occurred so as to assist health area actions. Said system is used by the CEREST - *Centro de Referência em Saúde do Trabalhador* (Worker Health Reference Centre) in the cities of São José do Rio Preto and Ilha Solteira, both in the interior of São Paulo State, to collect, register, follow-up and manage work accident notifications [15].

To summarise the adopted approach so as to optimise the visual resources applied to the spatial data mining, a database of work accident notifications, having more than 70 thousand registers with more than 20 thousand of them georeferenced, was used.

That repository has the following stored data: occupation of the injured person, labour market situation, cause of accident, if hospitalisation and absence from work was necessary, among others [15]. Said repository is updated daily and an analysis of the data is done periodically which justifies the relevancy of the proposed work and its contribution to that activity.

To do the experiments, a CLARANS - Clustering Large Applications based on Randomised Research [11] algorithm of the spatial clustering system was selected so that results could be analysed and enriched by the resources implemented in this work. Two executions of the algorithm were done: the first about the year 2009 accidents and the second about 2010, and in both was used the same parameters, showed in Table 1.

Table 1. Parameters used in the CLARANS algorithm

Number of centroids	20
Maximum iterations	50
Maximum points in each cluster	20
Maximum neighbours	30

4.1 Experiment 1: Cluster of 2009 Data

The following attributes were chosen for this experiment: lost work time, sex and branch of activity in the year 2009. The analysed cluster was made up of 119 work accidents, shown in Fig. 4.

By means of the implemented resources, it is possible to observe that, in the 3D graphs of Fig. 5, Fig. 6 and Fig. 7, it was seen that, for each chosen attribute, the cluster had many more accidents than those plotted in the two dimensional map. This is due to the fact that many happened in the same place.

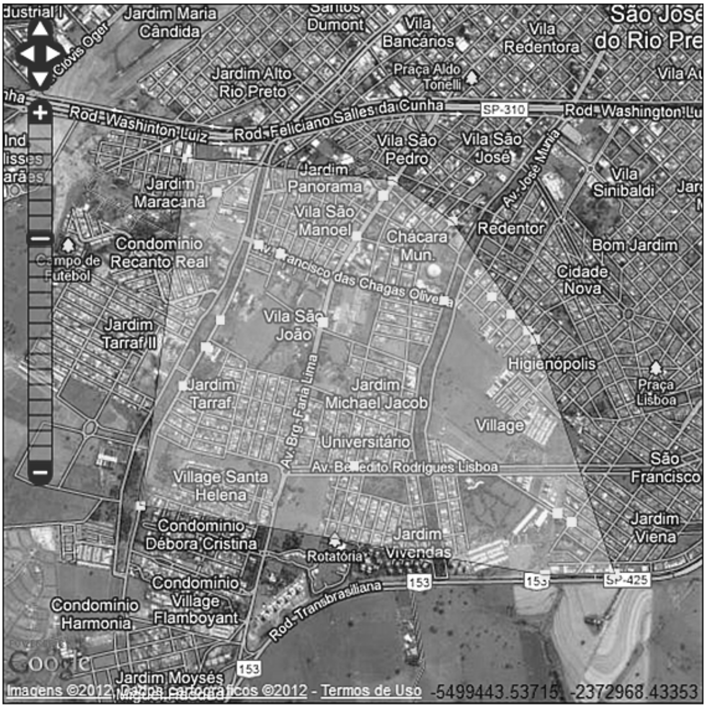


Fig. 4. Cluster used for analysis – 2009

Accidents in this cluster resulted in 100% of lost work time and the principal branches of activity were; hospitals, clinics, laboratories, supermarkets, markets, shops, mini-markets and metalwork, with 32% women and 68% men.

With the support of the map in Fig. 4, it is possible to see the region in which such characteristics are concentrated. 3D graphs in Fig. 5, Fig. 6 and Fig. 7 show the occurrences of work accidents at each point for each characteristic besides discovering new correlations, as for example, on analysing the graphs in Fig. 6 and Fig. 7 together, it can be seen that most of the accidents in the hospital, clinic, and laboratory branches of activity happened in the same locality (as can be seen by the signalise vertical line in the graph of Fig. 7) and, on analysing the same geographic position in the graph of Fig. 6, it can be verified that most of those accidents were with women. Moreover, the map's zoom tool permits visualising the exact location of those accidents, thus giving the CEREST, the agency responsible for work accidents vigilance, an indicative for the planning of preventive and corrective strategies for those accidents in the right region or even in a specific locality that has such characteristics.

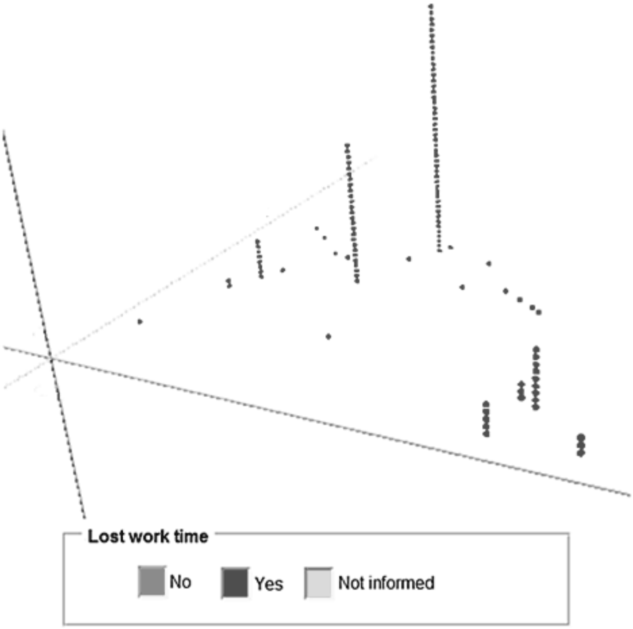


Fig. 5. D graph generated for the lost work time attribute

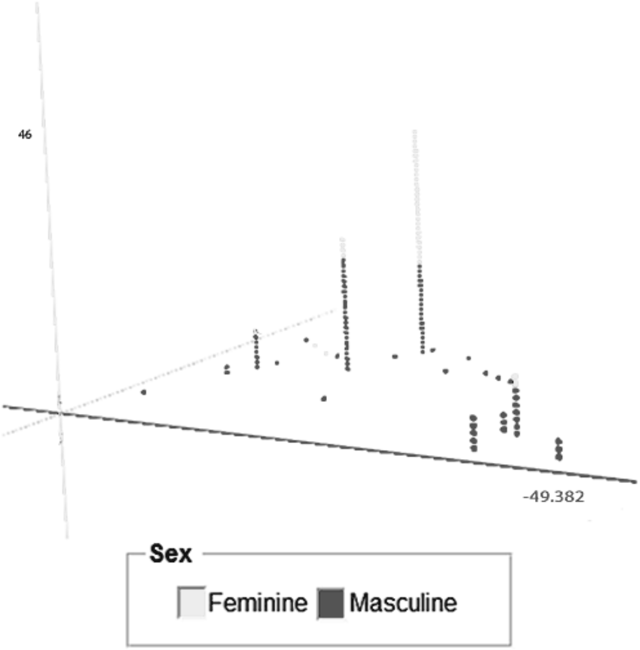


Fig. 6. 3D graph generated for the activity attribute

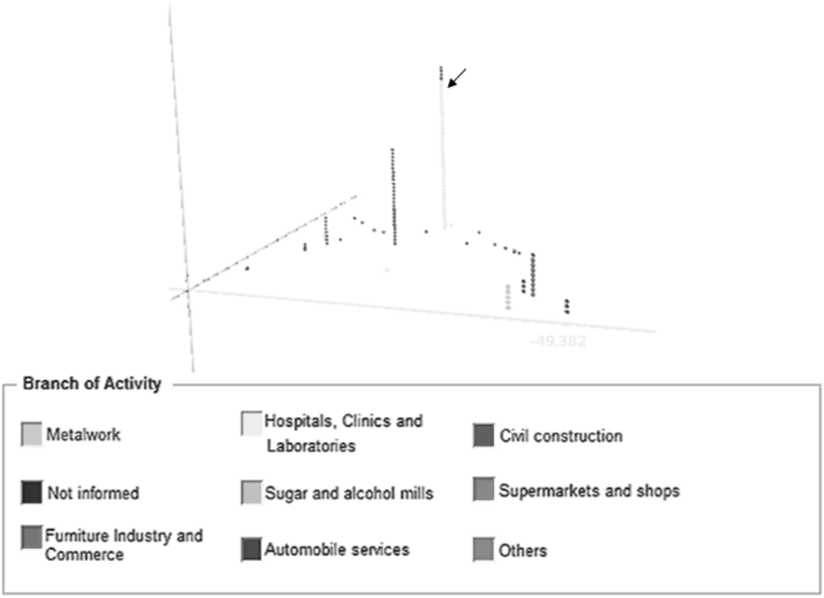


Fig. 7. 3D graph for the sex attribute

4.2 Experiment 2: Cluster of Year 2010 Data

Experiment 2 was done from year 2010 data for the following attributes: lost work time, hospitalisation and the period of the day in which the accident occurred. To better identify the periods of the day, they were discretised as dawn, morning, afternoon and night. The cluster in Fig. 8 only shows the points that make up those periods.

As can be seen, the cluster covers the Washington Luiz Highway region and surrounding streets. A very interesting point of this cluster is the fact that all the accidents occurred during the afternoon (13:00 to 18:59) as can be seen in Fig. 9.

Although the cluster covered the Washington Luiz Highway, the accidents in that local did not result in the worker being absent from neither his activities nor being hospitalised. Fig. 10 and Fig. 11 show the 3D graphs for lost work time and hospitalisation.

The cluster shown in Fig. 8 also covers work accidents in one of the principal streets of the municipality (Independência street), accounting for 91 work accidents, which allied to the fact that they all occurred during the afternoon, is an indication that the risk of accidents on this street are higher in that period than in others and, therefore, preventative measures can be taken by the responsible agencies to avoid them.

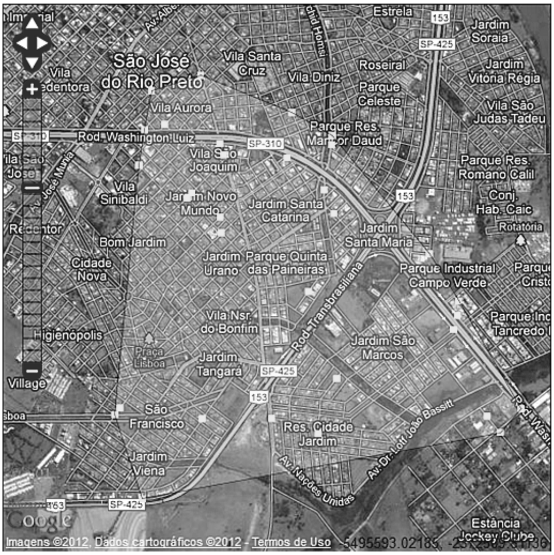


Fig. 8. Cluster used for analysis – 2010

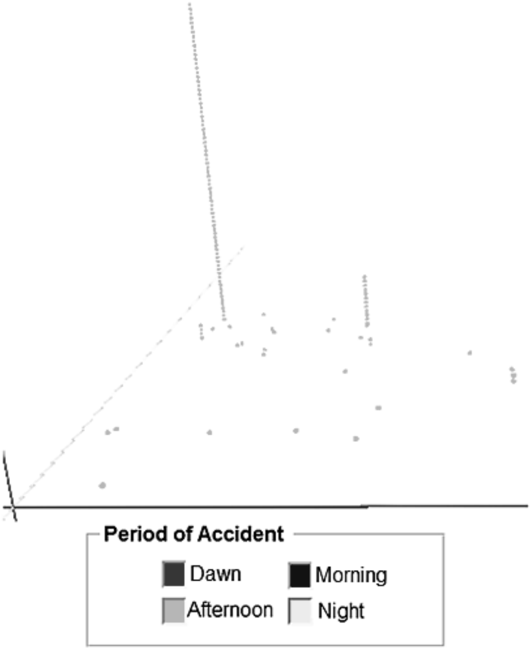


Fig. 9. 3D Graph for a day

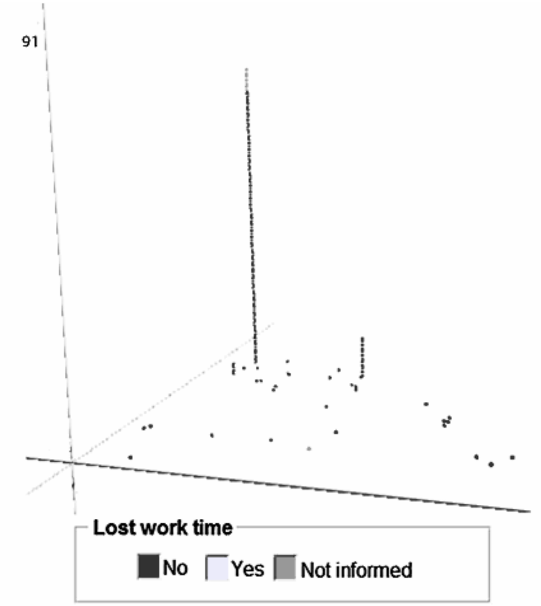


Fig. 10. 3D Graph for the lost work time attribute

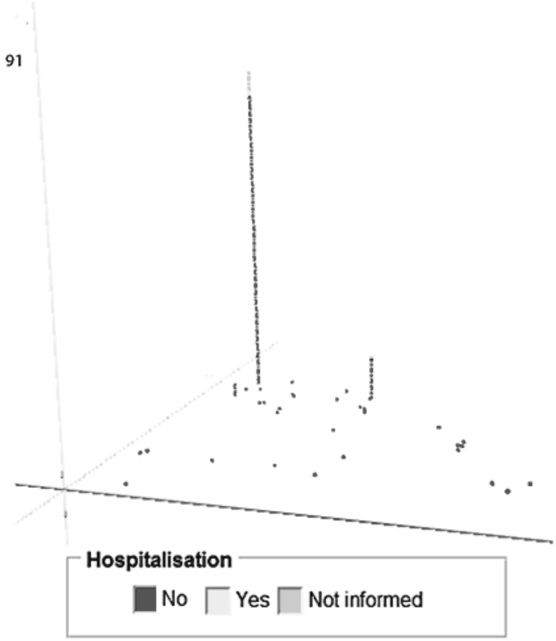


Fig. 11. 3D Graph for the hospitalisation attribute

5 Conclusions

A large number of valuable discoveries can be done on spatial databases by means of the spatial data mining algorithm. To make these discoveries, a system was constructed that does the spatial clustering and can visualise said clusters on a map of the region in which the points are shown.

To support with the understanding of obtained results, as well as to permit new correlations, a geovisualisation 3D technique was incorporated that makes it possible to visualise and search for patterns in certain regions in which the incidence of georeferenced data is high. 3D graphs that can be rotated by the user were constructed to permit a more adequate visualisation of the results. Moreover, a filter was developed of the points that are not part of the cluster, the information related to their georeferenced data and their location on the map. That visual interaction of the generated clusters, together with the map and the 3D graphs, represent a new solution for the extraction of knowledge from spatial databases, offering the user various interactive resources that analyse results, permitting a wider vision of the georeferenced data, as well as supporting with decision making in certain localities.

The use of the proposed system on a real database of work accident reports revealed some interesting spatial correlations that, by means of the 3D graphs and the visualisation of points on the map, enabled the CEREST to plan specific strategies in accordance with the localities and accident characteristics.

Therefore, this work proves to be relevant in the computational area by means of frontier technologies as well as for public health, since work accidents are responsible for jeopardising workers due to corporal wounds and great expense for public agencies that deal with those workers.

References

1. Amirbekyan, A.: Applying Data Mining and Mathematical Morphology to Borehole Data Coming from Exploration and Mining Industry. In: 2010 IEEE Sixth International Conference on E-Science, e-Science 2010, Brisbane, Australia, December 7-10, pp. 113–120. IEEE Computer Society (2010)
2. Bae, D.-H., Baek, J.-H., Oh, H.-K., Song, J.-W., Kim, S.-W.: SD-Miner: A spatial data mining system. In: Proc. 2009 IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2009, Beijing, China, November 6-8, pp. 803–807. IEEE (2009)
3. Cai, Q., He, H., Man, H.: SOMSO: A self-organizing map approach for spatial outlier detection with multiple attributes. In: International Joint Conference on Neural Networks, IJCNN 2009, Atlanta, USA, June 14-19, pp. 425–431. IEEE Computer Society (2009)
4. Compieta, P., Di Martino, S., Bertolotto, M., Ferrucci, F., Kechadi, T.: Exploratory spatio-temporal data mining and visualization. Journal of Visual Languages & Computing 18, 255–279 (2007)
5. Han, J.: Geographic Data Mining and Knowledge Discovery, 2nd edn. CRC Press, Taylor & Francis Group (2009)

6. Ji, M., Jin, F., Zhao, X., Ai, B., Li, T.: Mine geological hazard multi-dimensional spatial data warehouse construction research. In: 2010 18th International Conference on Geoinformatics, Beijing, China, June 18-20, pp. 1–5. IEEE (2010)
7. Jin, H., Miao, B.: The research progress of spatial data mining technique. In: 2010 3nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010, Chengdu, China, July 9-11, vol. 3, pp. 81–84. IEEE (2010)
8. Li, C., Li, F., Tian, Y.: Geovisualization of knowledge diffusion a case study in data mining. In: 2nd International Conference on Computer Engineering and Technology, ICCET 2010, Chengdu, China, April 16-18, vol. 2, pp. V2-590–V2-595. IEEE (2010)
9. Medeiros, C.A., Ichiba, F.T., Souza, R.C.G., Valencio, C.R.: Ferramenta de Apoio ao Spatial Data Mining. In: International Association for Development of the Information Society Ibero-Americana Conference on WWW/Internet, Rio de Janeiro, Brazil, November 5-7, pp. 396–398. IADIS Press (2011) (in Portuguese)
10. Mennis, J., Guo, D.: Spatial data mining and geographic knowledge discovery—An introduction. Computers, Environment and Urban Systems 33, 403–408 (2009)
11. Ng, R.T., Han, J.: Efficient and Effective Clustering Methods for Spatial Data Mining. In: Proc. 20th International Conference on Very Large Data Bases, VLDB 1994, Santiago de Chile, Chile, pp. 144–155. Morgan Kaufmann Publishers (1994)
12. Pattabiraman, V., Parvathi, R., Nedunchezian, R., Palaniammal, S.: A Novel Spatial Clustering with Obstacles and Facilitators Constraint Based on Edge Deduction and K-Mediods. In: 2009 International Conference on Computer Technology and Development, ICCTD 2009, Kota Kinabalu, Malaysia, November 13-15, vol. 1, pp. 402–406. IEEE Computer Society (2009)
13. Shun, H.Y., Wei, X.: A study of spatial data mining architecture and technology. In: 2009 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009, Beijing, China, August 8-11, pp. 163–166. IEEE Computer Society Press (2009)
14. Valêncio, C.R., Medeiros, C.A., Ichiba, F.T., Souza, R.C.G.: Spatial Clustering Applied to Health Area. In: 2011 12th International on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2011, Gwangju, Korea, October 20-22, pp. 427–432. IEEE Computer Society (2011)
15. Valêncio, C.R., Oyama, F.T., Scarpelini Neto, P., Colombini, A.C., Cansian, A.M., Souza, R.C.G., Corrêa, P.L.P.: MR-Radix: a multi-relational data mining algorithm. Human-centric Computing and Information Sciences (HCIS), SpringerOpen Journal 2, 1–17 (2012)
16. Wang, P., Ma, L., Xi, Y., Jin, L.: Research on Logistics Oriented Spatial Data Mining Techniques. In: 2009 International Conference on Management and Service Science, MASS 2009, Wuhan, China, September 20-22, pp. 1–4. IEEE (2009)

Improving the Efficiency of Distributed Data Mining Using an Adjustment Work Flow

Jie Gao and Jörg Denzinger

Department of Computer Science,
University of Calgary Calgary, Canada
jie.gao.email@gmail.com, denzinge@cpsc.ucalgary.ca

Abstract. We present an extension of the usual agent-based data mining cooperative work flow that adds a so-called adjustment work flow. It allows for the use of various knowledge-based strategies that use information gathered from the miners and other agents to adjust the whole system to the particular data set that is mined. Among these strategies, in addition to the basic exchange of hints between the miners, are parameter adjustment of the miners and the use of a clustering miner to select good working data sets. Our experimental evaluation in mining rules for two medical data sets shows that adding a loop with the adjustment work flow substantially improves the efficiency of the system with all the strategies contributing to this improvement.

1 Introduction

Data mining has become an important tool for decision makers in all kinds of areas. Tools like Weka (see [5]) are available to provide a wide variety of data mining algorithms but also methods for preparing data for mining and analyzing mining results under a common user interface. Despite the efforts in these tools to simplify the task of running the entire data mining work flow, finding suitable algorithms, parameters for each algorithm, and relevant sub sets of data to work on remains a very time-consuming and difficult task.

Using groups of cooperating mining agents (miners) presents a first step towards a solution to overcome the difficulties mentioned above, since the agents can try out different mining methods in parallel and, by exchanging good mining results and other information, they can essentially create a "super"-miner combining the strengths of the individual miners. As works like [4] or [3] (see also [11] for an overview) have shown, this combination can achieve substantial synergetic gains. But there are also quite a number of problems around agent-based approaches. These include the decision when to communicate what to which other agents, as well as focusing individual agents on the right parts of the data at the right time and detecting and replacing useless agents.

In this paper, in order to solve the problems from the last sentence, we propose an approach named CoLe^2 that enhances the CoLe cooperative mining system (see [4]), by adding an outer *adjustment* work flow to the existing *cooperative*

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 69–83, 2013.

work flow. This outer work flow essentially iterates over the usual preparation-mining-analysis data mining work flow to automate the process of finding the right set-up for the miners to create the desired results efficiently. To achieve this, the CoLe^2 system uses so-called knowledge-based strategies that incorporate knowledge about data mining in general, the mentioned three phases, the miners, and how the miners act in the cooperative environment.

Our CoLe^2 approach uses the adjustment work flow to perform iterative data selection for the miners using an X-means clustering based method (see [12]), which runs asynchronously with the inner distributed cooperation work flow. In addition, it selects miners based on their performance history in the inner cooperation iterations. It also does parameter adjustment for the miners during the mining which includes feature selection based on hints from the other miners. And we use so-called combiners that form additional more complex rules out of the results of the miners.

We instantiated the general concepts of CoLe^2 in a system aimed at mining rules about two medical databases, one focussing on diabetes (as in [4]) and the other focussing on chronic kidney disease. In our experiments we compared this system to CoLe and also looked at the individual knowledge-based strategies and their contributions. Our experiments showed that CoLe^2 substantially improves the efficiency of the mining while creating rules of the same or even better quality. The experiments also showed that all knowledge-based strategies contribute to this improvement.

2 Data Mining Basics

Data mining is the attempt to extract patterns or knowledge from data with the help of computer programs. The typical data mining process involves 4 steps (see [6]): data cleaning and integration, data selection and transformation, executing a data mining algorithm, and evaluating and presenting the results. There are many different types of data mining tasks within the mining process. The most common ones are classification, clustering and association analysis. There are also various representations for knowledge for these different tasks, like decision trees, neural networks or rules. While our CoLe and CoLe^2 concepts can be applied to all the tasks and to most of the knowledge representations, the system we use for evaluating CoLe^2 is performing data mining of rules for classifying patients.

In general, a (relational) database D can be seen as a number of tables T_1,...,T_l, where a table has a set of attributes A_1,...,A_m. Then an entry in a table is a tuple $(a_1, ..., a_m)$, where a_i is a value for A_i. A rule over D has the form *condition* ⇒ *consequence* where *condition* is an expression consisting of predicates and certain operators, as is *consequence* (although *consequence* is usually very short). Predicates are about values of the attributes, usually having the form *att rel-op value*, where *rel-op*, a relational operator, compares the attribute *att*'s value of an entry in D with *value*. In our system, we allow $=$, $\neq$, $>$, $\geq$, $<$, and $\leq$ as relational operators. On the condition level, we use a logical operator (*and*)

and a temporal operator (*before*), together with parenthesis which override the precedence of the operators. The (*or*) logical operator is not explicitly used in the rules. Instead, the rules in a rule set have the (*or*) relation between each other.

A rule is true for an entry in a table if whenever the *condition* is true for the attribute values in the tuple, then also the *consequence* is true. Such an entry is a true positive. On the contrary, if for a tuple *condition* is true, but *consequence* is false, then this tuple is a false positive for the rule. A false negative for a rule is a tuple that does not fulfill its *condition*, but the *consequence* of the rule is true. A general goal of mining of rules is to find rules with very few (preferably zero) false positives. And the combination of all mined rules should have very few (preferably zero, again) false negatives, if classification is the goal. Based on this, there are various measures for the quality of a rule in the literature. For example, accuracy is the ratio of the number of true positives to the sum of true and false positives for the rule. Generalness is the percentage of true positives out of all entries in D for which the consequence of the rule is true. In our system in Section 4, we will combine these two measures into a *rule fitness*.

3 CoLe^2

In this section, we present our agent-based CoLe^2 data mining model. We first provide a general overview, then present the cooperative work flow that is essentially identical to the CoLe work flow from [4] and finally look at the new outer adjustment work flow loop and some knowledge-based strategies that are possible to use due to this loop.

3.1 The CoLe^2 Model: Overview

CoLe^2 extends the CoLe approach (CoLe stands for **Co**operative **Le**arning) by having two loops (hence CoLe squared). It has not only a loop where several miners perform data mining iterations in parallel – the cooperative work flow, but also a loop iterating over the data selection, data mining, and result evaluation sequence of the usual data mining work flow, which we call the adjustment work flow. This produces the general work steps for a CoLe^2-based system as depicted in Figure 1. Naturally, details of these agents and steps depend on the concrete application. Here we provide a high-level view. An application example is provided in the next section.

Within a CoLe^2-based system, there are 3 types of agents, a controller, several miners, and one or more combiners. The controller is the agent in charge of the whole CoLe^2 work flow, realizing Steps 2, 3 and 6 in Figure 1 (naturally with help from the other agents). The controller is also responsible for Step 1 and Step 7 of Figure 1, but these steps are rather standard and not really part of the distributed agent-based approach. We will look more closely at the controller and its various tasks in the next two subsections.

The miner agents obviously are at the core of the approach, performing the mining of the data they are given. In contrast to a standard mining algorithm,

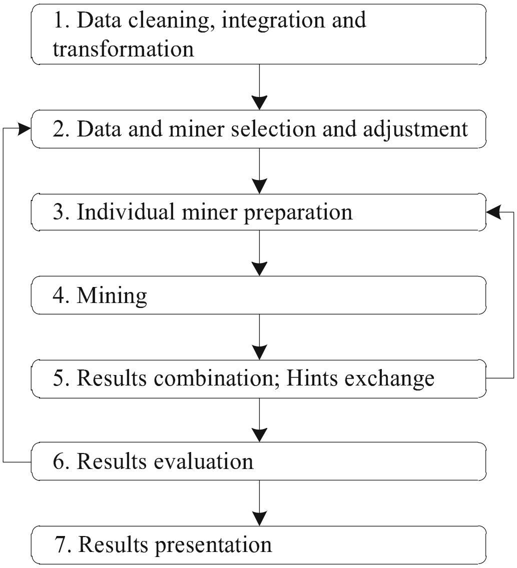

1. Data cleaning, integration and transformation
2. Data and miner selection and adjustment
3. Individual miner preparation
4. Mining
5. Results combination; Hints exchange
6. Results evaluation
7. Results presentation

Fig. 1. CoLe^2 work flow

they have to additionally be able to make use of information communicated from other miners that run concurrently and from combiners, so-called hints, into their mining as much as possible. How much use a miner can make of hints depends on the concrete mining algorithm the miner uses. We use a group of miners that are heterogeneous, i.e. they all use different mining algorithms, although it would be possible to also have some miners using the same mining algorithm, if too much redundant work by such miners can be avoided (by, for example, having them focus on different parts of the data). Importantly, as demonstrated in CoLe, the miners can also be heterogeneous in the kind of knowledge they produce. CoLe used miners for different kinds of rules (conjunctive ones that use only *and* and temporal ones that use only $before$) and the combiners allowed to create rules combining the types. CoLe^2 allows also to use miners that do clustering to help with the adjustment work flow (to provide miners with better focus, see Section 3.3).

In general, combiner agents, as the name suggests, collect data mining results from several miners and combine them into single rule sets (i.e. realizing Step 5 in Figure 1). This goes beyond just creating the union of these rule sets. Instead, as mentioned above, the combiners use knowledge-based combination strategies to create out of pieces derived from the results of the various miners new hybrid rules. Combiners can be used at the end of each cooperation phase, in which case they are also generating hints that are made available to the miners and the controller. But a combiner can also be used as the final step of the result evaluation and to help the controller with the result preparation.

3.2 The Cooperative Work Flow

The inner loop around the cooperative work flow in CoLe^2 is still the core of a CoLe^2-based cooperative mining system. Multiple data mining agents using different mining algorithms work in parallel on creating new rules. In the first iteration of the cooperative work flow, a miner receives the data it is supposed to mine, together with any additional information the controller deems necessary for this miner (like values for its parameters). In the other iterations, a miner is only receiving information (hints) from other miners and the combiner. This information in addition to the data is used by the miner during the mining (Step 4 in Figure 1), but also in the preparation phase (Step 3) of the miner.

The preparation phase of the work flow is performed by each miner individually. In addition to the integration of information from the other agents, this allows the miner to do its individual preparation based on the requirements and characteristics of the concrete mining algorithm it uses. Usually nearly all such algorithms can profit from *attribute selection*, techniques to reduce the number of attributes in the data, which usually leads to a smaller amount of data, potentially less sparse, and faster run times. In our example application, we use an attribute selection technique based on relevance factors (as in [10]) and the hints from other agents for two of our miners (PART and the Apriori miner). The relevance factor is based on the idea that an attribute is relevant for a particular concept, if it has different distributions over data instances that are in the concept and instances that are not in the concept. Attributes that are relevant (based on a threshold regarding the difference of the distributions) get preferential treatment by the miners when creating rules. The same is true for attributes that occur often (again, defined via a threshold) in hints from other miners.

The preferential treatment during the mining phase, as mentioned before, is the main modification that we make in miners for usage in CoLe^2. Predicates using relevant attributes and predicates occurring in the best rules from other miners (as determined by the combiner, see below) should be chosen with higher probability whenever a miner has a choice that involves predicates. While from an implementation point of view, this is usually easy to accomplish (the choice needs to be made, so there must be a clearly defined place making an evaluation and only this evaluation needs to be modified), it is naturally very specific to the particular miner.

The final phase of the cooperative work flow is the domain of the combiners. Each miner sends its best rules (based on the rule fitness) to one combiner (in our example application we use only this one combiner, but it is naturally possible to use several), which then performs four subtasks: original rule evaluation, candidate rule generation, combined rule evaluation and pruning, and hint generation. It has to be noted that a combiner has to be able to handle different kinds of knowledge (hybrid rules in our case). The first subtask re-evaluates the rules from the miners, allowing for a different set of data entries to be used in the fitness evaluation (for example the combination of the sets of the different miners, if they got different data sets from the controller), but also collects information on the frequency and quality of predicates or groups of predicates to be used in the next subtask.

In this next subtask, good predicates and groups of predicates are connected by logical or temporal operators to construct candidate rules (that now often contain pieces of knowledge from different miners). Naturally, there are many different ways how such combinations can be done, which means that there are many possible combiner agents. One simple way is to just concatenate the conditions from two original rules, but also more sophisticated methods are possible, like adding predicates to an original rule that have high potential for improvement and are connected using operators not used in the original rule.

In the third substep, these candidate rules are evaluated and pruned. The evaluation computes the fitness of the new rules (using the combiner's data set) and only rules above given thresholds survive and will be put into the result set for this iteration of the outer loop. Those rules coming from the miners that are also above the thresholds and that are not covered by a combined rule generated out of it will also be put into this result set. Finally, the combiner will also use the predicates and groups of predicates it identified as of good quality as hints that are sent to the miners for the next iterations of the cooperative loop.

3.3 The Adjustment Work Flow

Our concept of an adjustment work flow loop is motivated by the observation that decision makers that use data mining usually go through quite a number of mining attempts before they get the knowledge they are interested in. While this is partially because the decision makers have additional knowledge that they have to find ways to include into the mining process (usually by taking away data, which is what non-experts in data mining can manipulate and understand the easiest), it is also due to the various parameters and inputs todays data mining algorithms have that need to be adjusted the right way to focus on the right data to create the knowledge a user is interested in. And having several miners (and additional agents like combiners) available makes this even more difficult.

Our adjustment work flow tackles this second cause for repetitive attempts at data mining by integrating knowledge about the miners and how to influence them into the controller in form of knowledge-based strategies. Similar to the supervisor in a teamwork-based search system (see [2]), but able to deal with heterogeneous agents, the controller combines general knowledge with the observations made in the mining run so far, in order to adjust the whole agent-based mining system towards more efficiently performing mining and reducing the need for a human user to provide help. Again, the type of miners and the types of knowledge mined and generated will influence how the adjustment work flow needs to be instantiated, but there are several general knowledge-based strategies that offer guidance in this regard and that we will present in the following.

The first step in an iteration of the adjustment work flow is the selection of miners and parameters for them and the selection of the data sets these miners should work on. While in the first iteration there is not really much to help with this, so that a standard team of miners with standard parameter settings will have to be used and, if the database is too big, standard methods for data selection, like random sampling of the data, the moment performance data for miners is available it is possible to adjust their parameters and to provide focus to the data selection. Parameter adjustment of the miners should aim to improve the whole system's effectiveness and therefore needs information about the effectiveness of the components from previous iterations of the outer loop. This information includes the number and quality of newly discovered rules. Using this information, the controller can

- inform miners with too few rules (in the last few iterations) to produce more rules (for many miners, this means lowering thresholds) and
- inform miners that produced only sub-par rules to improve rule quality (by raising thresholds, for example).

While data mining is aiming at large data sets, the reality is that most mining algorithms run into serious efficiency problems if the data set to mine gets too big. Consequently, some selection of the data to mine needs to take place and while theoretically this could be integrated into the cooperative work flow, conceptually it fits better into the adjustment work flow and in this flow we also can make use of more information. There are several possible data selection strategies, the easiest is the already mentioned random sampling. Random sampling can be slightly improved by biasing the sampling, for example by requiring the same distribution of a particular attribute in the sample as in the whole data set. Another data selection strategy aims at rule coverage by not selecting entries that are already covered by rules in the result sets from previous iterations.

For an agent-based mining system, data selection based on clustering is a very interesting method. As the name suggests, the basic idea of this method is to run a clustering algorithm (resp. agent) over the data set to create clusters of similar data entries. Then the entries in each cluster can be used as the data set for all miners for one iteration of the outer loop. It should be noted that this data selection method, due to using a rather complex algorithm, naturally itself also has a potential data selection problem. This has to be solved using another data selection strategy, like random sampling, again. And still, the clustering miner might take several iterations of the outer loop to finish. Therefore, in our experiments, we run this miner asynchronously in parallel to the adjustment work flow, using the results when they become available and before that we use random sampling or rule coverage for data selection.

The next step in the outer loop is the inner loop that we presented in the last subsection. After each iteration of the inner cooperative loop, the miners and the combiner also provide the controller with information about this last iteration, especially number and quality of the rules found by the miners and execution time differences between the miners. If a miner is not pulling its weight over several iterations, the controller can stop the inner loop prematurely and then also finish the outer loop iteration, so that it can perform adjustments in the then immediately starting next outer loop iteration.

The final step of the adjustment work flow is the results evaluation, which includes both looking at the produced rules, but also the performance of the miners. We already presented these tasks, since they also represent the first steps of the next iteration of the adjustment work flow.

4 Case Studies with CoLe^2

In this section, we present an instantiation of the CoLe^2 method to mine medical databases. One interest of doctors is the prediction of a diagnosis (disease) based on attributes other than the diagnosis itself in order to perform the appropriate tests early and then treat the disease earlier (with usually less cost and higher chances for a cure). This will be the application of our mining system. We will first provide more detail about this application, then describe the CoLe^2 instantiation and finally report on our experimental evaluation of the system.

4.1 Mining Medical Databases

We evaluated our CoLe^2 example system with two medical data sets, one about diabetes and one about kidney disease. The diabetes set was already used to evaluate CoLe (see [4]) and contains data from between 1995 and 2000 about 3150 people with a diabetes diagnosis in 2001 and 6300 control cases. Together, these people produced 436776 transactions with the Calgary health care system, each of which, among others, have a date and at least 1, but up to 3 diagnoses in the international ICD-9 code (see [7]).

The kidney disease data set has been collected by the Alberta Kidney Disease Network and contains data about all Alberta patients that had a chronic kidney disease related test in the time interval July 2001 to March 2006. Our medical partners were interested in finding rules predicting death of a patient and we had 11775 such patients among the 110452 in the database. The data base contains 6476150 transactions for these patients, again, among others, containing a date and between 1 and 3 diagnoses in ICD-9.

Since both data sets come from the Alberta health care system, they have a similar structure. We mined both data sets with a specific goal (target) in mind, namely finding rules predicting diabetes, respectively rules predicting death. This means that the consequences of the mined rules for each data set are fixed. Similar to [4], we need some preprocessing on the data, since ICD-9 is very specific and as a result there are too many only slightly different basic diagnoses. Instead of those, we used the disease groups that ICD-9 associates with the different diagnoses.

4.2 Instantiating CoLe^2

For the mining task described in the previous subsection, we instantiated the CoLe^2 method as follows. We use five miners and two combiners in addition to the controller. The four miners for the inner loop are a PART conjunctive miner and an Apriori miner from the Weka mining library (see [5]), a sequence miner SeqGA from [4] and a relevance-factor-based descriptive miner. Due to lack of space, in the following we concentrate on the modifications we made to these miners for CoLe^2 and otherwise describe only the basic ideas of these miners.

The PART miner is based on the well-known C4.5 mining approach (see [13]) and builds partial decision trees, which are transformed into conjunctive rules by simply collecting all of the predicates on a path in the tree from the root to a leaf. We modified the PART implementation from Weka by using the predicates in hints from other miners and combiners to do attribute selection (selecting them and adding also some randomly selected attributes). We also integrated an additional pruning of the created rules before they are forwarded to the combiner, by checking for each predicate in a rule, if its elimination does not influence the rule fitness. If a predicate is found irrelevant in this way, it is eliminated. This is in addition to the usual rule pruning in a PART miner. If the controller wants the PART miner to spend less time, the miner implements this by having less pruning rounds (and vice versa).

The Apriori association miner (see [1]) also creates rules that have conjunctions of predicates as conditions. The Apriori miner builds rules from the bottom up by determining item (predicate) combinations that appear frequently in the transactions. Starting with single items that appear more often than a threshold, items (combinations) are combined and this is iterated with combinations that are above the threshold. Finally, combinations with the target predicate are split into a rule where the target predicate is the consequence and all other predicates form the condition. We modified this general method for CoLe^2 by, again, using hints from other miners for attribute selection, thus limiting the potentially huge number of item combinations. The Apriori miner can adjust its runtime by raising or lowering the threshold mentioned above.

The SeqGA miner uses a genetic algorithm (GA) to create sequence rules, i.e. rules with a condition that is a (temporal) sequence of events. The individuals of the GA are exactly such sequences, with events being a particular diagnosis and the sequence indicating that each element in it happened before the next element. As usual, the GA works on sets of such individuals and each individual in the set is evaluated with the rule fitness. We use genetic operators to create new individuals that replace the least fit individuals. Crossover just chooses two individuals, cuts them at a randomly chosen point (in the sequence) and concatenates the head part of one individual with the tail of the other. Mutation randomly either adds an event to an individual, deletes an event or substitutes an event with a random new one (or one from the received hints). In addition to sending the rules with the best fitness (above a threshold) to the combiner, the SeqGA miner also sends the predicates in these rules to the other miners as hints. The SeqGA miner can react to adjustment requests by the controller by adjusting the number of generations and, again, by adjusting the fitness threshold.

All these 3 miners evaluate rules using the same rule fitness. As in [4], this fitness fit of a rule r is calculated as

$$fit(r) = \left(\frac{tp}{tp + fp}\right)^x \times \frac{\log(tp)}{\log(positive)}$$

with tp being the number of data entries correctly classified by r, fp the number of incorrectly classified entries and *positive* being the number of entries for which the rule condition is true. x is a real number parameter that was set to 1.5 in all our experiments (following [4]).

One of the strengths of multi-agent-based data mining is the cooperation of miners, which also allows for the use of miners that alone would never be very good or even acceptable. Our relevance-factor-based miner is such a miner that essentially acts as "material generator" for the other miners. It first generates a set of predicates from attributes in the data set given to it. Nominal attributes create predicates that test equality with each of the possible values. Predicates for numerical attributes are collected from all the communicated hints from previous iterations of the cooperative work flow (in previous outer loop iterations). Then the miner calculates the relevance factor for each predicate and all predicates that are relevant (according to the factor) are retained. For each of those predicates p, a rule is constructed either of the form $p \Rightarrow$ *target concept* (if the predicate had a positive relevance) or $not(p) \Rightarrow$ *target concept* (else). These rules are sent to the combiner and the predicates are sent as hints to the other miners. The relevance-factor-based miner is only used in the first iteration of the cooperative loop within an adjustment work flow iteration.

As stated in the last section, we also use an X-means clustering miner (see [12]) as kind of an assistant to the controller to help with the selection of data sets for the miners. In contrast to the well-known k-means clustering, X-means does not produce a pre-determined number of clusters, but determines the best number of clusters between given lower and upper bounds using the so-called Schwarz criterion. In our CoLe^2-based system, we use random sampling to produce a data sub-set for the X-means miner, perform a full clustering, and use the cluster with the highest score of the Schwarz criterion as the data set for the next execution of the cooperative loop. After that, we determine all entries that are correctly predicted by the resulting rules, remove them from the remaining clusters created by the X-means miner, update the cluster centers and reassign the remaining data entries to the clusters and repeat the cooperative loop with the new best cluster. If there are no clusters left, we repeat the general X-means clustering with a new data sub-set of the whole database.

With the description of the miners, we have already presented most of the instantiation of the cooperative work flow. What remains is the instantiation of the combiner. The concrete combiner we use follows rather closely what we described in Section 3.2. It reevaluates all rules communicated by the miners and puts those above a certain threshold into the candidate set. It then performs *direct combination* and *cross combination* to create more candidates. Direct combination simply combines the conditions of two rules with an *and* operator. Cross combination computes the number of occurences of the predicates in the rules from the miners. It then goes through all of these rules and randomly chooses a number of predicates to add and the insertion points of these predicates into the rule. For each of these points, a predicate is chosen with predicates with higher occurrence counts having a higher chance to be selected. The operator to connect the predicate to the condition is then randomly chosen and the new rule is evaluated. If it is better than the parent rule, this process is repeated until no improvement is achieved and the rule before that last attempt is added to the candidate set. Then we prune the candidate set, which includes simplifying the conditions of a rule by eliminating redundant occurrences of a predicate, throwing away predicates for which a stronger predicate is present in the condition and throwing away duplicates of a rule and rules similar to a rule with a smaller fitness. Finally, all remaining rules below a given threshold are also eliminated and the remaining rules are the result of this iteration.

Rule similarity is based on the similarity of the predictions the rules make, which essentially creates a similarity on conditions. In theory, we should compare the performance of two rules on each of the data entries, but this becomes computationally much too expensive. Instead, we assign the data entries into n bins (in our experiments we used $n = 256$) and compare two rules based on the number of correct classifications (match score) for the bins. If the difference between two rules with this regard is below a threshold, the rules are considered similar.

We have also already presented many of the pieces that form the adjustment work flow, like the data selection and how miners can adjust their parameters. As already stated in the last section, the controller performs the data selection for the first round using random sampling. It uses one agent from each of the miner types (that produce rules) in each of the iterations and tries to keep their output over all iterations balanced by advising individual miners what parameter adjustments they should make. The X-means clusterer runs in parallel to the cooperative loop, so that its results are only available in the following iteration of the inner loop. The controller collects the result sets from the combiner over all of the loops. After a given number of iterations of the adjustment loop, the controller lets a variant of the combiner perform a final simplification of the complete result set (as already described for the combiner).

4.3 Experimental Results

In this subsection, we present some experimental evaluations of our CoLe^2-based system. As stated in the introduction, our main claim is that CoLe^2 improves the efficiency of an agent-based mining system, so that our experiments focus on this aspect. Naturally, efficiency improvement should not be achieved by reducing quality, so that we also look at the quality of the rules (expressed using the rule fitness). Since some of the miners and some of the knowledge-based strategies used in the loops involve random decisions and in an agent-based system timing of messages can happen in slightly different ways, we did not perform only single mining runs, but always look at the results of three runs.

Table 1. Runtimes: with and without outer loop

Data Set	Experiment No.	Running Time (sec) without outer loop	Running Time (sec) with outer loop	Time Ratio
diabetes	1	1715.54	1234.57	71.96%
	2	2719.63	1329.73	48.89%
	3	1795.72	1535.19	85.49%
	Average	2076.96	1366.49	65.79%
kidney	1	43263.95	7242.02	16.74%
	2	17714.54	5403.13	30.50%
	3	43597.24	6590.54	15.12%
	Average	34858.58	6411.90	18.39%

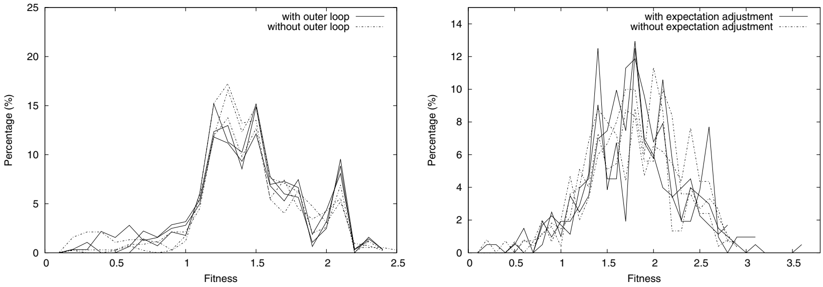


Fig. 2. Fitness histograms: left: comparison with vs without outer loop; right: comparison with vs without parameter adjustment

Effect of the Outer Loop: Obviously, the outer adjustment loop is the major contribution of this paper, so that our first experimental series evaluates the effect of this loop against a configuration of the system that simulates just the CoLe method. More precisely, the CoLe mode still uses random sampling to provide small enough working data sets for the miners and performs a loop around the inner loop that creates different working data sets, but the other features of the outer loop are not used. Both systems used otherwise the same parameter settings and were run until more than 95% of the data entries were used in at least one run of the inner loop.

As Table 1 shows, the outer loop clearly reduces the run time for the system and importantly does so substantially for the larger data set. For this data set, even the worst mining run with outer loop is 3 times faster than the run not using the outer loop (but starting with the same initial working data set).

Figure 2 (left) shows a fitness histogram of the quality of the mined rules for the 6 runs on the larger kidney set. While it could be argued that the rule quality with the outer loop is slightly better (higher peaks after a fitness value of 2), using the outer loop definitely does not result in loosing quality, so that our primary goal with CoLe^2 is clearly fulfilled.

Parameter Adjustment: One of the knowledge-based strategies employed by the controller is adjusting the parameters of the miners to have them contribute

Table 2. Run time (seconds): with and without parameter adjustment

Dataset	Test No.	Without Parameter Adjustment	With Parameter Adjustment	Time Ratio
diabetes	1	1722.55	1223.51	71.03%
	2	1667.44	907.87	54.45%
	3	2473.14	1014.09	41.00%
kidney	1	5122.68	3943.20	76.98%
	2	9710.18	6156.67	63.40%
	3	8805.31	4655.57	52.87%

evenly to the mining results and to have them running in the inner loop iterations without large idle times. To test this strategy, we compare in Table 2 runs of the system using all knowledge-based strategies with runs where no parameter adjustments of the miners take place.

As can be seen, using all strategies always results in faster runs and for the smaller data set it can be up to only half of the time when using this strategy versus not using it. And, as Figure 2 (right) shows, this, again, is achieved without loss of quality.

X-means Data Selection: To look into the improvement the X-means miner represents, we measured the quality of rules generated by the miners and the combiner of essentially one iteration of the outer loop and compared the average rule quality using X-means directed data selection versus random sampling. We did this for the larger kidney data set, since this data set definitely needs data selection.

As Table 3 shows, while the results with regards to the miners are mixed, when combined by the combiner, the average rule quality is clearly enhanced, showing the usefulness of this knowledge-based strategy.

Table 3. Average rule fitness of one iteration: X-means vs random sampling data selection

Test No.	Miner or Combiner	Data Selection Method Random Sampling	X-means
1	PART miner	0.324	0.242
	SeqGA miner	0.851	0.867
	Relevance factor miner	0.553	0.576
	Combiner	**1.275**	**1.464**
2	PART miner	0.228	0.600
	SeqGA miner	0.985	0.625
	Relevance factor miner	0.558	0.560
	Combiner	**1.374**	**1.790**
3	PART miner	0.309	0.310
	SeqGA miner	1.043	1.107
	Relevance factor miner	0.558	0.560
	Combiner	**1.606**	**1.858**

Other Strategies: Due to lack of space, we cannot provide detailed results for the other strategies. But our experiments showed that having an asynchronous workflow resulted in improvements between 30 and 70 percent. Although [4] already showed the usefulness of the created hints, we performed appropriate experiments, again, and the hints produce an efficiency improvement between 20 and 50 percent and, more importantly, substantial improvements in the quality of the produced rules (compared to not using hints in the miners).

5 Related Work

Agent-based data mining is of interest because of the ability to deal with distributed databases. The data mining work has to be done distributively either due to privacy concerns, or due to the large total data amount which prohibits a centralized database. An example of a method aimed at this reason is the JAM framework described in [14], where multiple data sites exist and each of them employs a classification agent to perform data mining and a meta learning agent to exchange and combine the classification models. Another example is the frame presented in [9], where the focus is to establish distributed cooperative data mining in a competitive and privacy restricted environment, with each data mining agent working on their own private data set using Naive Bayes classifiers. In these situations, it is natural that the agents in the data mining system cooperate towards the same data mining goal. In comparison to typical work in this area, we are not only interested in mining large amounts of data and enabling cooperation, but also particularly interested in producing hybrid knowledge that goes beyond what each of the individual mining agents can produce.

Most approaches in the literature that have some centralized control concept like our controller have it as part of the inner work flow and do not use heterogeneous agents (which naturally makes the control task much easier). In [3], a manager agent divides the data into disjunct working sets for analyzer agents that, after having done their own mining, vote on whether to keep the results of the other agents. [8] is another example for a simple mining loop. We are not aware of any other work using two loops and the benefits of the outer loop, like mining cooperatively a certain part of the database or having a specialized assistant miner to prepare for that.

6 Conclusion and Future Work

We presented CoLe^2, a concept for agent-based mining around two work flow loops, an inner cooperative loop and an outer adjustment loop. The outer loop allows for various knowledge-based strategies that allow a controller agent to adapt and focus the whole team of agents on the particular database and mining task. Our experimental evaluation showed that the outer loop substantially improves the efficiency of the mining system while still producing results of comparable quality to without using the outer loop.

The outer loop offers several more possibilities for using knowledge-based strategies that we intend to look into in the future. While in this paper we concentrated on generally applicable strategies, naturally for databases for which the mining needs to be periodically repeated it makes sense to integrate application specific knowledge into the mining. The outer loop also offers the possibility to integrate human judgement in form of advice to the controller.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Fayyad, U.M. (ed.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI Press (1996)
2. Denzinger, J., Kronenburg, M.: Planning for distributed theorem proving: The teamwork approach. In: Görz, G., Hölldobler, S. (eds.) KI 1996. LNCS (LNAI), vol. 1137, pp. 43–56. Springer, Heidelberg (1996)
3. de Paula, A.C.M.P., Ávila, B.C., Scalabrin, E.E., Enembreck, F.: Using distributed data mining and distributed artificial intelligence for knowledge integration. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS (LNAI), vol. 4676, pp. 89–103. Springer, Heidelberg (2007)
4. Gao, J., Denzinger, J., James, R.C.: A cooperative multi-agent data mining model and its application to medical data on diabetes. In: Gorodetsky, V., Liu, J., Skormin, V.A. (eds.) AIS-ADM 2005. LNCS (LNAI), vol. 3505, pp. 93–107. Springer, Heidelberg (2005)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations Newsletter 11, 10–18 (2009)
6. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann (2011)
7. Karaffa, M.C. (ed.): International Classification of Diseases, 9th Revision, Clinical Modification, 4th edn. Practice Management Information Corp., Los Angeles (1992)
8. Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, distributed data mining using an agent based architecture. In: Proc. 3rd KDD, pp. 211–214 (1997)
9. Lisý, V., Jakob, M., Benda, P., Urban, Š., Pěchouček, M.: Towards cooperative predictive data mining in competitive environments. In: Cao, L., Gorodetsky, V., Liu, J., Weiss, G., Yu, P.S. (eds.) ADMI 2009. LNCS, vol. 5680, pp. 95–108. Springer, Heidelberg (2009)
10. Liu, H., Lu, H., Yao, J.: Toward multidatabase mining: Identifying relevant databases. IEEE Transactions on Knowledge and Data Engineering 13(4), 541–553 (2001)
11. Moemeng, C., Gorodetsky, V., Zuo, Z., Yang, Y., Zhang, C.: Agent-based distributed data mining: A survey. In: Cao, L. (ed.) Data Mining and Multi-agent Integration, pp. 47–58. Springer (2009)
12. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Proc. 17th ML, pp. 727–734 (2000)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann (1993)
14. Stolfo, S.J., Prodromidis, A.L., Tselepis, S., Lee, W., Fan, D.W., Chan, P.K.: JAM: Java agents for meta-learning over distributed databases. In: Proc. 3rd KDD, pp. 74–81 (1997)

Sign Language Recognition with Support Vector Machines and Hidden Conditional Random Fields: Going from Fingerspelling to Natural Articulated Words

César Roberto de Souza and Ednaldo Brigante Pizzolato

Universidade Federal de São Carlos, São Carlos, Brasil
{cesar.souza,ednaldo}@dc.ufscar.br

Abstract. This paper describes the authors' experiments with Support Vector Machines and Hidden Conditional Random Fields on the classification of freely articulated sign words drawn from the Brazilian Sign Language (Libras). While our previous works focused specifically on fingerspelling recognition on tightly controlled environment conditions, in this work we perform the classification of natural signed words in an unconstrained background without the aid of gloves or wearable tracking devices. We show how our choice of feature vector, extracted from depth information and based on linguistic investigations, is rather effective for this task. Again we provide comparison results against Artificial Neural Networks and Hidden Markov Models, reporting statistically significant results favoring our choice of classifiers; and we validate our findings using the chance-corrected Cohen's Kappa statistic for contingency tables.

Keywords: Gesture Recognition, Sign Languages, Libras, Support Vector Machines, Hidden Conditional Random Fields, Neural Networks, Hidden Markov Models, Discriminative Models.

1 Introduction

Humans are social creatures; and therefore depend heavily on communication to perform daily tasks and achieve their goals. But in the event one of the main communication channels have been damaged or have been deemed unavailable, humans will inevitably find a way to restore this communication. In the case of deafness or speech impairment, the communication is often restored through Sign Languages.

However, one of the most immediate problems of the Sign Languages is that very few people outside the deaf community are actually able to speak them. Bridging this gap with an autonomous translator seems like a logical step towards a more inclusive society. Hence, this work aims to walk the first steps in this direction.

This paper enlists comparative results for the automatic recognition of a finite number of words drawn from the Brazilian Sign Language, heretofore Libras. Our system works in an unconstrained environment without the aid of gloves, markers or controlled lighting. At the heart of our system lies a two-layer architecture based on the automatic learning of discriminative models to work on different stages of the

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 84–98, 2013.

recognition process, as suggested in [1]. However, unlike the aforementioned work, our layers are composed of Support Vector Machines (SVMs) and Hidden Conditional Random Fields (HCRFs), as in [2]. Besides, our system works directly with naturally articulated words rather than fingerspelled ones. These words may also involve more than just one hand and even the user's facial expressions. We intend to present our system to the reader, along with comparison results against other approaches and also discuss the overall linguistic foundations behind our recognition strategy.

This paper is organized as follows. After this introduction, Section 2 gives a list of related works, raising some points of interest and discussions. Section 3 gives an overview of Sign Languages, giving special consideration to Libras. Section 4 presents the methods, models and tools used in this work. In Section 5 we detail our approach to Sign Language recognition. Section 6 then lists our experiments, detailing the dataset and strategies we have used. Section 7 presents and discusses our findings, while Section 8 concludes this work.

2 Related Works

Sign Language recognition is closely related to gesture recognition. Following a comprehensive survey on this topic given in [3], one can say gesture recognition methods have been traditionally divided into two main categories: Device-based [4] and vision-based [5, 6, 7, 8]. Device-based approaches often constrain the users to wear a tracking device, such as a tracking glove, resulting in less natural interaction. Vision based approaches, on the other hand, free the user from wearing potentially movement-limiting and otherwise expensive devices. In this paper, we will deal only with vision-based approaches.

Gestures can also be either static or dynamic. Static gestures, often called poses, are still configurations performed by the user, passive to be registered in a single still image. Dynamic gestures, in turn, vary on time, and have to be captured as a sequence of still images, such as image streams. Often, gestures have both elements, such as in the case of sign languages [3]. In this paper we will be covering both gesture types.

Several works have also been published aiming the recognition of specific Sign Languages. As examples we have systems targeting the American Sign Language (ASL) [9], the German Sign Language (*Deutsche Gebärdensprache*, DGS) [6], the British Sign Language (BSL) [10] and the Australian Sign Language (Auslan) [11]. We also have works focusing the same language as ours, such as the works done by Pizzolato *et al.* [1] and colleagues [2], who addressed only fingerspelling; and the works by Dias *et al.* [7], who focused specifically on the movement aspects of Libras.

The work by Dias *et al.* provided a convenient mathematical formulation of the movement recognition problem, presenting a solution using Self-Organizing Maps (SOM) networks and Vector Quantization. Interestingly enough, Carneiro *et al.* [12] also used the SOM model as a preprocessing step to classify signs from the Libras manual alphabet, both reporting high classification rates.

Moreover, other papers have already explored HCRFs [13] and other variants for gesture recognition. Morency *et al.* used Latent Dynamic-CRFs (LD-CRFs) [14] to perform gesture recognition in continuous image streams, with excellent results.

Elmezain *et al.* [8] also studied CRFs, HCRFs and LD-CRFs in the recognition of alphabet characters and numbers drawn in mid-air using hand trajectories, obtaining 91.52%, 95.28% and 98.05% for each model, respectively.

We consider our work to be more closely related to the work done by [10], in which the authors considered a linguistic model for signed words. Their structural approach attempted to decompose the sign into *visemes* in the same way a spoken word can be decomposed into *phonemes*. The next sections should make it clear how we took a similar approach by decomposing the Libras sign into its appropriate component units.

3 Sign Languages and the Brazilian Sign Language

Languages based on visual signs have arisen in the same manner as all spoken languages. Contrary to popular belief, those languages are not mimics. Most often they are also not a sign version of the spoken languages, such as English or Portuguese. They are fully qualified languages, with their own grammar, lexicons and semantic rules to be obeyed.

Furthermore, Sign Languages are not universal – and thus signers from one country or community should not be expected to be able to talk with signers from other communities unless they know and agree to sign in the same language. Hence Brazil's official Sign Language is the Libras, recognized as such since early 2002.

The recognition of the Libras as the official language for the country was perceived as a victory for the deaf communities in Brazil. This eventually led to its regulation in 2005, making Libras classes mandatory in teacher formation curses – such as for obtaining a Pedagogy degree in Higher Education. The deaf also acquired the right to the interpreter in court and in education, further increasing the importance of the interpreter in the Brazilian society. The availability of tools for the automatic recognition of Sign Language could thus be of major interest to help those professionals.

The Libras also have been studied under linguistics grounds. One of the first and still most comprehensive descriptions of the structural organization of the Libras was given by Brito [15]. Her effort at characterizing the elements of the Libras sign has been one of the major guides in designing this system. Brito had identified 46 fundamental Hand Configurations (HC) in the Libras. She also identified other parameters such as the Articulation Points (AP), the Movement types (M), the Hand Palm Orientation (HO), the Contact Region (CR) and the Non-Manual Components (NM) used in the language. If one could identify all possible elements in the aforementioned parameter sets, then a sign could possibly be denoted as a tuple

$$S = \langle HC, AP, M, HO, CR, NM \rangle.$$

One must also give special consideration to the NM set. Non-manual information often plays a crucial role in determining the true meaning of a signal in Libras. Facial information, for instance, can be used to resolve ambiguities, further qualify the sign being performed and even completely negate or change the meaning of a sign. This information thus cannot be simply ignored by an automatic recognition system.

4 Methods and Tools

4.1 Artificial Neural Networks

As the name implies, at their creation Artificial Neural Networks (ANNs) had a strong biologic inspiration. However, despite their biological origins, ANNs can be regarded as simple functions $f\colon \mathbb{R}^n \rightarrow \mathbb{Y}$ mapping a given input $\boldsymbol{x} \in \mathbb{R}^n$ to a corresponding output $\boldsymbol{y} \in \mathbb{Y}$. The output vectors $\boldsymbol{y} = \langle y_1, \dots, y_m \rangle$ are also restricted to a specific subset of $\mathbb{R}^n$. Each $y_i \in \boldsymbol{y}$ is restricted to a particular range according to the choice of activation function for the output neurons. In the case of sigmoid activation functions, this range is $[0; 1]$; in case of bipolar sigmoid functions, it is $[-1; +1]$.

Since ANNs can be seen as standard mathematical functions, the learning problem can also be cast as a standard optimization problem, in which one would like to minimize a divergence, in some sense, between the network outputs $\hat{\boldsymbol{y}}$ and the desired answers $\boldsymbol{y}$. One possible way to achieve it is through the minimization of the error gradient; and a promising method for this is given by the Resilient Back-propagation algorithm (Rprop) [16, 17].

The Rprop algorithm is one of the fastest methods for gradient learning restricted solely to first-order information. Its basic operational principle is to eliminate the (possibly bad) influence of the gradient magnitude in the optimization step. Unlike other gradient based methods, such as Gradient Descent, in which the step size is always proportional to the gradient vector, Rprop takes into account only the direction of the gradient, making it a local adaptive learning algorithm. Because Rprop relies only in first-order information, it is not required to compute (and hence store) the Hessian matrix of second derivatives for the learning problem, making it especially suitable for high dimensional problems.

One of the biggest challenges in learning ANNs is the presence of multiple local minima and the relatively large number of hyper parameters which have to be carefully adjusted in order to ensure a good generalization. In despite of this, they have been finding much renewed interested from the machine learning and artificial intelligence community thanks to recent advances in learning algorithms for deep layer architectures, in a new approach which is now referred as the deep learning paradigm [18].

4.2 Support Vector Machines

In face of the problems often found with other learning models, such as the presence of multiple local minima and the curse of dimensionality, the SVM had been specifically conceived to avoid such issues. Although some may argue that the SVM does not have any edge over the curse of dimensionality [19], their practical importance cannot be diminished. These models have shown great performance in many real-world problems [5, 20, 19], including problems of high dimensionality [21] and of large-scale [22]. In the linear case, the SVM decision is given by a simple hyperplane decision function on the form

$$h(\boldsymbol{x}) = sgn(\boldsymbol{\theta} \cdot \boldsymbol{x} + b) \overset{\omega_1}{\underset{\omega_2}{\lessgtr}} 0 \tag{1}$$

in which we decide for class ω_1 if a point $\boldsymbol{x}$ lies on one side of the hyperplane defined by the parameter vector $\boldsymbol{\theta}$ and threshold b; or class ω_2 if it lies on the other. The SVM decision function is usually given in its dual form, in terms of Lagrange multipliers $\boldsymbol{\alpha}$, selected support vectors $\boldsymbol{z}$ and output labels y as

$$h(\boldsymbol{x}) = sgn\left(\sum_{i \in SV} y_i \boldsymbol{\alpha}_i \boldsymbol{z}_i \cdot \boldsymbol{x} + b\right) \overset{\omega_1}{\underset{\omega_2}{\lessgtr}} 0\,. \tag{2}$$

In order to generalize this model to non-linear decision surfaces, one can introduce a nonlinear transformation $\varphi(\cdot)\colon \mathbb{R}^n \rightarrow \mathcal{F}$ such that, when applied to the input vectors $\boldsymbol{x_i} \in \mathbb{R}^n$, creates a projection in a high-dimensionality feature space $\mathcal{F}$. Using the kernel trick, one can replace inner products in eq. (3) with a Mercer's kernel of the form $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{z}) \rangle$. Since φ does not have to be explicitly computed, the feature space $\mathcal{F}$ can have an arbitrarily high dimensionality.

The learning procedure for such model can be done using an approximate version of the Structural Risk Minimization principle [23]. In this work, all training has been performed using Platt's Sequential Minimal Optimization (SMO) algorithm [20, 24].

Multiclass Classification Approaches

An immediate problem arising from the SVM's original hyperplane formulation is that it is not very obvious how to make the model applicable to more than two classes. Several approaches have been proposed to overcome this limitation, two of them being known as the 1-vs-1 and 1-vs-all strategies for multiple class classification.

For a decision problem over c classes, 1-vs-all requires the creation of c classifiers, each trained to distinguish one class from the others. The decision then is taken in a winner-takes-all approach. However there is no clear indication this approach results in an optimum decision. In the 1-vs-1 strategy, the problem is divided into $c(c-1)/2$ sub-problems considering only two classes at a time. At the decision phase, each machine casts a vote for one of the classes and the label with highest number of votes wins. This leaves the problem of evaluating an increased number of machines every time a new instance is classified – which could easily become troublesome or prohibitive in time sensitive applications.

The Decision Directed Acyclic Graph (DDAG), first proposed in [20], provides the fast evaluation times of the 1-vs-all strategy while at the same time offering strong generalization bounds on the generalization error. The DDAG also keeps the original hyperplane decision formulation by sequentially cutting the decision space until a decision is found. For a decision problem over c classes, only $(c-1)$ machines need be evaluated [20]. The performance of this approach improves significantly when using linear machines since each SVM evaluation is reduced to a single vector multiplication.

4.3 Hidden Markov Models

Hidden Markov Models (HMMs) attempt to model the joint probability distribution of a sequence observations $\boldsymbol{x}$ and their relationship with time through a sequence of hidden states $\boldsymbol{y}$. A HMM is described by a tuple $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ in which $\boldsymbol{A}$ denotes a matrix of possible state transition probabilities, $\boldsymbol{B}$ is a vector of probability distributions governing the observations and $\boldsymbol{\pi}$ is a vector of initial states probabilities. In the literature, HMMs are often described alongside three associated canonical problems: evaluation, learning and decoding. Although we will not be discussing those in detail, a very comprehensive explanation is due to Rabiner [25].

Exploring the fact that an HMM is able to provide the likelihood for a given sequence $\boldsymbol{x}$, it is possible to create a classifier by creating a model λ_i for each sequence label $\omega_i \in \Omega$. Treating each model λ_i as a density model conditioned to an associated class label ω_i, one can apply the Bayes' rule to obtain the a posteriori probability and then decide for the class with *maximum a posteriori.*

4.4 Hidden Conditional Random Fields

The HCRF can be seen as a latent-variable extension of the Conditional Random Field (CRF), first proposed by [26]. The HCRF attempts to model $p(\omega|\boldsymbol{x})$, the probability of a class label given a sequence, without incorporating an specific model for $p(\boldsymbol{x})$. This is in direct contrast with their generative counterpart given by generative classifiers based on sets of hidden Markov models (HMMs), which model $p(\boldsymbol{x}|\omega)$ individually for each class label and attempt to convert those to the posterior probabilities $p(\omega|\boldsymbol{x})$ either using Maximum Likelihood or Maximum a Posteriori estimates with the aid of Bayes' rule.

A general and comprehensive definition of a CRF can be found in [27], in which the authors define a CRF based on the partitioning of a factor graph. As such, consider a factor graph G partitioned in a set of clique templates $\mathcal{C} = \{C_1, C_2, \dots, C_P\}$. Each clique template C_p should specify a set of sufficient statistics $\{f_{pk}(\boldsymbol{x_p}, \boldsymbol{y_p})\}$ and parameters $\boldsymbol{\theta_p} \in \Re^{K(p)}$, in which $\boldsymbol{x}$ is the sequence of observations and $\boldsymbol{y}$ the sequence of hidden states associated with the observations $\boldsymbol{x}$. Then, a general model for a CRF can then be written as

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\boldsymbol{x_c}, \boldsymbol{y_c}; \theta_p) \tag{3}$$

in which $\Psi_c(\boldsymbol{x_c}, \boldsymbol{y_c}; \theta_p) = exp\{\sum_{k=1}^{K(p)} \theta_{pk} f_{pk}(\boldsymbol{x_c}, \boldsymbol{y_c})\}$ and $Z(\boldsymbol{x})$ is the partition function used to keep results as probabilities.

Unlike HMMs, CRFs assume the label sequence $\boldsymbol{y}$ to be known during training. One possible solution to this problem is to handle $\boldsymbol{y}$ as latent variables. By adding a variable ω to designate class labels, and setting $\boldsymbol{y}$ to be hidden, one arrives at the HCRF formulation given by

$$p(\omega|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \sum_{\boldsymbol{y}} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c, \omega_c;\ \theta_p) \tag{4}$$

which can be computed by the same exact algorithms used to compute $Z(\boldsymbol{x})$ in the CRF case. Given a training dataset of N input observation sequences $\boldsymbol{x}^{(i)}$ and corresponding class labels $\omega^{(i)}$, the model can be estimated by taking the gradient of the log-likelihood function

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} log\, Z(\omega^{(i)}, \boldsymbol{x}^{(i)}) - \sum_{i=1}^{N} log\, Z(\boldsymbol{x}^{(i)}) \tag{5}$$

and using any off-the-shelf optimizer such as L-BFGS or Conjugate Gradient to estimate $\boldsymbol{\theta}$. In this work we will be using the Resilient Backpropagation (Rprop) algorithm, first proposed to be used in ANNs, but also applicable to arbitrary optimization problems. The work by Mahajan *et al.* has shown Rprop to be one of the best algorithms for estimating HCRF models [28].

5 A Two-Layered Approach for Sign Recognition

Our approach for recognizing sign words from Sign Languages is clearly inspired by the field of speech recognition, which has enjoyed an extensive and ever-increasing literature over the years. In the same way speech recognition methods often build language models over phoneme classifiers, here we have a first layer, aimed to detect static gestures such as hand shapes from Brito's set of hand configurations; and a second layer, aimed to classify sequences of static gestures plus temporal, trajectory and facial information into words from a finite lexicon.

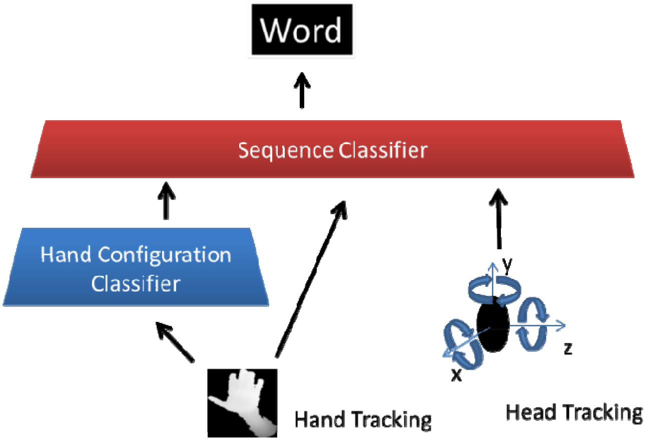


Fig. 1. The two layered architecture used in this work

Considering Brito's work, along this paper we will consider the feature vector

$$\boldsymbol{x_t} = \langle h_c, h_{rx}, h_{ry}, h_\theta, f_\theta \rangle$$

$$\begin{array}{lc} h_c \in \mathbb{N}, & 1 \leq h_c \leq 46 \\ h_x, h_y \in \mathbb{R} & -1 \leq h_x, h_y \leq 1 \\ h_\theta, f_\theta \in \mathbb{R} & -\pi \leq h_\theta, f_\theta \leq \pi \end{array} \tag{6}$$

in which h_c denotes the hand configuration detected by the hand configuration classifier; h_{rx} and h_{ry} are the relative positions of the hand compared to the center of the head; and h_θ and f_θ are the angular orientation of the hand and face, respectively. All relative positions are normalized to the unit interval, and angular information is given in radians. For clarity, we will refer to the set of 46 possible configurations as $\mathbb{H}$. To maintain consistency with the following sections, the feature vector will be named $\boldsymbol{x_t}$, and the individual element at the i-th position will be indicated by $\boldsymbol{x_{t_i}}$.

5.1 Static Gesture Recognition Layer

In order to estimate the location of the hands and the face of the user in a stream of continuous images, we use combined information gathered through depth and intensity sensors. To estimate the position of the face and hands, we combine the standard Haar object detection algorithm of Viola and Jones [29] and the Camshift [30] object tracker (with a few modifications for increased stability and robustness) with a Dynamic Virtual Wall Algorithm [31] for depth image segmentation.

Although we will not be discussing the segmentation algorithm in detail, our approach is centered on quickly detecting a tracking failure and quickly recovering from this error. This technique is able to work even when facing noisy and uncontrolled environments. As we shall see shortly, since the subsequent processing layers are able to cope with isolated frame errors, this approach works very well.

Continuing the processing flow, after the hands have been located, a bank of SVMs disposed in a Large-Margin DDAG is used to classify the hand image into one of the possible 46 hand configurations from $\mathbb{H}$. The initial experiments and results shown in [2] have been proven particularly useful to adequately learn the models used in this layer, particularly due the hyperparameter heuristics we had explored earlier.

5.2 Dynamic Gesture Recognition Layer

Our second processing layer takes the output of the first layer, combined with trajectory, spatial and facial information and creates the feature vectors shown in (eq. 6). Considering each feature vector as an individual observation $\boldsymbol{x_t}$ belonging to a sequence of observations $\boldsymbol{x}$, the goal of this layer is to estimate the word label ω which is most likely associated with a given $\boldsymbol{x}$.

To create and learn the dynamic gesture models of this layer we considered feature functions of both discrete and continuous nature. We initialize our HCRF models with probabilities taken from corresponding HMMs, which have been crafted to use independent, mixed joint-distributions of both discrete and continuous variables. This independent formulation could be seen as the application of the naïve Bayes assumption to our feature vectors. The emission distributions for the observation sequences could then be expressed in the form

$$p(\boldsymbol{x}_t) = \prod_{i=1}^{5} p_i\left(\boldsymbol{x}_{t_i}\right) \tag{7}$$

in which p_1 is a discrete distribution and $p_{2\ldots5}$ are assumed approximately normal with unknown mean and variance. We note there may be some concerns since a normal distribution is being assumed for variables which are of circular nature. However, the importance of this imprecision can be diminished when we consider that face and hands movements are limited by the joints of the body and, in case of the face, cannot possibly wrap. Choosing a Normal distribution also makes it easier to draw our linear-chain HCRF features as

$$\begin{array}{lll}
f_{\omega'}^{(Label)}(\omega, \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = & \mathbf{1}_{\{\omega=\omega'\}} & \forall\omega' \in \Omega \\
f_{i,j}^{(Tr)}\ (\omega, \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = & \mathbf{1}_{\{\boldsymbol{y}_t=i\}}\mathbf{1}_{\{\boldsymbol{y}_{t-1}=j\}} & \forall i,j \in S_\omega \\
f_{i,o,d}^{(Em)}\ (\omega, \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = & \mathbf{1}_{\{\boldsymbol{y}_t=i\}}\mathbf{1}_{\{(\boldsymbol{x}_t)_d=\boldsymbol{o}\}} & \begin{array}{l}\forall i \in S_\omega \\ \forall o \in \mathbb{H} \\ \boldsymbol{x}_{t_d} \in \mathbb{N}\end{array} \\
f_{i,d}^{(Occ)}\ (\omega, \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = & \mathbf{1}_{\{\boldsymbol{y}_t=i\}} & \begin{array}{l}\forall i \in S_\omega \\ \boldsymbol{x}_{t_d} \in \mathbb{R}\end{array} \\
f_{i,d}^{(M1)}\ (\omega, \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = & \mathbf{1}_{\{\boldsymbol{y}_t=i\}}\left(\boldsymbol{x}_{t_d}\right) & \begin{array}{l}\forall i \in S_\omega \\ \boldsymbol{x}_{t_d} \in \mathbb{R}\end{array} \\
f_{i,d}^{(M2)}\ (\omega, \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = & \mathbf{1}_{\{\boldsymbol{y}_t=i\}}\left(\boldsymbol{x}_{t_d}\right)^2 & \begin{array}{l}\forall i \in S_\omega \\ \boldsymbol{x}_{t_d} \in \mathbb{R}\end{array}
\end{array} \tag{8}$$

in which Ω is the set of all possible class labels in our classification problem, S_ω is the number of states assumed for sequences of class ω. The label features $f_{\omega'}^{(Label)}$ trigger when a sequence belongs to class ω'. Transition features $f_{i,j}^{(Tr)}$ trigger whenever there is a transition from state i to state j. Emission features $f_{i,o,d}^{(Em)}$ trigger when a discrete symbol o occurs in the d-th position of the observation vector while inside state i. Occupancy features $f_{\mathrm{i,d}}^{(Occ)}$ trigger whenever state i is reached, while the first and second moment features $f_{\mathrm{i,d}}^{(M1)}$ and $f_{i,d}^{(M2)}$ perform the sum and sum of squares of the observation features at positions d when the state is i.

Using this set of feature functions, an HMM classifier created after each class label ω with prior probabilities α_ω , transition matrices $\boldsymbol{A}_\omega$ and emission densities $\boldsymbol{B}_\omega$ can be viewed as a HCRF with the corresponding components given as

$$\begin{array}{lcll}
\lambda_{\omega'}^{(Label)} & = & \log\alpha_{\omega} & \forall\omega' \in \Omega \\
\lambda_{i,j}^{(Tr)} & = & \log A_{i,j} & \forall i,j \in S_{\omega} \\
\lambda_{i,o,d}^{(Em)} & = & \log B_{\boldsymbol{i}}(o) & \begin{array}{c} \forall i \in S_{\omega} \\ \forall o \in \mathbb{H} \\ \boldsymbol{x}_{\boldsymbol{t}_{d}} \in \mathbb{N} \end{array} \\
\lambda_{i,d}^{(Occ)} & = & -0.5\left(\log 2\pi\sigma^{2} + \dfrac{{\mu_{i,d}}^{2}}{{\sigma_{i,d}}^{2}}\right) & \begin{array}{c} \forall i \in S_{\omega} \\ \boldsymbol{x}_{\boldsymbol{t}_{d}} \in \mathbb{R} \end{array} \\
\lambda_{i,d}^{(M1)} & = & \dfrac{\mu_{i,d}}{{\sigma_{i,d}}^{2}} & \begin{array}{c} \forall i \in S_{\omega} \\ \boldsymbol{x}_{\boldsymbol{t}_{d}} \in \mathbb{R} \end{array} \\
\lambda_{i,d}^{(M2)} & = & -\dfrac{1}{2{\sigma_{i,d}}^{2}} & \begin{array}{c} \forall i \in S_{\omega} \\ \boldsymbol{x}_{\boldsymbol{t}_{d}} \in \mathbb{R} \end{array}
\end{array} \tag{9}$$

in which $\mu_{i,d}$ and ${\sigma_{i,d}}^{2}$ refer to the mean and variance for the emission density at state i for the observation vector element at position d in case this element has an assumed normal distribution. In case this element is discrete, $B_{\boldsymbol{i}}(o)$ denotes the probability mass function for the state i for the hand configuration symbol $o \in \mathbb{H}$.

6 Experiments

6.1 Datasets

In order to create and learn our classification models, we acquired and organized a gesture dataset containing both static and dynamic gestures. We collected samples from 21 distinct subjects using Microsoft's Kinect sensor, registering both color and depth information. We note, however, that any of the results arising from our experiments are not restricted to this particular choice sensor, as all image processing has been done at the depth and intensity representation level. Those also have been gathered at varying luminosity levels, with both natural and artificial light sources.

The data acquisition occurred in two phases. At the first phase, the subjects had been asked to perform each of the fundamental 46 hand configurations of the Libras, purposely varying the location and orientation of the hand while keeping the configuration fixed. We sampled a total of 300 frames for each class to serve as training data to our static classifiers, giving a total of 13,800 training instances. Another independent and mutually exclusive sample of the same size has been drawn to be used as a validation set in the intermediate static gesture classification step.

In the second phase, we asked subjects to perform 13 natural words from the Libras. Those words have been repeated multiple times for a single subject, accommodating small variations between different performances. This gave us a total of 939 sequences of frames and a total sum of 139,154 frames in the dynamic gesture database. Furthermore, those sequences have been further divided into 10 mutually exclusive sets in a preparation for applying 10-fold cross-validation.

The words explored in this work are shown in Table 1. Those have been chosen due their particular difficulties: *Cabinet* involves the occlusion of the face, *Sorry* involves the tilting of the head; *I*, *Shoes, Like and Buy* require touching other body parts. *Sorry* and *Age* are performed with the same hand configuration but differs in spatial location and in a head tilting movement. *Cabinet* and *Car* are performed giving equal importance to both hands.

Table 1. Signed words contained in our dataset

Armário (Cabinet)	Idade (Age)
Sapato (Shoes)	Carro (Car)
Desculpa (Sorry)	Comprar (To Buy)
Eu (I)	Gostar (To Like)
Dia (Day)	Nome (Name)
Tchau (Bye)	Querer (To Want)
Oi (Hi)	

6.2 Static Gesture Classifiers

For the first processing layer we have created a number of SVMs with varying kernel functions and multi-class decision strategies. We also created feed-forward activation neural networks with a varying number of hidden neurons for comparison purposes. All classification machines have been designed to learn directly on a rescaled depth image of the user's hand. Those images have been gathered in the segmentation step as described in Section 5, but rescaled to a uniform size of 32×32 and then converted into a feature vector of 1024 positions. Despite the extremely simple nature of those features, the raw performance of this layer in correctly recognizing each of the hand configurations in $\mathbb{H}$ will not be as important as will the regularity of the classifier in classifying similar gestures with similar labels. Here, the hand configuration labels of the linguistic model are being used only as a guide – as long as the gesture recognition layer can detect patterns in the class labels generated at this stage, an absolute accuracy will not be required.

6.3 Dynamic Gesture Classifiers

After creating our static classifiers, we tagged the entire dataset of signed words and formed the feature vectors containing both hand configuration labels and trajectory information. In each cross-validation run an HMM has been created for each of the word labels in our dynamic gesture dataset. Those HMMs have then been used to initialize our HCRFs to compose the second processing layer. All results for Cohen's Kappa (κ) were averaged using ten-fold cross-validation, with variance pooled from all validation runs. We also compared the performance of the system without using the static classifier information at all, relying solely on trajectory and orientation information to perform classification.

7 Results

The static classifiers have shown some interesting results. Table 2 enlists selected results from tested SVM configurations and the average number of support vector (SV) evaluations as a measure of sparseness and efficiency.

Table 2. Results for the hand configuration candidate machines (first processing layer)

Kernel function	**Multiclass Strategy**	**Kappa** ± (0.95 C.I.)	**Average Number of Vector Evaluations**
Linear	DDAG	0.2390 ± 0.0074	45
	Max-Wins	0.2404 ± 0.0074	1,035
Quadratic	DDAG	0.4737 ± 0.0085	8,069
	Max-Wins	0.4790 ± 0.0085	339,509
Gaussian	DDAG	0.3401 ± 0.0042	8,707
	Max-Wins	0.3417 ± 0.0042	375,372

The graph below also shows the performance of the ANNs as a function of the number of neurons in their hidden layer. The best result reported by a neural network occured at 1000 neurons, with $\hat{\kappa}_{ANN} = 0.301$ and $\widehat{var}(\hat{\kappa}_{ANN}) = 4.05 \times 10^{-3}$. We note that after this peak the networks seemed to start overfitting the training data.

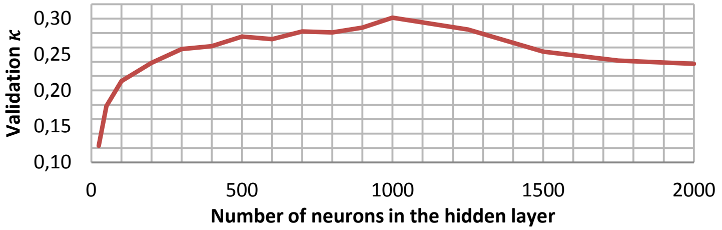


Fig. 2. Results for the hand configuration candidate networks (first processing layer)

Considering accuracy alone a quadratic SVM would be the clear choice to serve in the first classification layer. It performed statistically significantly better than all other models considered in this experiment. However, the performance cost in running such a machine can be huge. In contrast, the cost of computing a linear-SVM DDAG is much reduced. The DDAG based on linear SVMs has a constant evaluation rate, since its evaluation does not depend on the number of support vectors, but rather on the number of classes in the problem. Since a DDAG reduces the evaluation effort from computing 1035 decisions to only 45 constant-time decisions, we attain a much efficient, but weaker classifier, to serve as the first step in our dynamic gesture recognition system. We shall see shortly that the reduced performance will not be a problem due the probabilistic nature of the models on the second layer.

Table 3 below shows the results for the dynamic gesture experiments. The best combination of models was given by a SVM and a HCRF as the first and second

Table 3. Recognition results for the sequence classification models (second processing layer)

Static Gesture	Dynamic Gesture	Training	Validation
		Kappa ± (0.95 C.I.)	**Kappa** ± (0.95 C.I.)
SVM	HMM	0.8886 ± 0.0186	0.7951 ± 0.0717
SVM	HCRF	0.9466 ± 0.0131	0.8019 ± 0.0708
ANN	HMM	0.8610 ± 0.0205	0.7473 ± 0.0771
ANN	HCRF	0.9330 ± 0.0148	0.7727 ± 0.0743
None	HMM	0.6411 ± 0.0287	0.5308 ± 0.0898
None	HCRF	0.6638 ± 0.0283	0.5524 ± 0.0897

processing layers in the system, respectively. However, we note that results using ANNs were not significantly distinguishable from SVMs. The real difference occurred between the use of HMMs and HCRFs.

While this difference was not statistically visible in validation sets, the same could not be said of the training sets. It can be seen the models did not *overfit* – we obtained a statistically significant ($p < 0.01$) performance gain over training instances but no loss of generality over unobserved instances. Discriminative models were able to retain more knowledge without losing generalization when compared to HMMs.

On the other hand, when compared with models which did not use the hand configuration information at all, we achieved statistically significant results both for training *and* validation sets. As hinted before, the relatively small values for κ reported in Table 2 were no hurdle for the overall system performance. In fact, as in boosting mechanisms, where the combination of weak classifiers is able to produce one strong classifier, here the second processing layer is able to detect the patterns being output by the first layer, consolidating them into notable useful information when classifying new gestures. Thus the first processing layer effectively acts as a supervised feature extraction stage guided by linguistic information. Interestingly enough, the increased knowledge absorption by the discriminative models was mostly noticeable only in the presence of this first classification layer.

8 Conclusion

Here we have presented our approach to sign word recognition in Libras. By combining linear SVMs organized in Decision Directed Acyclic Graphs with Hidden Conditional Random Fields, we have shown how the use of discriminative models over generative ones helped improve the system's performance without causing a likely overfit. We have shown how the use of linguistic information has been helpful at designing such a gesture recognition system; and how our choice of simple features, based on a mixed vector with both discrete and continuous components have been suitable for this task. One can regard the first processing layer of our system as a guided feature extraction step rather than a definite classification stage. The use of a fast hand posture recognition layer based on DDAGs had been shown extremely useful when combined with trajectory and temporal information, achieving statistically significant results in comparison to models which did not use the presented technique.

Acknowledgements. We would like to express our thanks to FAPESP and CNPq. FAPESP sponsored the elaboration of this paper, and CNPq, the entire research. We would like to also express our gratitude towards our anonymous referees for sharing their valuable comments and opinions with us.

References

1. Pizzolato, E., Anjo, M., Pedroso, G.: Automatic recognition of finger spelling for LIBRAS based on a two-layer architecture. In: Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, pp. 969–973 (2010)
2. de Souza, C.R., Pizzolato, E.B., dos Santos Anjo, M.: Fingerspelling Recognition with Support Vector Machines and Hidden Conditional Random Fields. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) IBERAMIA 2012. LNCS, vol. 7637, pp. 561–570. Springer, Heidelberg (2012)
3. Mitra, S., Acharya, T.: Gesture recognition: A survey. IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews 37(3), 311–324 (2007)
4. Chen, X., Xiang, L.Y., Lantz, V., Wang, K., Yang, J.: A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41(6), 1064–1076 (2011)
5. Yang, H.-D., Sclaroff, S., Lee, S.-W.: Sign Language Spotting with a Threshold Model Based on Conditional Random Fields. IEEE Trans. Pattern Anal. Mach. Intell. 31(7), 1264–1277 (2009)
6. Bauer, B., Kraiss, K.-F.: Video-based sign recognition using self-organizing subunits. In: Proceedings of the16th International Conference on Pattern Recognition, vol. 2, pp. 434–437 (2002)
7. Dias, D., Madeo, R., Rocha, T., Bíscaro, H., Peres, S.: Hand movement recognition for brazilian sign language: a study using distance-based neural networks. In: Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, Georgia, USA, pp. 2355–2362 (2009)
8. Elmezain, M., Al-Hamadi, A., Michaelis, B.: Discriminative Models-Based Hand Gesture Recognition. In: International Conference on Machine Vision, Los Alamitos, CA, USA, pp. 123–127 (2009)
9. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P.: American Sign Language Recognition with the Kinect. In: Proceedings of the 13th International Conference on Multimodal Interfaces, Alicante, Spain, pp. 279–286 (2011)
10. Bowden, R., Windridge, D., Kadir, T., Zisserman, A., Brady, M.: A Linguistic Feature Vector for the Visual Interpretation of Sign Language. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 390–401. Springer, Heidelberg (2004)
11. Holden, E.-J., Lee, G., Owens, R.: Australian sign language recognition. Machine Vision and Applications 16(5), 312–320 (2005)
12. Carneiro, A., Cortez, P., Costa, R.: Reconhecimento de Gestos da LIBRAS com Classificadores Neurais a partir dos Momentos Invariantes de Hu. In: Interaction 2009, South America, São Paulo, pp. 190–195 (2009)
13. Wang, S., Quattoni, A., Morency, L.-P., Demirdjian, D.: Hidden Conditional Random Fields for Gesture Recognition. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, vol. 2, pp. 1521–1527 (2006)

14. Morency, L.-P., Quattoni, A., Darrell, T.: Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8 (2007)
15. Ferreira-Brito, L.: Por uma gramática de Línguas de Sinais, 2nd edn. Tempo Brasileiro, Rio de Janeiro (2010)
16. Igel, C., Hüsken, M.: Improving the Rprop Learning Algorithm. In : Symposium A Quarterly Journal In Modern Foreign Literatures, pp.115-121 (2000)
17. Riedmiller, M.: RProp - Description and Implementation Details. Technical Report, University of Karlsruhe, Karlsruhe (1994)
18. Dahl, G., Yu, D., Deng, L., Acero, A.: Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing (2012)
19. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn., 2200935th edn. Springer (2009)
20. Platt, J., Cristianini, N., Shawe-taylor, J.: Large Margin DAGs for Multiclass Classification. Advances in Neural Information Processing Systems, 547–553 (2000)
21. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features Machine Learning. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
22. Joachims, T.: Making large-scale support vector machine learning practical. In: Advances in Kernel Methods, pp. 169–184. MIT Press, Cambridge (1999)
23. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods, 1st edn. Cambridge University Press, Cambridge (2000)
24. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Comput. 13(3), 637–649 (2001)
25. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Waibel, A., Lee, K.-F. (eds.) Readings in Speech Recognition, pp. 267–296. Morgan Kaufmann Publishers Inc., San Francisco (1990)
26. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, pp. 282–289 (2001)
27. Sutton, C., McCallum, A.: Introduction to Statistical Relational Learning. In: Taskar, L. (ed.) An Introduction to Conditional Random Fields for Relational Learning. MIT Press (2007)
28. Mahajan, M., Gunawardana, A., Acero, A.: Training algorithms for hidden conditional random fields. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 273–276 (2006)
29. Viola, P., Jones, M.: Robust Real-time Object Detection. International Journal of Computer Vision (2001)
30. Bradski, G.: Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technology Journal(Q2) (1998)
31. Anjo, M., Pizzolato, E., Feuerstack, S.: A Real-Time System to Recognize Static Hand Gestures of Brazilian Sign Language (Libras) alphabet using Kinect. In: Proceedings of IHC 2012, the 6th Latin American Conference on Human-Computer Interaction, Cuiabá, Mato Grosso, Brazil (2012)

Classification and Outlier Detection Based on Topic Based Pattern Synthesis

Samrat Kokkula[1,⋆] and Narasimha Murty Musti[2]

[1] Citrix R&D India Pvt. Ltd
samrat.kokkula@citrix.com
[2] Indian Institute of Science, Bangalore, India
mnm@csa.iisc.ernet.in

Abstract. In several pattern classification problems, we encounter training datasets with an imbalanced class distribution and the presence of outliers, which can hinder the performance of classifiers. In this paper, we propose classification schemes based on the pre-processing of data using Novel Pattern Synthesis (NPS), with the aim to improve performance on such datasets. We provide a formal framework for characterizing the class imbalance and outlier elimination. Specifically, we look into the role of NPS in: Outlier elimination and handling class imbalance problem. In NPS, for every pattern its k-nearest neighbours are found and a weighted average of the neighbours is taken to form a synthesized pattern. It is found that the classification accuracy of minority class increases in the presence of synthesized patterns. However, finding nearest neighbours in high-dimensional datasets is challenging. Hence, we make use of Latent Dirichlet Allocation to reduce the dimensionality of the dataset. An extensive experimental evaluation carried out on 25 real-world imbalanced datasets shows that pre-processing of data using NPS is effective and has a greater impact on the classification accuracy over minority class for imbalanced learning. We also observed that NPS outperforms the state-of-the-art methods for imbalanced classification. Experiments on 9 real-world datasets with outliers, demonstrate that NPS approach not only substantially increases the detection performance, but is also relatively scalable in large datasets in comparison to the state-of-the-art outlier detection methods.

Keywords: Class Imbalance, Outlier detection, Classification, Latent Dirichlet Allocation, Dimensionality Reduction.

1 Introduction

One of the greatest challenges in data mining applications is dealing with imbalanced datasets [1]. A dataset is "imbalanced" if one or more classes has significantly less number of samples than the other classes. We encounter imbalanced datasets in many real-world domains, such as credit card fraud detection,

[⋆] This work was done while the author was a graduate student in the Dept of CSA, IISc, India.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 99–114, 2013.

oil spill identification [2] and text classification [3]. For example, in medical diagnosis of a certain type of cancer, out of total diagnosed samples, only a small number of samples actually have cancer. In case of binary classification problems, the class which contains substantially less number of samples is called the minority class (class c) and the other class is called the majority class (class $\overline{c}$). Usually in imbalanced datasets, cost of misclassifying the minority class samples is higher than the cost of misclassifying the majority class samples. To handle class imbalance problem we need to correctly classify the minority class samples. In this paper our focus is on 'two-class imbalanced classification problem'.

Traditional classifiers assume that the training set has roughly an equal number of patterns from both the classes. When such a classifier is trained on imbalanced dataset, it often shows a strong bias towards the majority class. This situation occurs as the standard learning algorithms aim to maximize the overall prediction accuracy. For example, a high classification accuracy may be achieved by simply assigning all samples to the majority class. In such a case, performance of data mining algorithms is reduced, especially the accuracy corresponding to the minority class. This leads to suboptimal classification performance on the minority class.

Sampling is a popular approach to address the imbalanced class problem, where extra patterns are added to the minority class and few patterns are removed from the majority class, as shown in SMOTE [4]. Although these sampling techniques are straightforward and computationally efficient, these methods are prone to either increased noise and duplicated samples or informative sample removal [5]. Alternatively, a cost-metric can be specified to force the classifier to give more importance to the minority class [6]. This requires choosing a correct cost-metric which is often unknown a priori.

An outlier [7] is an observation that is quite distinct from the remaining data in the sample space. Removal of outliers can improve the performance of a classifier. Hence, our aim is to pre-process the data so as to reduce the impact of outliers and class imbalance which could be associated with the dataset. In this paper, we exhibit a Novel Pattern Synthesis technique, which addresses both these issues simultaneously.

Novel Pattern Synthesis (NPS) technique [8] is developed to improve classification accuracy of the conventional k-Nearest Neighbour Classifier(kNNC) [9] as well as the edited 1-NN classifier. In NPS, for every pattern its k-nearest neighbours are found from the same class and a weighted average of the neighbours is calculated. Such a pattern is called a **Synthesized pattern**. For each pattern in the class, we find the corresponding Synthesized pattern. Then we use the Synthesized patterns in two ways: either we add Synthesized patterns to the given training data as shown in Algorithm 1 or replace the existing data with the Synthesized patterns as shown in Algorithm 2. In this article, we use NPS to handle both class imbalance and outlier detection simultaneously.

Our specific contributions in this paper are:

- We analyse the impact of class imbalance and outliers on classifiers and provide a formal framework for characterizing class imbalance.

- We empirically show how NPS can improve the classification accuracy of minority class samples in imbalanced datasets and how NPS handles outliers.
- We use two variants of Novel pattern synthesis and show that NPS improves classification accuracy using C4.5 Classifier.
- We thoroughly evaluate the differences between SMOTE and NPS, and explain how NPS outperforms SMOTE.
- We present comprehensive empirical results which show that NPS outperforms state-of-the-art imbalanced learning on several real-world datasets. For high-dimensional dataset, LDA is used for dimensionality reduction.
- We empirically show that NPS is superior to other state-of-the-art outlier detection methods in the literature.

The paper is organized as: Section 2 reviews existing techniques in literature to handle class imbalance problem and outlier detection methods. Section 3 introduces two variants of the NPS. Section 4 describes how pattern synthesis solves class imbalance and outlier detection. Section 5 presents the results of our experiments on real world datasets including a comparison with state-of-the-art methods and Section 6 concludes the paper.

2 Related Work

A majority of the current research to tackle the imbalanced class distribution problem can be grouped into two categories: algorithmic level approaches and data level approaches.

The focus of algorithmic level methods has been on modification of existing classification algorithms so that they can be more effective in dealing with imbalanced data, such as biasing towards the minority class, learning from one class [10] and assigning different classification-error costs on both class instances by weighting each type. For example, modifications to decision tree algorithm, such as HDDT [11] have been proposed to improve the standard C4.5.

Data level approaches rebalance the class distribution by resampling the data space. This way, preprocessing is done on the data and learning algorithm is not modified. Therefore, they are independent of the classifier used and usually more versatile. There have been several proposals for coping with skewed datasets [12]. For instance, there are sampling approaches in which we over-sample (i.e., duplicate) examples of the minority class, under-sample (i.e., remove) examples of the majority class. Random under-sampling [13] removes majority class instances at random until it contains as many examples as the minority class, the rare instances are often treated as noise. Hovy [14] discussed the impact of different re-sampling methods on the classifier accuracy and showed under sampling to be more effective than over sampling for improving the performance of models built using C4.5 decision trees.

Domingos [6] proposed Metacost, a re-costing method, which relabels the samples using loss function along with bagging. Risk on the original data is the same as that of relabelled data. According to Domingos, C4.5 Rules produce lower cost classifiers with under-sampling than with over-sampling. Also, Metacost reduces costs compared to cost-blind classifier which use C4.5 rules.

SMOTE [4] is an over-sampling technique, that synthesizes new minority class examples by randomly interpolating pairs of closest neighbours in the minority class. Minority class examples are introduced along the line segments that join any/all of the k minority class nearest neighbours. However, SMOTE encounters over generalization problem. It blindly generalizes the region of a minority class without considering the majority class.

SMOTEBoost [15] uses SMOTE to get new minority class examples in each iteration. Unlike the general boosting strategy, that is, changing the distribution of training data by updating the weights associated with each example, SMOTE-Boost alters the distribution by adding new minority-class examples. In [11], the Hellinger distance (HDDT) metric was used as the decision tree splitting criterion and shown to be insensitive towards class distribution skewness. HDDT capitalizes on the importance of designing a decision tree splitting criterion that captures the divergence in distributions while being skew-insensitive [16].

Random Forest (RF) [17] is designed to produce accurate predictions that do not overfit the data. RF combines both bagging and random feature selection. RF is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of classes output by individual trees. For imbalanced learning in case of Decision trees, branches useful to predict the minority are likely to be pruned since replacing them by a majority leaf can lead to a lower error rate. Class Confidence Proportion Decision Tree (CCPDT) [18] is robust and insensitive to size of classes and generates rules which are statistically significant.

Outlier detection is an important task in data mining with numerous applications including credit card fraud detection, video surveillance, stock market analysis, intrusion detection, and severe weather prediction. In [19], identification and elimination of outliers is done by a variant of SVM training. This modification of SVM training will explicitly identify outliers from training set. Our scheme deals with outlier elimination by using synthesized patterns.

Local Outlier Factor (LOF) [20] is density estimation based technique which has been extensively studied as a method for outlier detection. LOF approach is based on computing distances between data records. After LOF, many local density-based methods were proposed to compute the outlier factors, such as LOcal Correlation Integral (LOCI) [21], connectivity-based outlier factor [22], spatial local outlier measure [23], and local peculiarity factor [24]. Local Density Function (LDF) is an extension to LOF, which uses kernel density functions.

LDF [25] presents an outlier detection framework that is closely related to statistical non parametric density estimation methods with a variable kernel to yield a robust local density estimation. Outliers are then detected by comparing the local density of each point to the local density of its neighbours. In Feature Bagging method [26], results are combined from multiple outlier detection algorithms that are applied using different sets of features.

The behaviour of many classification algorithms degrades with an increase in dimensionality [27]. Data in high-dimensional space is quite sparse and finding nearest neighbours of a pattern is very challenging. So, there is a need for designing approaches which can work well with high-dimensional data. In our approach we use Latent Dirichlet Allocation (LDA) [28] to reduce dimensionality, LDA reduces any document to a fixed set of real-valued features, thus reducing the dimensionality called topics.

3 Novel Pattern Synthesis (NPS)

Novel pattern synthesis (NPS) is a technique to generate new patterns from the available set of patterns in a class. In NPS, for each pattern its corresponding k-nearest neighbours are found from the same class and a weighted average of the neighbours is taken; such a pattern is called a *Synthesized pattern.* For each pattern in the class, we find the corresponding Synthesized pattern. Japkowicz [12] presented algorithms to select the prototypes by using data preprocessed by NPS.

Bagging or Bootstrapped aggregation is a resampling technique used to reduce the variance by sampling with replacement from the given dataset. Here samples are selected from the given dataset with the replacements. Note that our scheme is different from bagging [29]. Also, SMOTE is an improvement over bagging [4].

3.1 Add Patterns to Minority Class

Algorithm 1. Add patterns to minority class

Input: A minority class dataset D = $\{X_1, X_2, ..., X_m\}$, no. of neighbours k and the no. of Synthesis iterations l

Output: Synthesized dataset D

```
Let new dataset $D_2 = D$
for $p = 1 \to l$ do
    dataset $D_1 = \emptyset$
    for $i = 1 \to m$ do
        let $k$ nearest neighbours of $X_i$ in $D_2$ be $X_i^1, X_i^2, ..., X_i^k$
        $X_B = \frac{1}{k} \sum_{j=1}^{k} X_i^j$
        $D_1 = D_1 \cup \{X_B\}$
        $D_2 = D_2 - \{X_i\}$
    end for
    $D_2 = D_1$
    $D = D \cup D_1$
end for
```

Property 1: Class imbalance is handled by Algorithm 1

Proof. Using Algorithm 1 we are adding Synthesized patterns to only minority class, resulting in an increase in the number of minority class samples. Thus class imbalance is reduced, as the number of minority class samples has increased. We use the Minimum Distance Classifier (MDC) [9] for Property 2.

Property 2: Post pattern synthesis the decision boundary will intersect the line segment between the two means for the Minimum Distance Classifier

Proof. Let us assume that the two classes are normally distributed:

$$c : N(\mu_i, \sigma^2 I),\ \overline{c} : N(\mu_j, \sigma^2 I),\ \sum = \sigma^2 I$$

The corresponding linear discriminant is of the form $w^T(x - x_0) = 0$ as mentioned by Burges [30], where $w = \mu_i - \mu_j$ and

$$x_0 = \frac{\mu_i + \mu_j}{2} - \frac{\sigma^2}{\parallel \mu_i - \mu_j \parallel^2} log \frac{P(\overline{c})}{P(c)} (\mu_i - \mu_j)$$

$$= \left\{ \frac{1}{2} - \frac{\sigma^2}{\parallel \mu_i - \mu_j \parallel^2} log \frac{P(\overline{c})}{P(c)} \right\} * \mu_i + \left\{ \frac{1}{2} + \frac{\sigma^2}{\parallel \mu_i - \mu_j \parallel^2} log \frac{P(\overline{c})}{P(c)} \right\} * \mu_j ..(1)$$

(1) is of the form,

$$x_0 = \lambda * \mu_i + (1 - \lambda) * \mu_j$$

where,

$$\lambda = \tfrac{1}{2} - \tfrac{\sigma^2}{\|\mu_i - \mu_j\|^2} log \tfrac{P(\overline{c})}{P(c)}$$

Typically, $0 \leq \lambda \leq 1$ such that the point x_1 lies between the two means on the line joining them. However, when there is a class imbalance such that

$$e^{\frac{\|\mu_i - \mu_j\|^2}{2\sigma^2}} < \frac{p(\overline{c})}{p(c)} \quad \Rightarrow \quad \frac{\parallel \mu_i - \mu_j \parallel^2}{2\sigma^2} < \ln \frac{p(\overline{c})}{p(c)} \Rightarrow \frac{1}{2} < \frac{\sigma^2}{\parallel \mu_i - \mu_j \parallel^2} * \ln \frac{p(\overline{c})}{p(c)}$$

$$\Rightarrow \frac{1}{2} - \frac{\sigma^2}{\parallel \mu_i - \mu_j \parallel^2} * \ln \frac{p(\overline{c})}{p(c)} < 0 \quad \Rightarrow \lambda < 0$$

Then, decision boundary (D_1) passes through x_0 which lies to the left of μ_i as shown in Figure. 1. This results in classifying all patterns with class label as $\overline{c}$. **Thus NPS improves classifier performance on minority class**.

Note: Similarly, if there exists class imbalance, such that

$$e^{\frac{\|\mu_i - \mu_j\|^2}{2\sigma^2}} < \tfrac{p(c)}{p(\overline{c})} \quad \Rightarrow \tfrac{\|\mu_i - \mu_j\|^2}{2\sigma^2} < \ln \tfrac{p(c)}{p(\overline{c})} \quad \Rightarrow \tfrac{1}{2} < \tfrac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \tfrac{p(c)}{p(\overline{c})}$$

$$\Rightarrow -\tfrac{1}{2} > \tfrac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \tfrac{p(\overline{c})}{p(c)} \quad \Rightarrow \tfrac{1}{2} - \tfrac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \tfrac{p(\overline{c})}{p(c)} > 1 \Rightarrow \lambda > 1$$

Therefore, decision boundary (D_2) lies to the right of μ_j as shown in Figure. 1, going in favour of c.

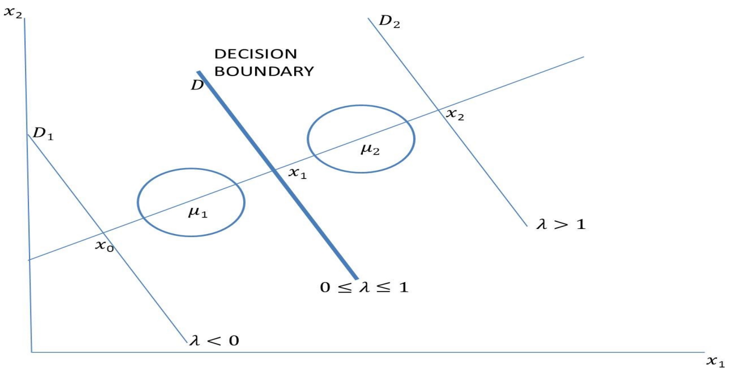


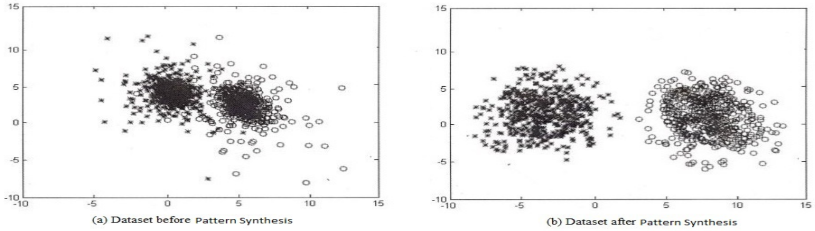
Fig. 1. Shift of Decision Boundary with Novel Pattern Synthesis

3.2 Replace Data by Synthesized Data

In Algorithm 2 the size of input and output datasets will be the same, but output dataset contains Synthesized patterns. This is depicted in Figure 2. Note that after NPS, separation between the two classes becomes prominent.

Algorithm 2. Replace data by Synthesized data

```
Input: A set D = {$X_1$, $X_2$, ..., $X_m$}, size $m$ and no. of neighbours $k$
Output: Synthesized dataset D
dataset $D_1 = \emptyset$
for $j = 1 \rightarrow m$ do
    let $k$ nearest neighbours of $X_i$ in $D = X_i^1, X_i^2, ..., X_i^k$
    $X_B = \frac{1}{k}\sum_{j=1}^{k} X_i^j$
    $D_1 = D_1 \cup \{X_B\}$
end for
$D = D_1$
```



(a) Dataset before Pattern Synthesis

(b) Dataset after Pattern Synthesis

Fig. 2. Synthesis of Data using Algorithm 2

Theorem 1. *Expected value of i.i.d patterns in a class remains same even after Novel pattern synthesis.*

Proof. Let k nearest neighbours of X be X_1, X_2, ..., X_k which are independent and identically distributed (i.i.d) random variables and $\overline{X}$ be the Synthesized point corresponding to pattern X. Let μ be the mean. Then

$$E[\overline{X}] = E[\tfrac{1}{k}\sum_{i=1}^{k} X_i] = \tfrac{1}{k} * E[\sum_{j=1}^{k} X_j] \ (since\ E[aX] = aE[X])$$

$$= \tfrac{1}{k}\sum_{j=1}^{k} E(X_j) \ (Since\ X_i^s\ are\ i.i.d) = \tfrac{1}{k}\sum_{j=1}^{k}\mu = \tfrac{k*\mu}{k} = \mu$$

Hence expected value of i.i.d patterns remains same after pattern synthesis.

Theorem 2. *Variance of the set of i.i.d patterns in a class decreases after pattern synthesis*

Proof. Consider the assumptions as in Theorem 1,

$$Var(X) = E[(X-\mu)^2] = E(X^2) - \mu^2$$

$$Var(\overline{X}) = Var(\tfrac{1}{k}\sum_{j=1}^{k} X_j) = \tfrac{1}{k^2}Var(\sum_{j=1}^{k} X_j) \ (\ Var(aX) = a^2Var(X))$$

$$= \tfrac{1}{k^2}\sum_{j=1}^{k} Var(X_j) \ (Since\ X_j^s\ are\ independent)$$

$$= \tfrac{1}{k^2}\sum_{j=1}^{k}\sigma^2 (since\ X_j^s\ are\ i.i.d) = \tfrac{1}{k^2}(k*\sigma^2) = \tfrac{\sigma^2}{k}$$

Therefore, $Var(\overline{X}) = \frac{\sigma^2}{k} \Rightarrow std.deviation(\overline{X}) = \frac{\sigma}{\sqrt{k}} = \frac{std.deviation(X)}{\sqrt{k}}$

Hence, standard deviation decreases by a factor of $\sqrt{k}$ after NPS where k is the number of nearest neighbours considered while generating Synthesized patterns.

4 Problems Associated with Classifiers

4.1 Class Imbalance

Class imbalance can affect the performance of a classifier as mentioned in Property 2. Prior probability for class c is given as

$$p(c) = \frac{number\ of\ patterns\ belonging\ to\ class\ c}{total\ number\ of\ patterns}$$

In this paper, we use NPS to circumvent the class imbalance problem. We add Synthesized patterns to minority class using Algorithm 1. Hence, number of patterns in the minority class will increase after Pattern synthesis. From Property 1 we proved that there is an improvement in the ratio of number of patterns of minority class to the number of patterns of majority class after NPS.

$$p(c) < p^b(c)$$

We have $p^b(\overline{c}) = p(\overline{c})$, since number of patterns in majority class remain the same after NPS.

$$\ln \frac{p^b(\overline{c})}{p^b(c)} < \ln \frac{p(\overline{c})}{p(c)} (\ \textit{since}\ p^b(\overline{c})\ =\ p(\overline{c})\ )\ ..\ (2)$$

Consider Property 2, where λ is defined as

$$\lambda = \frac{1}{2} - \frac{1}{(\mu_i - \mu_j)^T \sum^{-1} (\mu_i - \mu j)} log \frac{p(\overline{c})}{p(c)} \qquad and,$$

$$\lambda^b = \frac{1}{2} - \frac{1}{(\mu_i - \mu_j)^T \sum^{-1} (\mu_i - \mu j)} log \frac{p^b(\overline{c})}{p^b(c)}$$

where $\sum$ is the covariance vector. Using (2), we get

$$\lambda^b > \lambda$$

where λ^b is the λ value after NPS. Thus, λ increases after NPS. Repeat the NPS technique till there is a significant increase in the number of minority class patterns, such that $\lambda^b > 0$, and point x_0 lies between the two means μ_1 and μ_2. Hence the hyperplane separates the patterns of the two classes appropriately. As a result, class c is no more a minority class. NPS solves class-imbalance problem.

4.2 Outliers

An *Outlier* or outlying observation, is one that appears to deviate markedly from other members of the class in which it occurs.

In general, presence of outliers will drastically reduce the performance of a classifier. The outlier pattern typically is away from μ(Sample Mean). Using Algorithm 2, we replace the original pattern with the synthesized pattern. From Theorems 1 and 2, we conclude that after NPS the mean of the modified dataset remains the same but the standard deviation decreases. As a consequence, the distance between the mean and the outlier decreases after NPS. This indicates that impact of the outlier is averaged out or reduced. Hence NPS technique as in Algorithm 2 is useful in handling outliers.

The VC-dimension [30] of gap-tolerant classifier, a variant of SVM is

$$\min\{d, \lceil D_{max}^2 / M_{min}^2 \rceil\} + 1$$

Note that NPS using Algorithm 2 helps in increasing Margin(M_{min}) and decreasing Diameter(D_{max}), both these factors help in outlier elimination. It implies that NPS reduces the VC-dimension.

4.3 High Dimensional Space

In a high-dimensional datasets, the curse of high dimensionality will affect the performance of a classifier. For a query point in high-dimensional space, the ratio of the distance to the nearest neighbour to that of the farthest neighbour is almost 1. Let D_{max} be the farthest neighbour distance, D_{min} be the nearest neighbour distance and ϵ be a very small positive real number. Then

$$D_{max} \leq (1 + \epsilon) \times D_{min}$$

Sufficient condition of unstability is that the Pearson variation of the corresponding distance distribution degrades to 0 with increasing dimensionality. Distance function for high-dimensional data is given by Hsu and Chen [31].

5 Experiments

In order to evaluate the performance of NPS, we carried out a number of experiments. NPS is compared with seven models: unpruned C4.5 decision trees (C4.5) [32], SMOTE (SMT) [4], MetaCost (Meta) [6], Hellinger Distance decision trees (HDDT) [11], AdaBoost with SMOTE (SMT-BST) [15], Random Forests with SMOTE applied (RF-SMT) [4] and Class Confidence Proportion Decision Tree (CCPDT) [18]. We have used C4.5 decision tree [32] as baseline classifier. C4.5 is widely used to deal with imbalanced datasets [33], and C4.5 has also been included as one of the top-ten data-mining algorithms [34].

All methods (expect HDDT) were modeled based on the WEKA data mining toolkit [35] and HDDT is modeled using C++ on a computer with a Windows 7 64 bit OS running on 3.1-GHz Intel Core i5-3450 with 3.25 GB RAM. J48 in WEKA is used to model C4.5. -M1 option is set to C4.5, to increase the sensitivity of minority class. Value of k is set to 4 by default for Pattern synthesis and in all over-sampling techniques. To undersample the majority class for uniform distribution, we used SpreadSubsample [6]. For Metacost, cost of each class was set to the inverse of the class ratio.

Our experiments are performed on both high and low dimensional datasets. Table 1 summarizes high-dimensional datasets of domains: Text categorization datasets such as Cade, 20 News Group, WebKB and R8 are obtained from (http://web.ist.utl.pt/ acardoso/datasets/); Medical diagnosis datasets such as CNS, LYMPH, OVARY, PROST, NIPS (http://www.cs.toronto.edu/ roweis). High-dimensional datasets have less number of samples, we had artificially controlled the class skew on text categorization datasets and NIPS. The set was rebalanced for five separate class ratios: 1:1, 1:2, 1:4, 1:8, and 1:16. As finding nearest neighbours in high-dimensional space is challenging, we used LDA to reduce the dimensionality and then applied pre-processing technique.

Table 2 describes 16 real-world low-dimensional imbalanced datasets from various domains used in our experiments, from highly imbalanced (the minority 2.32%) to moderately imbalanced (the minority 30.00%). Oil dataset is obtained from Holte [2]. The CM1, KC1 and PC1 datasets were obtained from NASA

Table 1. Description of high-dimensional datasets

Dataset	Size	Dimensionality	Classes (pos, neg)	Minority(%)	Class Ratio
20NG	1908	15584	(hockey, others)	6.96%	133/1775
Cade	1504	25895	(ciencias, others)	8.75%	131/1372
WebKB	2765	7502	(student, others)	13.75%	380/2385
NIPS	320	13649	(neurobiology, others)	14%	45/275
R8	6215	14309	(acq, others)	15.5%	963/5251
OVARY	66	6000	(malignant, benign)	17%	11/55
PROST	89	6000	(prostate, others)	20%	18/71
LYMPH	77	7129	(folicular, B-cell)	22%	17/60
CNS	90	7129	(others, medulloblastomas)	30%	30/60

Table 2. Description of low-dimensional datasets, ordered in decreasing level of imbalance

Dataset	Size	Dimensionality	Classes (pos, neg)	Minority (%)	Class Ratio
Mammography	11,183	6	(true, false)	2.32%	260/10923
Oil	937	49	(true, false)	4.38%	41/896
Hypo-thyroid	3163	25	(true, false)	4.77%	151/3012
PC1	1109	21	(true, false)	6.94%	77/1032
Forest cover	38,500	10	(true, false)	7.00%	2695/35805
Glass	214	9	(3, other)	7.94%	44/170
Satimage	6435	36	(4, other)	9.73%	174/6261
CM1	498	21	(true, false)	9.84%	49/449
Pendigits	10,992	16	(true, false)	9.99%	1099/9892
New-thyroid	215	5	(3, other)	13.95%	30/185
KC1	2109	21	(true, false)	15.46%	326/1783
SPECT-F	267	44	(0, 1)	20.60%	55/212
Hepatitis	155	19	(1, 2)	20.65%	32/123
Vehicle	846	18	(van, other)	23.52%	199/647
Adult	48,842	14	(1, 2)	23.99%	11722/37120
German	1000	20	(2, 1)	30.00%	300/700

IV & V Facility MDP repository (http://mdp.ivv.nasa.gov/index.html). The remaining datasets were compiled from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml).

We empirically evaluate the effectiveness of NPS on real world datasets, with state-of-the-art methods, including LOF [20], LDF [25], Feature Bagging [26], Bagging [12], and Boosting [13]. In real datasets, the features of the original data include discrete features and continuous features. We present the results of a comparison between the NPS using C4.5 as the base classifier in Table 5. All the data are processed using the standard text processing techniques following the original steps of the methods [24].

Datasets used for our experiments are summarized in Table 3. Since rare class analysis is conceptually the same problem as the outlier detection, we employed those datasets for the purpose of outlier detection, where we detected rare classes as outliers. The datasets we used are Mammography, Lymphography, Ann-Thyroid, Satimage, Shuttle, Segment, LED which are available from UCI repository and KDD-Cup 1999 dataset is available from the UCI KDD Archive (http://kdd.ics.uci.edu/). These datasets do not directly correspond to the rare class problems or outlier detection problems but can be converted into binary problems by taking one small class (with less than 10% proportion present in the dataset) and remaining data records or the biggest remaining class as a second class. Therefore, we converted these datasets to suit binary classification.

Table 3. Description of datasets with outliers

Dataset	Modifications made in the dataset	Size	Dimensionality	Number of outliers	Outlier (%)
KDDCup 1999	U2R vs normal	60839	41	246	0.4%
Ann-thyroid	Class 1 vs Class 3	3428	21	73	2.13%
Mammography	Class 1 vs Rest	11183	6	260	2.32%
Lymphography	Merged Classes 2&4 vs Rest	148	18	6	4.05%
Ann-thyroid	Class 2 vs Class 3	3428	21	177	5.16%
Shuttle	Class 2, 3, 4, 6, 7 vs Class 1	14500	9	2 - 809	0.017% - 6.58%
Satimage	Smallest Class vs Rest	6435	36	626	9.73%
LED	Each Class vs Rest	10000	7	1000	10%
Segment	Each Class vs Rest	2310	19	330	14.29%

The evaluation was focused on each model's predictive performance on the testing sets. The Receiver Operating Characteristic (ROC) curve represents the trade-off between the detection rate on y-axis and the false alarm rate on x-axis. Area Under the ROC Curve (AUC) [36] measures the overall classification performance, and a perfect classifier has an AUC of 1.0. In order to compare the classifiers, we used 10-fold cross validation where each dataset was broken into 10 disjoint sets such that each set had (roughly) the same class distribution. Within each fold, the classification method is repeated ten times considering that the sampling of subsets introduces randomness. We then computed the AUC value as the average of each of these runs.

5.1 Performance Evaluation

Algorithm 1 is used to handle imbalanced datasets described in Table 1 and 2, AUC values of the compared methods are summarized in Table 4. Algorithm 2 is used to handle datasets with outliers mentioned in Table 3, AUC values of the compared methods are summarized in Table 5. Following observations are derived from the results:

Table 4. The (Average)AUC values for Novel pattern synthesis, in comparison with state-of-the-art imbalanced techniques, with the best highlighted in bold.

Dataset	C4.5	NPS	SMT	Meta	HDDT	SMT-BST	RF-SMT	CCPDT
Mammography	0.884	**0.948**	0.817	0.791	0.912	0.863	0.906	0.859
Oil	0.787	**0.869**	0.782	0.753	0.799	0.827	0.835	0.836
Hypo-thyroid	0.977	**0.988**	0.901	0.912	0.981	0.928	0.934	0.897
PC1	0.722	**0.897**	0.722	0.672	0.842	0.775	0.876	0.842
Forest cover	0.978	0.981	0.886	0.913	**0.982**	0.941	0.967	0.918
Glass	0.882	**0.913**	0.751	0.745	0.884	0.834	0.762	0.886
Satimage	0.906	**0.953**	0.782	0.934	0.911	0.951	0.936	0.917
CM1	0.675	**0.776**	0.647	0.561	0.684	0.736	0.754	0.716
Pendigits	0.985	**0.998**	0.979	0.751	0.992	0.996	0.995	0.989
New-thyroid	0.906	**0.993**	0.911	0.914	0.952	0.976	0.968	0.952
KC1	0.715	**0.842**	0.695	0.646	0.681	0.734	0.784	0.712
SPECT-F	0.674	**0.814**	0.642	0.659	0.665	0.736	0.749	0.703
Hepatitis	0.668	**0.819**	0.806	0.796	0.804	0.812	0.810	0.809
Vehicle	0.972	**0.986**	0.944	0.938	0.957	0.959	0.974	0.977
Adult	0.785	**0.886**	0.772	0.798	0.838	0.852	0.864	0.865
German	0.631	0.732	0.654	0.668	0.692	0.714	**0.736**	0.699
20NG	0.736	**0.862**	0.752	0.773	0.811	0.826	0.840	0.844
Cade	0.643	**0.764**	0.651	0.643	0.674	0.719	0.727	0.682
WebKB	0.769	**0.898**	0.780	0.766	0.813	0.851	0.856	0.817
NIPS	0.822	**0.938**	0.816	0.815	0.868	0.827	0.884	0.902
R8	0.827	0.984	0.815	0.868	0.911	0.931	0.945	**0.985**
OVARY	0.807	**0.898**	0.820	0.807	0.861	0.811	0.839	0.866
PROST	0.822	**0.879**	0.836	0.821	0.841	0.843	0.869	0.861
LYMPH	0.865	**0.954**	0.850	0.874	0.896	0.887	0.935	0.925
CNS	0.872	**0.927**	0.841	0.866	0.877	0.896	0.902	0.884

1. Compared to other models, NPS wins on 22 of the imbalanced datasets and 7 of the datasets with outliers. Results confirm that NPS is more effective than other class imbalanced techniques and outlier detection methods.
2. Average percentage improvement in AUC values of NPS over state-of-the-art methods are shown in Table 6 on imbalanced datasets and in Table 7 on datasets with outliers.
3. NPS has the best AUC result of **0.998** on Pendigits, which has a relatively high level of imbalance of **9.99%**.
4. As mentioned by Chawla [4], minority class sample may have majority class samples as their nearest neighbour, but this scenario is not seen in NPS as we find the nearest neighbour of a minority class sample in a minority class sample set only. Hence NPS performs better than different variants of SMOTE mentioned in experiments.

Table 5. The (Average)AUC values for NPS, in comparison with state-of-the-art outlier methods, with the best highlighted in bold

Dataset	C4.5	NPS	LOF	LDF	Bagging	Boosting	Feature-Bagging
KDDCup 1999	0.595	**0.961**	0.61	0.941	0.611	0.51	0.74
Ann-thyroid (Class 1 vs Class 3)	0.875	**0.992**	0.869	0.943	0.98	0.64	0.869
Mammography	0.67	**0.896**	0.64	0.824	0.74	0.56	0.8
Lymphography	0.795	**0.886**	0.717	0.879	0.832	0.643	0.685
Ann-thyroid (Class 2 vs Class 3)	0.51	0.958	0.761	0.957	**0.96**	0.54	0.769
Shuttle(average)	0.776	**0.991**	0.852	0.962	0.985	0.784	0.839
Satimage	0.881	**0.943**	0.864	0.930	0.928	0.781	0.806
LED	0.706	**0.811**	0.699	0.804	0.736	0.658	0.722
Segment	0.833	0.912	0.820	**0.915**	0.914	0.806	0.814

Table 6. Average percentage(%) improvement in AUC values of NPS over state-of-the-art imbalance classification methods

Datasets	C4.5	SMOTE	MetaCost	HDDT	SMT-BST	RF-SMT	CCPDT
Low-Dim	11.6	15.3	18.2	9.8	10.4	7.8	9.1
High-Dim	9.4	11.8	14.6	8.2	8.9	6.3	7.8

Table 7. Average percentage(%) improvement in AUC values of NPS over state-of-the-art outlier detection methods

Datasets	C4.5	LOF	LDF	Bagging	Boosting	Feature-Bagging
Outlier-datasets	18.4	16.8	3.8	7.3	22.4	14.5

5. In Section 4.1 and 4.2, we analyzed and theoretically proved that NPS performs better with imbalanced datasets and in the presence of outliers respectively. NPS will add a smoothing effect on outliers.
6. C4.5 with NPS performs well in low-dimensional space as it is able to capture best features. In the high-dimensional cases, the dimensionality is reduced using LDA and again C4.5 with NPS seems to be performing better than the other methods.
7. For German dataset, C4.5 with NPS does not perform better because of a larger height of decision tree and it causes over-fitting.

6 Conclusion

There are many techniques for outlier detection and handling the class imbalance problem. We present two variants of NPS to handle imbalanced datasets and outliers. We theoretically prove that NPS will solve class imbalance problem by adding synthesized patterns and eliminates outliers. The experiments on real-world datasets show that the performance of NPS evaluated by AUC metric is better than state-of-the-art imbalanced techniques and outlier detection methods, with C4.5 as the base classifier. LDA is useful in reducing the dimensionality of a high-dimensional datasets. Theoretical analysis and empirical evaluations on the real datasets demonstrate that NPS is more robust and effective for handling imbalanced datasets and outliers. In the future, we will work on multi-class imbalanced datasets and outlier detection on very high-dimensional datasets.

References

1. Yang, Q., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology and Decision Making 5(4), 597–604 (2006)
2. Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine Learning 30(2-3), 195–215 (1998)
3. Liu, Y., Loh, H.T., Sun, A.: Imbalanced text classification: A term weighting approach. Expert Syst. Appl. 36(1), 690–701 (2009)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res (JAIR) 16, 321–357 (2002)
5. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
6. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: KDD, pp. 155–164 (1999)
7. Hawkins, D.: Identification of outliers. Monographs on applied probability and statistics. Chapman and Hall, London (1980)
8. Hamamoto, Y., Uchimura, S., Tomita, S.: A bootstrap technique for nearest neighbor classifier design. IEEE Trans. Pattern Anal. Mach. Intell. 19(1), 73–79 (1997)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
10. Raskutti, B., Kowalczyk, A.: Extreme re-balancing for svms: a case study. SIGKDD Explor. Newsl. 6(1), 60–69 (2004)
11. Cieslak, D.A., Chawla, N.V.: Learning decision trees for unbalanced data. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 241–256. Springer, Heidelberg (2008)
12. Japkowicz, N.: Class imbalances: Are we focusing on the right issue?, pp. 17–23 (2003)
13. Dehmeshki, J., Karaky, M., Casique, M.V.: A rule-based scheme for filtering examples from majority class in an imbalanced training set. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 215–223. Springer, Heidelberg (2003)
14. Zhu, J., Hovy, E.H.: Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In: EMNLP-CoNLL, pp. 783–790. ACL (2007)
15. Chawla, N., Lazarevic, A., Hall, L., Bowyer, K.: Smoteboost: Improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)

16. Cieslak, D.A., Hoens, T.R., Chawla, N.V., Kegelmeyer, W.P.: Hellinger distance decision trees are robust and skew-insensitive. Data Min. Knowl. Discov. 24(1), 136–158 (2012)
17. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
18. Liu, W., Chawla, S., Cieslak, D.A., Chawla, N.V.: A robust decision tree algorithm for imbalanced data sets. In: SDM, pp. 766–777. SIAM (2010)
19. Xu, L., Crammer, K., Schuurmans, D.: Robust support vector machine training via convex outlier ablation. In: AAAI. AAAI Press (2006)
20. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) SIGMOD Conference, pp. 93–104. ACM (2000)
21. Papadimitriou, S., Kitagawa, H., Gibbons, P., Faloutsos, C.: Loci: Fast outlier detection using the local correlation integral. In: Proceedings of the 19th International Conference on Data Engineering, pp. 315–326. IEEE (2003)
22. Tang, J., Chen, Z., Fu, A., Cheung, D.: Enhancing effectiveness of outlier detections for low density patterns. In: Advances in Knowledge Discovery and Data Mining, pp. 535–548 (2002)
23. Sun, P., Chawla, S.: On local spatial outliers. In: ICDM, pp. 209–216. IEEE Computer Society (2004)
24. Yang, J., Zhong, N., Yao, Y., Wang, J.: Local peculiarity factor and its application in outlier detection. In: Li, Y., Sarawagi, B.L. (eds.) KDD, pp. 776–784. ACM (2008)
25. Latecki, L.J., Lazarevic, A., Pokrajac, D.: Outlier detection with kernel density functions. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 61–75. Springer, Heidelberg (2007)
26. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: Grossman, R., Bayardo, R., Bennett, K.P. (eds.) KDD, pp. 157–166. ACM (2005)
27. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2000)
28. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
29. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
30. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. 2(2), 121–167 (1998)
31. Hsu, C., Chen, M.: On the design and applicability of distance functions in high-dimensional data space. IEEE Transactions on Knowledge and Data Engineering 21(4), 523–536 (2009)
32. Drummond, C., Holte, R.C.: C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, pp. 1–8 (2003)
33. Drown, D.J., Khoshgoftaar, T.M., Seliya, N.: Evolutionary sampling and software quality modeling of high-assurance systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A 39(5), 1097–1107 (2009)
34. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., et al.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2008)
35. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, San Francisco (2005)
36. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering 17(3), 299–310 (2005)

Decremental Learning of Evolving Fuzzy Inference Systems: Application to Handwritten Gesture Recognition

Manuel Bouillon[*], Eric Anquetil, and Abdullah Almaksour

Université Européenne de Bretagne, France
INSA de Rennes, Avenue des Buttes de Coesmes, F-35043 Rennes
IRISA, Campus de Beaulieu, F-35042 Rennes
{Manuel.Bouillon,Eric.Anquetil,Abdullah.Almaksour}@irisa.fr
http://www.irisa.fr/intuidoc/

Abstract. This paper tackles the problem of incremental and decremental learning of an evolving and customizable fuzzy inference system for classification. We explain the interest of integrating a forgetting capacity in such an evolving system to improve its performances in changing environments. In this paper, we describe two decremental learning strategies to introduce a forgetting capacity in evolving fuzzy inference systems. Both techniques use a sliding window to introduce forgetting in the optimization process of fuzzy rules conclusions. The first approach is based on a downdating technique of least squares solutions for unlearning old data. The second integrates differed directional forgetting in the covariance matrices used in the recursive least square algorithm. These techniques are first evaluated on handwritten gesture recognition tasks in changing environments. They are also evaluated on some well-known classification benchmarks. In particular, it is shown that decremental learning allow to adapt to concept drifts. It is also demonstrated that decremental learning is necessary to maintain the system capacity of learning new classes over time, making decremental learning essential for the life-time use of an evolving and customizable classification system.

Keywords: Online Classification, Handwriting Recognition, Incremental Learning, Decremental Learning, Evolving Fuzzy Inference System, Recursive Least Squares, Concept Drifts; Forgetting.

1 Introduction

Evolving classification systems have appeared in the last decade to meet the need for recognizers that work in changing environments. They use incremental learning to adapt to the data flow and cope with class adding (or removal) at run time. This paper focuses on integrating a forgetting capacity in evolving fuzzy inference systems to improve their performance in changing environments. The aim of that forgetting capacity is twofold: first, maintain system capacity of

[*] Corresponding author.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 115–129, 2013.

learning new classes over time, and second, enable the system to follow changes of its environment (so-called concept drifts).

The target application of this work is the use of online handwritten gesture classifiers to facilitate user interactions[1] on pen-based interfaces like tablet computers, smart-phones, whiteboards, etc. Gestures can be drawn differently from one user to another, and users may want to add or remove gestures, as long as they use the application. Moreover, users would often change progressively their writing style. Novice users start drawing carefully and slowly their gestures, while they do them in a more fluid and rapid manner as they become expert. The classifier hence needs to evolve and follow the changes of the data flow. If most users will use a common subset of gestures, each user will need some specific gesture classes for his own usage but that other users won't use. In addition, classifier usage may change with time, and the end user may need to add, remove, or change gesture classes to fit his needs. That is why the classifier needs to be customizable by end users. To cope with these requirements, a forgetting capacity must be used to increase system reactivity and performance in such dynamic environments.

In this paper, we extend our evolving classification system *Evolve* [1] by integrating a forgetting capacity by the use of decremental learning. Two new decremental learning strategies are presented, both relying on a sliding window of data samples. The first technique uses this window to completely unlearn old data, by downdating the least squares solutions. The second technique uses this window to integrate differed directional forgetting in the learning process, and allow old data to be forgotten when needed.

We briefly present the architecture of *Evolve* and its incremental learning algorithm in Section 2. Section 3 presents forgetting interest and existing techniques. Our two new approaches are presented in Section 4 and 5. These approaches are then evaluated on some handwritten gesture recognition tasks in section 6. Section 7 concludes and discusses future work.

2 System Architecture

We focus here on Fuzzy Inference Systems (FIS) [12], with first order conclusion structure [15]. FIS have demonstrated their good performance for incremental classification of changing data flows [2]. Moreover, they can easily be trained online (in real time) and have a good behavior when new classes are added. [2] and [1] are recent examples of evolving FIS used for online classification.

A fuzzy inference system consists of a set of fuzzy inference rules like the following rule example.

$$\textbf{Rule}^{(i)} : \textbf{ IF } \mathbf{x} \text{ is close to } C^{(i)} \textbf{ THEN } \hat{\mathbf{y}}^{(i)} = (\hat{\mathbf{y}}_1^{(i)}\,; \ldots ; \hat{\mathbf{y}}_c^{(i)})^\top \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the feature vector, $C^{(i)}$ the fuzzy prototype associated to the i-th rule and $\hat{\mathbf{y}}^{(i)} \in \mathbb{R}^c$ the output vector. Rule premises are the fuzzy membership

[1] See `http://youtu.be/qOx4IY6uYf8` for a demonstration of gestural commands.

to rule prototypes, which are clusters in the input space. Rule conclusions are fuzzy membership to all classes, that are combined to produce the system output.

2.1 Premise Structure

Our model uses rotated hyper-elliptical prototypes that are each defined by a center $\boldsymbol{\mu}^{(i)} \in \mathbb{R}^n$:

$$\boldsymbol{\mu}^{(i)} = (\mu_1^{(i)}; \ldots, \mu_n^{(i)})^\top \tag{2}$$

and a covariance matrix $\Sigma^{(i)} \in \mathbb{R}^{n \times n}$ (where n is the number of features):

$$\Sigma^{(i)} = \begin{bmatrix} \sigma_1^2 & \ldots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{n,1} & \ldots & \sigma_n^2 \end{bmatrix} \tag{3}$$

To measure the activation degree of each fuzzy prototype, we use the multivariate normal distribution:

$$\alpha^{(i)}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma^{(i)}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^{(i)})^\top (\Sigma^{(i)})^{-1} (\mathbf{x} - \boldsymbol{\mu}^{(i)})^\top\right) \tag{4}$$

2.2 Inference Process

The inference process consist of three steps:

1. Activation degree is computed for each rule with Equation 4, and then normalized as follow:
$$\alpha^{(i)}(\mathbf{x}) = \frac{\alpha^{(i)}(\mathbf{x})}{\sum_{k=1}^{r} \alpha^{(k)}(\mathbf{x})} \tag{5}$$
where r is the number of rules.
2. System output is obtained from rule outputs using sum-product inference:
$$\hat{\mathbf{y}} = \sum_{k=1}^{r} \alpha^{(k)}(\mathbf{x}) \cdot \hat{\mathbf{y}}^{(k)} \tag{6}$$
3. Predicted class is the one corresponding to the highest output:
$$\text{class}(\mathbf{x}) = \arg\max_{k=1}^{c}(\hat{\mathbf{y}}_k) \tag{7}$$

2.3 Conclusion Structure

In a first order FIS, rule conclusions are linear functions of the inputs:

$$\hat{\mathbf{y}}^{(i)} = (l_1^{(i)}(\mathbf{x})\,; \ldots\,; l_c^{(i)}(\mathbf{x}))^\top \tag{8}$$

$$l_k^{(i)}(\mathbf{x}) = \mathbf{x}^\top \cdot \boldsymbol{\theta}_k^{(i)} = \theta_{0,k}^{(i)} + \theta_{1,k}^{(i)} \cdot x_1 + \cdots + \theta_{n,k}^{(i)} \cdot x_n \tag{9}$$

The i-th rule conclusion can be reformulated as:

$$\hat{\mathbf{y}}^{(i)\top} = \mathbf{x}^{\top} \cdot \Theta^{(i)} \tag{10}$$

with $\Theta^{(i)} \in \mathbb{R}^{n \times c}$ the matrix of the c linear function coefficients of the ith rule:

$$\Theta^{(i)} = (\boldsymbol{\theta}_1^{(i)}\, ; \ldots ; \boldsymbol{\theta}_c^{(i)}) = \begin{bmatrix} \theta_{1,1}^{(i)} & \cdots & \theta_{1,c}^{(i)} \\ \vdots & \ddots & \vdots \\ \theta_{n,1}^{(i)} & \cdots & \theta_{n,c}^{(i)} \end{bmatrix} \tag{11}$$

2.4 Incremental Learning Process

Let $\mathbf{x}_i$ $(i = 1..t)$ be the ith data sample, M_i the model at time i, and f the learning algorithm. The incremental learning process can be defined as follow:

$$M_i = f(M_{i-1}, \mathbf{x}_i) \tag{12}$$

whereas a batch learning process would be:

$$M_i = f(\mathbf{x}_1, \ldots, \mathbf{x}_i) \tag{13}$$

In our recognizer *Evolve* [1] learning process, both rule premises and conclusions are incrementally adapted:

1. Rule prototypes are statistically updated to model the runtime data:

$$\boldsymbol{\mu}_t^{(i)} = \frac{(t-1) \cdot \boldsymbol{\mu}_{t-1}^{(i)} + \mathbf{x}_t}{t} \tag{14}$$

$$\Sigma_t^{(i)} = \frac{(t-1) \cdot \Sigma_{t-1}^{(i)} + (\mathbf{x}_t - \boldsymbol{\mu}_{t-1}^{(i)})(\mathbf{x}_t - \boldsymbol{\mu}_t^{(i)})}{t} \tag{15}$$

2. Rule conclusion parameters are optimized on the data flow, using Recursive Least Squares (RLS) algorithm [9]:

$$\Theta_t^{(i)} = \Theta_{t-1}^{(i)} + \alpha^{(i)} \mathrm{C}_t^{(i)} \mathbf{x}_t (\mathbf{y}_t^{\top} - \mathbf{x}_t^{T} \Theta_{t-1}^{(i)}) \tag{16}$$

$$\mathrm{C}_t^{(i)} = \mathrm{C}_{t-1}^{(i)} - \frac{\mathrm{C}_{t-1}^{(i)} \mathbf{x}_t \mathbf{x}_t^{\top} \mathrm{C}_{t-1}^{(i)}}{\frac{1}{\alpha^{(i)}} + \mathbf{x}_t^{\top} \mathrm{C}_{t-1}^{(i)} \mathbf{x}_t} \tag{17}$$

New rules, with their associated prototypes and conclusions, are created by the incremental clustering method *eClustering* [3] when needed.

3 Decremental Learning

This paper focuses on the decremental learning of an evolving fuzzy inference system optimized with the Recursive Least Squares (RLS) algorithm.

Introducing forgetting in Recursive Least Squares is a well studied problem. The principle of forgetting in the RLS algorithm is to prevent the covariance matrix, which can be seen as a representation of the system gain, to go to zero (but without making it going to infinity either).

Interest of Forgetting. The interest of integrating forgetting in a learning system is twofold. First, a forgetting capacity is necessary to limit the weight of past data, and thus to maintain system capacity of learning new classes. If every data sample has the same relative weight, learning gain will tend to zero and system will become set with time. Second, a forgetting capacity allows the system to follow any change – concept drift [17] – of the data flow by forgetting obsolete data.

On the other side, we mustn't forget too much. Our purpose here is to improve the behavior of our system in non-stationary scenarios, but we mustn't reduce our its performances in stationary scenarios, nor make our system collapse by forgetting too much – so-called "catastrophic forgetting".

Existing Techniques. Several forgetting techniques for the Recursive Least Square (RLS) algorithm already exist, mostly in the literature dedicated to control of complex systems. The most common approach is to introduce an exponential weighting of data with time by the use of an exponential forgetting factor [10]. It comes to using the RLS algorithm on an (exponentially weighted) sliding window which is a good idea.

However, this algorithm behaves poorly when systems are not uniformly exited – it is known as the covariance "wind-up" problem [11] – which is the case in classification problems.

This problem happens when some of the covariance matrix elements grows abnormally and make the estimator overstep and diverge from its optimum value. This "wind-up" problem is due to the fact that this exponential factor produce an uniform forgetting whereas the excitation of the system varies with time, and is not uniform over the input space. Several "had-oc" strategies [5][14][8] have been proposed to deal with this instability problem but no universal solution has been found.

Proposed Approaches. The idea of using a sliding window is not new, it has been extensively used as a way to get a reactive estimator that follows concept drifts. We use this window to obtain two new decremental learning strategies applied to FIS optimization with the RLS algorithm. The first approach is based on downdating the least square solutions, which unlearn "old" data that leave the sliding window. The second approach is based on differed directional forgetting of the covariance matrix used in the RLS algorithm. The length of this window is a sensitive issue that will be discussed in Section 5.

4 A First Approach: Downdating Least Square Solutions

The principle of this approach is simple, we maintain a sliding window over the latest data, and we optimize rule conclusions on this window of data only.

As we can't afford to rebuild rule conclusions at the arrival of each new data, they are updated using the recursive least squares (RLS) algorithm [1]. In the same way, we here downdate least squares solutions at the departure of each "old" data from the window, without complete rebuilding. To do so, we use the de-recursive least squares algorithm in order to recursively unlearn "old" data.

De-Recursive Least Squares (DRLS) Algorithm. Let $\Theta^{(i)}_{s\to t} \in \mathbb{R}^{n\times c}$ be the ith rule conclusion, optimized on data samples from $(\mathbf{x}_s, \mathbf{y}_s)$ to $(\mathbf{x}_t, \mathbf{y}_t)$ (with $\mathbf{x}_k \in \mathbb{R}^n$ the input vectors, and $\mathbf{y}_k \in \mathbb{R}^c$ the binary output vectors).

We propose to downdate least squares solutions with the following DRLS formulas to unlearn "old" data sample $(\mathbf{x}_s, \mathbf{y}_s)$.

$$\Theta^{(i)}_{(s+1)\to t} = \Theta^{(i)}_{s\to t} - \alpha^{(i)}_s \mathrm{C}^{(i)}_{(s+1)\to t}\mathbf{x}_s(\mathbf{y}_s^\top - \mathbf{x}_s^\top \Theta^{(i)}_{s\to t}) \tag{18}$$

Where the covariance matrices $\mathrm{C}^{(i)}$ are updated as follows.

$$\mathrm{C}^{(i)}_{(s+1)\to t} = \mathrm{C}^{(i)}_{s\to t} + \frac{\mathrm{C}^{(i)}_{s\to t}\mathbf{x}_s\mathbf{x}_s^\top \mathrm{C}^{(i)}_{s\to t}}{\frac{-1}{\alpha^{(i)}_s} + \mathbf{x}_s^\top \mathrm{C}^{(i)}_{s\to t}\mathbf{x}_s} \tag{19}$$

These DRLS formulas are proven in Appendix A.

This real time techniques gives similar results as a batch learning on the window of data. As a result, the learning gain is kept bounded and the system keeps its learning properties over time. New classes are learned as quickly after a long learning time as after a short one. Moreover, this sliding window enables the system to easily adapt to concept drifts. Data samples that leave the window are unlearned, and system quickly adapts to new concept as old one is unlearned.

5 A Second Approach: Differed Directional Forgetting

We present here a simple but very efficient new type of forgetting, without any exponential factor, and based on past data. Our strategy is to use forgetting but in the direction of "old" data samples. As for the previous approach, we use a sliding window that enable us to remove "old" data weight from the covariance matrices, when they leave the window. The weight removal can be done using only the second de-recursive formula for the covariance matrix downdating (Equation 19).

$$\mathrm{C}^{(i)}_{(s+1)\to t} = \mathrm{C}^{(i)}_{s\to t} + \frac{\mathrm{C}^{(i)}_{s\to t}\mathbf{x}_s\mathbf{x}_s^\top \mathrm{C}^{(i)}_{s\to t}}{\frac{-1}{\alpha^{(i)}_s} + \mathbf{x}_s^\top \mathrm{C}^{(i)}_{s\to t}\mathbf{x}_s}$$

From Unlearning to Forgetting. The De-Recursive Least Square (DRLS) algorithm enable us to unlearn, to remove completely the effect some data have had in the optimization process. This unlearning take place in two steps: a first step to downdate the covariance matrix and a second step to downdate the rule conclusion coefficients.

Downdating the covariance matrix is indispensable in order to limit old data weight. Unlearning rule conclusions make the system change and follow concept drifts. However, if downdating the covariance matrix has little impact on the system recognition performances, unlearning rule conclusions deconstructs the conclusion matrix – erases the knowledge previously acquired – and reduces system performances.

The learning of the conclusions at the arrival of a data sample is a step forward, their unlearning is a step backward. At the arrival of a new sample, this step backward will be redone, but with some minor adjustments. It is more interesting not to step backward, but to wait and sidestep instead when a new data sample arrives.

Following this philosophy, the second approach only updates the covariance matrix, and not the rule conclusions. By doing so, we go from a downdating approach to a forgetting one. Old knowledge isn't erased, but will be overwritten by future knowledge since no more weight is given to the corresponding data in the covariance matrix.

Discussion of the Window Length

The sensitive issue of these two approaches is the length of the sliding window. Indeed, performances are directly linked to the window length. In steady environment, the longer is the window, the lower is the error rate (until a minimum rate). However, a too long window reduces system reactiveness and thus deteriorates performances in changing environments.

We focus on having a large enough window to avoid performances reduction in stationary environments. Such a window is sufficient to limit old data weight and to adapt to concept drifts in changing environments. Even if a fixed length sliding window is not optimal, it provides a good behavior as it will be demonstrated experimentally.

The window minimum length is a problem dependent variable that can be empirically determined. It depend on the number of classes, the set of features used and the intrinsic difficulty of the classification problem. Nevertheless, a satisfying window length can easily be found experimentally as will be shown in the next section.

A drawback of these approaches is the need for memorizing all the data in the window to be able to unlearn/forget them. However, the increase of memory requirements remains quite small, compared to the initial algorithm, as will be shown in the next section.

6 Experimental Results

Starting from the state-of-the-art recognizer *Evolve* [1], we implemented our two new approaches: *Evolve D* – with Downdating – and *Evolve F* – with Forgetting.

As this work is applied to online handwritten gesture recognition, we first evaluate our two new systems, and the reference one, on handwritten gesture recognition tasks in changing environments. Then, we evaluate these two approaches on two well-known classification benchmark datasets.

Incremental Evaluation Protocol. To evaluate our systems in a realistic way, we used an incremental evaluation protocol called predictive sequential – or prequential [7] – with a sliding window to converge to the holdout error.

As an incremental system first tries to recognize a data sample, and then learns from it once it has the true label, we evaluated our systems in a similar way. Each data sample is first used as a test sample, and then as a learning sample. Error rates are then computed between every test points.

6.1 Evaluation on Gesture Recognition Tasks

As this work is applied to online handwritten gesture recognition, we evaluated our new approaches on two handwritten gesture databases: ILGDB[2] [13] and IRONOFF-digits [16].

The Heterogeneous Baseline Feature set (HBF49) [4] and a sliding window length of 50 data samples is used in both cases. Such a window length (with HBF49) only require to memorize a 50 by 50 matrix, which correspond to the size of the covariance matrix of one rule.

ILGDB: This database contains handwritten gestures that have been collected in an immersive environment. It is composed of 6629 mono-stroke gestures, belonging to 21 classes, which were written by 38 writers. This database is very interesting for several reasons.

First, gestures are ordered chronologically in their drawing order which allows us to see changes in writer style with time, as the writer changes from novice to expert. Second, class frequencies vary, from 5 to 17 examples per class.

Third, for part of the database, gesture classes are user defined. These features makes this database very realistic and representative of the real use of an online recognition system. Some gesture examples are shown in figure 1.

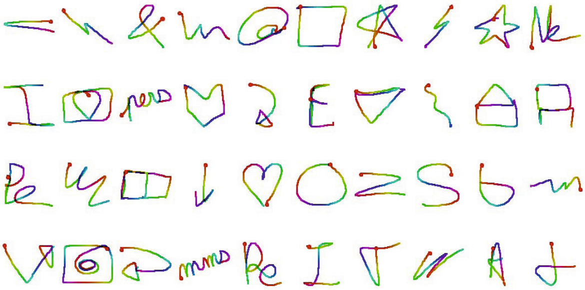


Fig. 1. Gesture samples from ILGDB group 1 (free gestures)

[2] Freely available at `http://www.irisa.fr/intuidoc/ILGDB.html`

Ironoff-digits: The interest of this database is that it offers more gestures (in writer independent mode), but like most classic benchmark databases, IRONOFF-digits is not ordered (there is no chronological evolution of the data with time).

System Inertia to Novelty (New Classes). We test the reactiveness of our systems, and the evolution of the inertia of their model with time. To that purpose, we train the different systems with seven classes during a varying period of time (70, 700 and 2100 training samples) and then introduce three other classes. We measure the time needed by the different systems to learn those new classes. Results averaged over 20 different data orders are shown Figure 2.

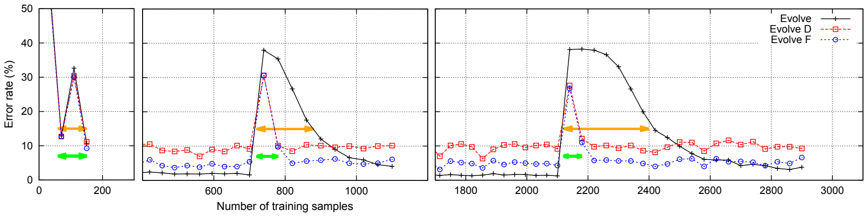


Fig. 2. Scenario "inertia and reactiveness" – IRONOFF-digits database – Adding new classes after different training times

If all systems behave similarly after 70 training samples, results are different after 700 or 2100 samples.

Looking at the results of our reference system without forgetting *Evolve*, we can clearly see that its reactiveness decreases with time, and that the model tends to become set. After 2100 training samples, the system needs 400 more samples to resume to an error rate under 10%, which makes it unusable for quite some time.

On the contrary, both systems using decremental learning need the same time to learn and adapt their model to the novelty, whenever it is introduced. Their reactiveness is independent of their age.

This test scenario shows clearly the necessity of integrating a forgetting capacity into an incremental learning system operating in a changing environment.

Performance in Slow Changing Environment. We test our systems on a scenario simulating slow concept drifts. For this purpose, we used the ILGDB 11 writers of group 3 (whose gestures for each class are identical) in a row. We computed the error rate for each writer, without taking into account the first 3 samples per class per writer, to measure system performance when concept drifts are learned. Mean results over 100 writer orders are plotted Figure 3.

Writers, and their drawing styles, change but the base gesture of each class stays unchanged. This test scenario is thus nearly stationary and doesn't really require the use of decremental learning. Our reference system *Evolve* achieves

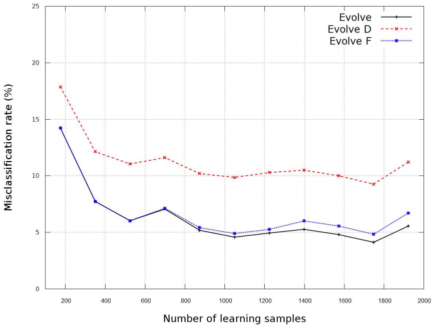


Fig. 3. Performance in slow changing environment – Using ILGDB 11 writers of group 3 (fixed gestures) in a row

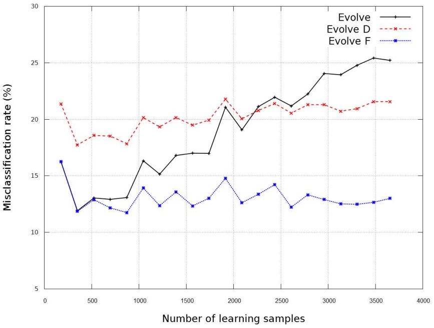


Fig. 4. Performance in fast changing environment – Using ILGDB 21 writers of group 1 (free gestures) in a row

quite good results here with an average error rate of 6.31%. Without forgetting, the system is learning from every writers and build a writer independent model.

Evolve D obtains an average error rates of 11.25%. Its performances are limited by the restrained number of data it is learning from (as old data samples are completely unlearned). The use of downdating prevent the system to build a writer independent model and force it to adapt to the current writer (with very few data).

Evolve F reaches an average error rates of 6.70% which is nearly as good as *Evolve*. This is due to the fact that old data samples are not unlearned, but only their corresponding weights are removed from the covariance matrix used in the RLS algorithm. This forgetting allow the system to keep previously acquired knowledge until it is overwritten by some new information.

Performance in Fast Changing Environment. We also test our systems on a scenario simulating fast concept drifts. For this purpose, we used the ILGDB 21 writers of group 1 (whose gestures are different for the same class) in a row. We computed the error rate for each writer, without taking into account the first 3 samples per class of each writer, to measure system performance after concept drifts are learned. Mean results over 100 writer orders are plotted Figure 4.

This testing scenario is quite difficult because the gesture of every classes completely change as the writer changes. The use of decremental learning is here mandatory.

The results on this scenario are sharply contrasted. Without any forgetting capacity, *Evolve* is not able to adapt its model to the changes in the data flow and ends up with an error rate of 25%.

Evolve D is again limited by the few data its model is build from. Although stable, its performance are quite poor: 20.26 % error rate in average.

Evolve F, behaves well and obtains reasonable performance with an average error rate of 13.08 %. The use of forgetting allow *Evolve F* to follow the fast concept drifts.

6.2 Evaluation on UCI Benchmark Datasets

We also evaluated our new methods on two well-known classification benchmarks from the UCI machine learning repository [6] to test our methods outside the field of handwritten gesture classification and with other feature sets. We chose some datasets following two criteria. First, they are multi-classes problems (our systems are optimized for classification problem with more than two classes), and second, they are large enough datasets and allow the incremental learning and testing of our systems on the long run. We chose the Pen-Digits and Japanese-Vowels datasets.

In this context, the size of the sliding window has to be changed to adapt to those problems difficulty. We used a window length of 500 samples which empirically gives the best results. It increases the memory requirements of our systems (by storing a 500 by 14 or 16 matrix) but is necessary to keep good performances for those more difficult problems.

Japanese-Vowels. The Japanese-Vowels dataset is composed of about 10'000 samples. This dataset contains 9 classes: 9 different male speakers that must be recognized using 14 features extracted from two Japanese vowels.

Pen-Digits. The Pen-Digits dataset contains about 11'000 handwritten digits that were recorded on a pen based interface. This dataset contains 10 classes and uses 16 features.

Performances in Stationary Environments. We evaluated our two new approaches in stationary environment with the Pen-Digits and Japanese-Vowels datasets to see the effects of decremental learning when it is not needed. Results averaged on 100 different data orders are presented Figure 5.

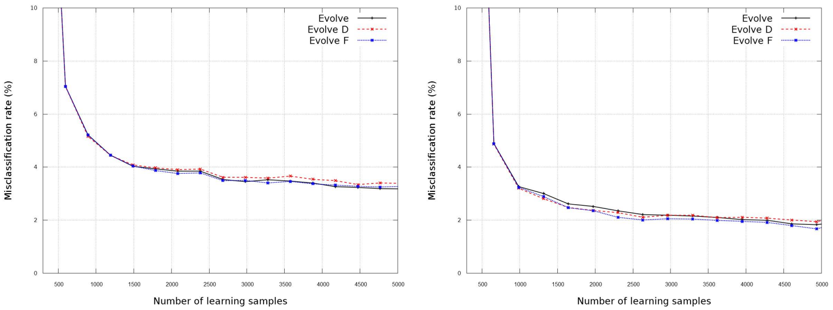


Fig. 5. Performance on Japanese-Vowels and Pen-Digits datasets (UCI repository) in stationary environment

If the use of decremental learning doesn't improve performances, which is quite normal in stationary environment, but doesn't deteriorate them either.

Performances in Changing Environments. As the aim of decremental learning is to improve performances in changing environments, we simulated one by adding half of the classes in the middle of the test. Results averaged on 100 different data orders are presented Figure 6.

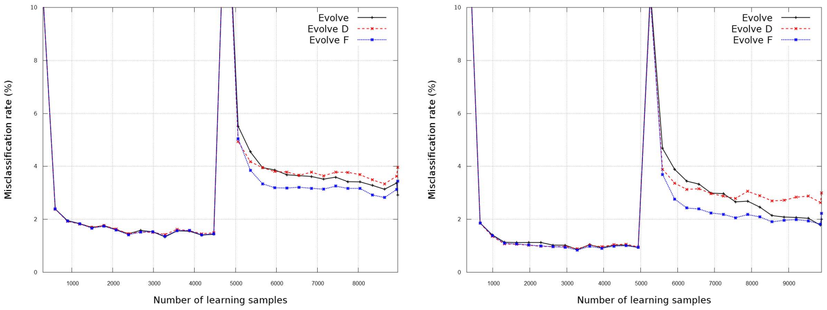


Fig. 6. Performance on Japanese-Vowels and Pen-Digits datasets (UCI repository) in changing environment (half of the classes are added in the middle of the test)

In this changing environments, decremental learning allow to adapt faster to the adding of new classes. Using downdating (*Evolve D*) provides a limited gain as final performances are limited, but using forgetting (*Evolve F*) allows to reach final performance faster. *Evolve F* reaches final performances in 1500 and 3000 samples respectively whereas *Evolve* needs 4500 and 5000 samples respectively to converge (on Japanese Vowels and Pen Digits respectively).

6.3 Results Discussion

The performances of the first approach – downdating (*Evolve D*) – are quite limited compared to the second approach – forgetting (*Evolve F*). When downdating, we unlearn completely "old" data and deteriorate least squares solutions. On the other hand, when forgetting, the least squares solutions aren't modified. "Old" data weight is removed, but the knowledge stemming from them is kept until any new data makes it mandatory to overwrite it.

On stationary scenarios, decremental learning isn't necessary. *Evolve F*, using (differed directional) forgetting, obtains similar recognition rates than *Evolve*, our reference model. The performances of *Evolve D*, using downdating, are not as good due to the limited number of samples its model is build from.

Results on non-stationary scenarios are quite clear-cut. They show the necessity of using decremental learning to maintain the reactiveness of the system over time. Without downdating, the system model tends to become set, and take ages to learn some novelty. Decremental learning is necessary to maintain system ability to learn new classes over time.

In the same way, we showed that decremental learning is essential to face concept drifts. Without downdating, the system model becomes more and more complex and performance slowly but surely collapses. Decremental learning allows to discard obsolete data and thus enables the system to focus on current system environment.

These features are very important as our goal is to obtain an evolving classifier to facilitate user interaction on touch sensitive interfaces. Use-cases of touch sensitive interfaces are various and aplenty, so it is essential to give the user the ability to customize the classifier to his needs and adapt to its changes of use.

7 Conclusion

In this paper, we investigated two decremental learning strategies, to introduce a forgetting capacity in evolving fuzzy inference system used for classification.

Both techniques use a sliding window to introduce forgetting in the optimization process of fuzzy rule conclusions. The first approach is based on a downdating technique of least squares solutions for unlearning old data. The second integrates differed directional forgetting in the covariance matrices used in the recursive least squares algorithm.

This work is applied to handwritten gesture recognition as our goal is to obtain an evolving and customizable classifier to facilitate user interaction on touch sensitive interfaces. Such an applicative context in clearly non-stationary and requires the classifier to allow class adding and to follow concept drifts to adapt to its use.

We showed that our second approach *Evolve F* – differed directional forgetting – performs well in changing environments, without deteriorating performances in stationary environments. Our method produces a similar effect than existing techniques to introduce forgetting in the recursive least squares algorithm, but without the problem of estimator “wind-up”.

Future Work A judicious improvement to this method would be to manage the window size adaptively. The window length could be increased as long as it improve performance, and reduced when concept drifts are detected.

Another direction that should be explored is managing adaptively the content of the sliding window. It could be interesting to keep more data samples of the classes that are harder to recognize, or that are more subject to confusion with other classes, and less samples of the classes that are easier to recognize.

A Appendix: Proof of De-recursive Least Squares

Let $\Theta^{(i)}_{s\to t} \in \mathbb{R}^{n\times c}$ be the i^{th} rule conclusion, optimized on data samples from $(\mathbf{x}_s, \mathbf{y}_s)$ to $(\mathbf{x}_t, \mathbf{y}_t)$ (with $\mathbf{x}_k \in \mathbb{R}^n$ the input vectors, and $\mathbf{y}_k^\top \in \mathbb{R}^c$ the binary output vectors).

A.1 Downdating Least Square Solutions

The rule cost functions are (for $1 \leq i \leq r$, where r is the number of rules)

$$J^{(i)}_{s\to t} = \sum_{i=s}^{t} \alpha^{(i)}(\mathbf{y}_i - \mathbf{x}_i^T \Theta^{(i)}_t)^2 \tag{20}$$

$$J^{(i)}_{s\to t} = (\mathrm{Y}_{s\to t} - \mathrm{X}_{s\to t}\Theta^{(i)}_{s\to t})^\top \mathrm{A}^{(i)}_{s\to t}(\mathrm{Y}_{s\to t} - \mathrm{X}_{s\to t}\Theta^{(i)}) \tag{21}$$

where $\mathrm{X}_{s\to t} = [\mathbf{x}_s; \dots; \mathbf{x}_t]^\top$ with $\mathbf{x}_k = [x_{k,1}; \dots; x_{k,n}]^\top$ and n the number of dimensions of the input space ; $\mathrm{Y}_{s\to t} = [\mathbf{y}_s^\top; \dots; \mathbf{y}_t^\top]^T$ with $\mathbf{y}_k = [y_{k,1}; \dots; y_{k,n}]$ and $y_{k,l} = 1$ if $y_{k,l}$ belong to class l and $y_{k,l} = 0$ otherwise. The weighting matrices $\mathrm{A}^{(i)}_{s\to t}$ are defined as

$$\mathrm{A}^{(i)}_{s\to t} = \alpha^{(i)}_s \cdot \mathrm{Id} \tag{22}$$

with $\alpha^{(i)}_k$ the firing level of the i^{th} rule by data sample x_k, and Id the identity matrix. The weighted least squares solutions are

$$\Theta^{(i)}_{s\to t} = \mathrm{C}^{(i)}_{s\to t}\mathrm{X}^\top_{s\to t}\mathrm{A}^{(i)}_{s\to t}\mathrm{Y}_{s\to t} \tag{23}$$

with the information matrices

$$R^{(i)}_{s\to t} = (\mathrm{C}^{(i)}_{s\to t})^{-1} = \mathrm{X}_{s\to t}\mathrm{A}^{(i)}_{s\to t}\mathrm{X}^\top_{s\to t} \tag{24}$$

The information matrices can easily be downdated

$$\mathrm{R}^{(i)}_{(s+1)\to t} = \mathrm{R}^{(i)}_{s\to t} - \mathbf{x}_s\alpha^{(i)}_s\mathbf{x}_s^\top \tag{25}$$

and using the matrix inversion lemma (Woodbury identity)

$$\begin{aligned}&\left(A + XBX^\top\right)^{-1} \qquad (26)\\ &= A^{-1} - A^{-1}X\left(B^{-1} + X^\top A^{-1}X\right)^{-1}X^\top A^{-1}\end{aligned}$$

with $A = \mathrm{R}^{(i)}_{s\to t}$, $X = \mathbf{x}_s$ and $B = -\alpha^{(i)}_s$, one can easily obtain equation 5 to downdate the covariance matrices.

A.2 Downdating Covariance Matrices

From equation 23 we can write:

$$\begin{aligned}\mathrm{R}^{(i)}_{(s+1)\to t}\Theta^{(i)}_{(s+1)\to t} &= \mathrm{X}^\top_{(s+1)\to t}\mathrm{A}^{(i)}_{(s+1)\to t}\mathrm{Y}_{(s+1)\to t}\\ &= \mathrm{X}^\top_{s\to t}\mathrm{A}^{(i)}_{s\to t}\mathrm{Y}_{s\to t} - \mathbf{x}_s\alpha^{(i)}_s\mathbf{y}_s\\ &= \mathrm{R}^{(i)}_{s\to t}\Theta^{(i)}_{s\to t} - \mathbf{x}_s\alpha^{(i)}_s\mathbf{y}_s\\ &= (\mathrm{R}^{(i)}_{(s+1)\to t} + \mathbf{x}_s\alpha^{(i)}_s\mathbf{x}_s^\top)\Theta^{(i)}_{s\to t} - \mathbf{x}_s\alpha^{(i)}_s\mathbf{y}_s\\ &= \mathrm{R}^{(i)}_{(s+1)\to t}\Theta^{(i)}_{s\to t} + \mathbf{x}_s\alpha^{(i)}_s\mathbf{x}_s^\top\Theta^{(i)}_{s\to t} - \mathbf{x}_s\alpha^{(i)}_s\mathbf{y}_s\\ &= \mathrm{R}^{(i)}_{(s+1)\to t}\Theta^{(i)}_{s\to t} - \mathbf{x}_s\alpha^{(i)}_s(\mathbf{y}_s - \mathbf{x}_s^\top\Theta^{(i)}_{s\to t})\end{aligned}$$

which yield equation 18.

References

1. Almaksour, A., Anquetil, E.: Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers. Evolving Systems 2, 25–33 (2011)
2. Angelov, P., Zhou, X.: Evolving fuzzy-rule-based classifiers from data streams. IEEE Transactions on Fuzzy Systems 16(6), 1462–1475 (2008)
3. Angelov, P., Filev, D.: An approach to online identification of takagi-sugeno fuzzy models. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetic 34(1), 484–498 (2004)
4. Delaye, A., Anquetil, E.: HBF49 feature set: A first unified baseline for online symbol recognition. Pattern Recognition 46(1), 117–130 (2013)
5. Fortescue, T., Kershenbaum, L., Ydstie, B.: Implementation of self-tuning regulators with variable forgetting factors. Automatica 17(6), 831–835 (1981)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
7. Gama, J., Sebastião, R., Rodrigues, P.P.: Issues in evaluation of stream learning algorithms. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 329–338 (2009)
8. Hägglund, T.: New estimation techniques for adaptive control (1983)
9. Haykin, S.O.: Adaptive Filter Theory, 4th edn. Prentice Hall (2001)
10. Kulhavy, R., Zarrop, M.: On a general concept of forgetting. International Journal of Control 58(4), 905–924 (1993)
11. Liavas, A., Regalia, P.: On the numerical stability and accuracy of the conventional recursive least squares algorithm. Trans. Signal Processing 47(1), 88–96 (1999)
12. Lughofer, E.: Evolving fuzzy models: incremental learning, interpretability, and stability issues, applications. VDM Verlag Dr. Müller (2008)
13. Renau-Ferrer, N., Li, P., Delaye, A., Anquetil, E.: The ILGDB database of realistic pen-based gestural commands. In: International Conference on Pattern Recognition (ICPR 2012), tsukuba, Japan (November 2012)
14. Salgado, M.E., Goodwin, G.C., Middleton, R.H.: Modified least squares algorithm incorporating exponential resetting and forgetting. International Journal of Control 47(2), 477–491 (1988)
15. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Transactions on Systems, Man, and Cybernetics 15(1), 116–132 (1985)
16. Viard-Gaudin, C., Lallican, P.M., Binter, P., Knerr, S.: The IRESTE On/Off (IRONOFF) dual handwriting database. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, pp. 455–458 (1999)
17. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)

Unsupervised Tagging of Spanish Lyrics Dataset Using Clustering

Fabio Leonardo Parra and Elizabeth León

Universidad Nacional de Colombia, MIDAS group
Bogotá D.C., Colombia
{flparraa,eleonguz}@unal.edu.co

Abstract. In this paper an approach for music clustering, using only lyrics features, is developed for identifying groups with similar feelings, content or emotions in the songs. For this study, a collection of 30.000 Spanish lyrics has been used. The songs were represented in a vector space model (Bag Of Words (BOW)), and some techniques of Part Of Speech (POS) were used as part of preprocessing. Partitional and hierarchical methods were used to perform clustering estimating the appropriate number of clusters (k). For evaluating the clustering results, some internal measures were used such as Davies Bouldin Index (DBI), intra similarity and inter similarity measures. At last, the final clusters were tagged using top words and association rules. Experiments show that music could be organized in related groups and tagged using unsupervised techniques as clustering with only lyrics information.

Keywords: Music Information Retrieval, Clustering, Unsupervised Learning, Feature Selection, Data Mining, Text Mining, Lyrics Analysis, Association Rules.

1 Introduction

Internet has remained in growth for years. It is estimated that between 2011 and 2016, the average growth in consumer internet traffic will be 32% [6]. For taking advantage of this new traffic, it is necessary to organize textual information in websites, offering better search options to users and improving website user experience.

Most of lyrics websites do not have tags in lyrics. However, clustering lyrics information could offer benefits to website users like new search options and recommendations based on previous lyrics visualizations. This could be a starting point for tagging lyrics and then, with the website users, verifying the relevance of those tags.

In a standard web search engine, results are based on search words (query) used by users. In some cases, the results are not relevant. This is because it is possible that the query has different meanings according to the context. To solve this problem, some search engines use clustering algorithms (groupings of the results by proximity or similarity; in the case of text mining, clustering of

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 130–143, 2013.

documents can be by topic), like Carrot[1] or Yippy[2], in order to organize search results. In lyrics websites, clustering results could allow users to find more related songs according to their needs.

Song lyrics are documents with some particularities. They are organized in blocks of chorus and verses; they have rhymes and rhythm and may contain spelling mistakes because they are added by websites users. These characteristics have to be managed in preprocessing step to perform a good classification or clustering.

Most of the studies in MIR that use lyrics have been conducted in classification tasks using audio and lyrics features. Although, there are some studies that use only lyrics information for classification tasks.

This study researches applicability and utility of clustering lyrics without any previous label or tags information using a dataset with 30.000 lyrics in Spanish, comparing results of clustering with Bag Of Words (BOW) features only and using Part Of Speech (POS) features. In order to cluster this dataset, K-means and repeated bisection algorithms are applied evaluating results with Davies Bouldin Index (DBI), intra similarity and inter similarity measures. Finally, top words and association rules with FP-growth algorithm were extracted in every cluster to help to discover cluster tags.

This paper is organized as follows: section 2 presents related works performed in music classification and music clustering; section 3 describes the process applied in the lyrics dataset using unsupervised learning. In this part, preprocessing of data, clustering and evaluation of the results are taken into account. Section 4 attempts to identify tags in clusters. Finally, conclusions and future work are found.

2 Related Work

In music there are not a standardized set of features for organizing repositories. MIR (Music Information Retrieval) researches areas working on obtaining information from music and organizing music repositories automatically. In 2005 MIREX (Music Information Retrieval Evaluation eXchange) was created and it took place during the 6th ISMIR Conference. The goal of MIREX is to compare state-of-the-art algorithms and relevant systems for Music Information Retrieval. Since then, MIR researchers have developed different techniques and have made important contributions to music classification using sound [3]. In recent years, some MIR researches developed techniques using mixing approaches (sound + lyrics) in order to improve previous approaches that used only sound [9,4,10,8,14,13].

MIREX has its own standardization and sound datasets that have been used for different music classification tasks as by genre and mood of the music.

[1] See: `http://search.carrot2.org/stable/search`

[2] See: `http://search.yippy.com`

2.1 Genre of Music

Music genres are not fully defined. These are categories that have arisen through a complex interplay of cultures, artists and market forces to organize music collections and characterize musicians [21]. The boundaries between genres still remain fuzzy, making their separation difficult. Pachet and Cazaly [16] have shown that there is no general agreement on genre taxonomy and it is not easy to build a general taxonomy.

Genre can also be classified in different ways [14], for example in the case of Christmas Carols there is a natural classification based on semantic information, but they may have styles of singing like rock, classical, pop, etc.

However, MIR researches have addressed this problem using pre-defined datasets with genre categories in order to evaluate music classification techniques and compare different approaches on an equal basis [3].

2.2 Mood of Music

The purpose of this approach is to classify music into different emotional categories. In MIR there are no standard mood categories and it is difficult to compare different mood classification researches because, in general, these researches use different mood categories.

Russell's model [19] is the most popular base model of mood that has been used in MIR researches [4,5,9]. This is a dimensional model where emotions are positioned in a multidimensional space. There are two dimensions in it: pleasure and arousal. The original model places 28 emotions in the dimensional space (see figure 1)

2.3 Data Mining Task in Music

In order to organize music in categories, MIR researches have applied different approaches using the info that music provides. There are approaches of supervised learning (classification) and unsupervised learning (clustering) that have shown important results for improving music repositories [21,3,8]. In order to apply these techniques, it is necessary first to make a feature extraction and after it is possible to apply the classification or clustering approach for finally annotating music.

Lyrics Feature Extraction. Lyrics are documents that exhibit specific properties different from traditional text documents. For example lyrics could have rhyming verses, several repetitions, incomplete phrases, spelling mistakes and lyrics are loaded by listeners. However, lyrics could be analysed with classical text information retrieval methods.

Bag-of-words Features. Bag-of-words (BOW) are collections of unordered words. In these, each unique term is regarded a feature and lyrics are represented as feature vectors. In those vectors, each word has a value that represents

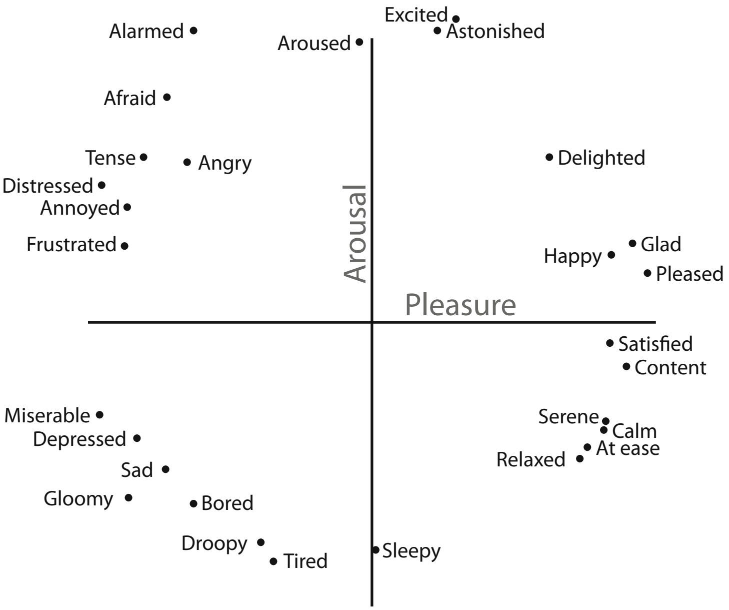


Fig. 1. Adapted form Russell's model of mood [19]

importance of the word in the lyrics. This importance could be frequency of the word, normalized frequency, TF-IDF weight or a boolean value indicating presence or absence of the word in the lyrics. TF-IDF is the most widely used technique in text analysis and MIR.

Additionally, to select the set of words that will make up the BOW is an important task. Tasks like converting all words to the same case folding, removing stopwords and word-stemming could be performed in order to obtain better results.

Part-of-Speech Features. Part-of-speech tagging is a grammatical tagging of words according to a linguistic category. For example nouns, verbs, pronouns, adjectives, etc. POS tagging has been used in lyrics classification tasks with the help of POS tagging software [24].

Rhyme Features. A rhyme is a repetition of similar sounds in two or more words. This similarity could be lexical word endings or with identical or similar ending phonemes. Rhyme is common in songs and has been used in previous lyrics classification tasks [11].

Text Stylistics Features. In lyrics, this refers to special words (e.g., oh, ah, etc.), punctuations and word statistics (e.g., number of unique words, length of words, character frequencies, words per line, etc.). These features have been used in genre identification by Mayer [11].

2.4 Supervised Learning in Music

In supervised learning (classification) the objective is to learn a target function f that maps each attribute set of objects x to one of the predefined class categories y [23].

Music classification could be performed using sound, lyrics or both. Classification with sound is the most widely studied approach, although recently mixed and only lyrics approaches have been studied too. Both cases of music classification are made mainly by genre and mood.

MIR researches have made important progress in this area using sound and lyrics content of music [9,4,10,14]. However, these classification methods could be made by other characteristics like artist, instrument (only for sound), style or similar songs [3].

Classification by genre attempts to identify music genre and is the most widely studied problem in MIR. Mood classification tries to divide music into different emotional categories like angry, sad or happy. Artist identification involves tasks of classifying artist, singer and composer of music. The purpose of instrument recognition is to identify instruments that are played in an interval or segment of the song (made by sound).

2.5 Unsupervised Learning in Music

Unsupervised learning approaches try to assign a set of objects into groups or clusters, so that the objects in the same cluster are more similar to each other than to those in other clusters. These approaches do not use labels or prior knowledge when training the model.

In music, unsupervised approaches have been applied less than supervised. However, there are investigations that attempt to identify genres without prior information [22,2] and there is another attempt that tries to organize lyrics by topic in an unsupervised way [8].

2.6 Music Annotation

Music annotation attempts to assign labels called tags to music that have some meaning to it. This music annotation could be genre, mood, artist, styles, etc. [3]. For Lyrics, music annotation could be performed with help of word frequency in clusters by extracting those words and finding a description that summarizes the cluster [8].

3 Lyrics Clustering Process

There are not enough researches in lyrics clustering. However, lyrics could be worked with document clustering techniques in order to find groups of songs with similar content and to tag them. This tagged groups could allow users to retrieve songs according to the feelings or topic of the song that the user is exploring at the moment.

A standard process, in text mining problems, presents a scheme of steps [15] which may be appropriated for the issue of clustering lyrics. The steps are: pre-processing, clustering, evaluation and topic detection (see figure 2).

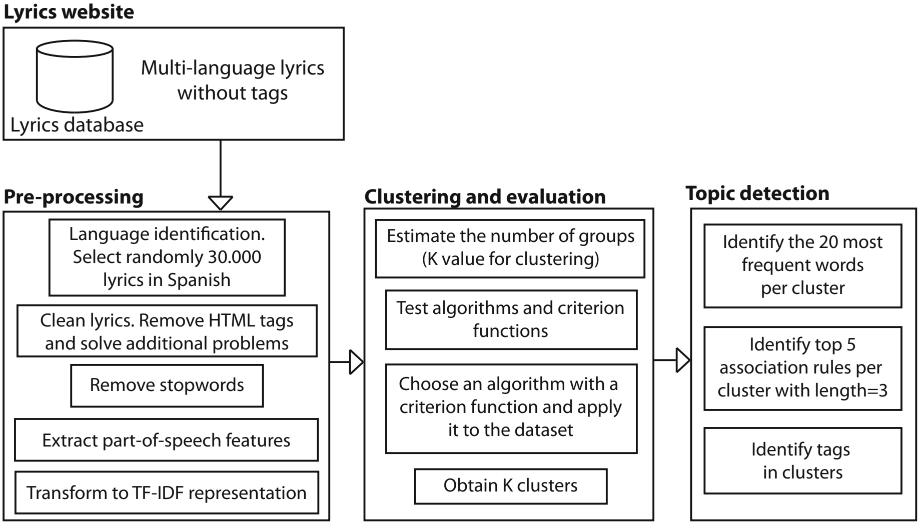

Lyrics website
Lyrics database
Multi-language lyrics without tags
Pre-processing
Language identification. Select randomly 30.000 lyrics in Spanish
Clean lyrics. Remove HTML tags and solve additional problems
Remove stopwords
Extract part-of-speech features
Transform to TF-IDF representation
Clustering and evaluation
Estimate the number of groups (K value for clustering)
Test algorithms and criterion functions
Choose an algorithm with a criterion function and apply it to the dataset
Obtain K clusters
Topic detection
Identify the 20 most frequent words per cluster
Identify top 5 association rules per cluster with length=3
Identify tags in clusters

Fig. 2. Lyrics clustering and tagging process

3.1 Pre-processing

The dataset selected for this investigation was taken from a lyrics website with multi-language lyrics[3]. The first part of the pre-processing consisted in the identification of the language. For identifying lyrics language, a language detection library that detects language using Naïve Bayes and n-grams [12] were used. From this dataset, there were selected randomly 30.000 lyrics with over 99% of probability of being Spanish and with their length over 500 characters.

Lyrics in this dataset are texts in html format and some of them contain mistakes introduced by the website users. For cleaning this dataset and extracting the text, some regular expressions, replacements, spelling corrections and encoding standardization were applied.

This clean dataset in Spanish was divided into two: one with lemmatization using freeling software [17], and the other with the original texts. In these two datasets, the following pre-processing steps were applied:

Term Selection. In order to reduce dimensionality, it is common to remove stopwords and select relevant terms. Stopwords removal was started by a list of standard Spanish stopwords, but adding words like ”coro”, ”estribillo”, ”bis”

[3] See: `http://www.albumcancionyletra.com`

and other words that were found with many repetitions in lyrics dataset and that did not provide additional information in clustering process. Words with two or less characters and words that are not in at least some amount of lyrics were eliminated for preventing spelling mistakes.

TF-IDF Weighting. TF-IDF [20] consists in finding words with high repetition frequency but low frequency in documents. This method allows words, with these characteristics, to be representative of the categories which were being associated. TF-IDF in this lyrics dataset was computed as:

$$tf \times idf(t, d) = tf(d) \times ln(N/df(t)) \tag{1}$$

Where $tf(d)$ is the number of times that term t appears in document d and $df(t)$ is the number of documents in the collection that term t occurs in.

3.2 Clustering and Evaluation

Previous results in document clustering have shown that hierarchical algorithms like partitional clustering algorithms are well-suited for clustering large document datasets due to their relative low computational requirements [25]. For this work K-means and Repeated Bisection algoritms were used and to achieve comparable results, the same validation measures were applied.

K-means is the most popular partitional algorithm and has been used widely to compare different results of clustering. For this algorithm a standard implementation was used. For Repeated Bisection algorithms CLUTO software[4] has an implementation in it.

Detecting the amount of clusters is not an easy task. Using small values of K, results are clusters with mixtures of multiple topics and increasing K, validation measures perform better [8] but there is not an optimal K value (see figure 3). For an approximation of K, Pham et al technique [18] was used and the K value was chosen where the graph starts stabilization (around K=20).

With this K, K-means and repeated bisections algorithms were applied and compared with the standard K-means results. Cluto has two implementations of repeated bisections: RB and RBR. The main difference is that RBR applies an additional local optimization that RB does not apply. In RB and RBR, it is possible to apply some optimization functions [7] (see figure 4).

K-means changes results in every execution because centroids were selected randomly. To obtain this result, K-means algorithm was applied 36 times and mean and standard deviation of the results were calculated.

Results with original dataset using DBI are in table 1 and results using Inter/Intra similarity, weighted by amount of objects in clusters, are in table 2.

RBR was the algorithm selected for clustering lyrics. This algorithm presents better results in most cases. POS dataset performs better in the majority of cases and I_2 was the criterion function that offers better results in RBR (POS) execution.

[4] See: `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview`

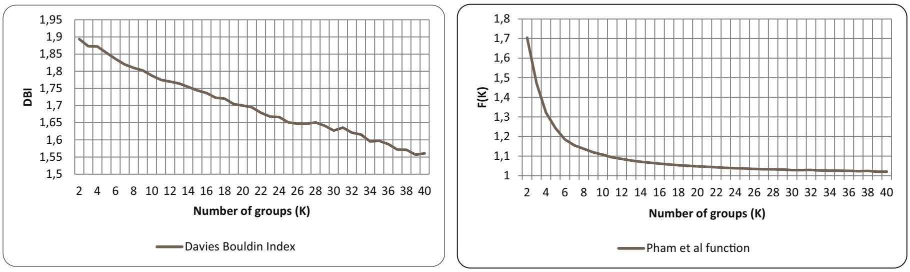


Fig. 3. K estimation for Repeated Bisection using DBI and Pham et al function

Criterion Function	Optimazition Function
$\mathcal{I}_1$	maximize $\sum_{i=1}^{k} \frac{1}{n_i} \left(\sum_{v,u \in S_i} \text{sim}(v,u) \right)$ (1)
$\mathcal{I}_2$	maximize $\sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} \text{sim}(v,u)}$ (2)
$\mathcal{E}_1$	minimize $\sum_{i=1}^{k} n_i \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sqrt{\sum_{v,u \in S_i} \text{sim}(v,u)}}$ (3)
$\mathcal{G}_1$	minimize $\sum_{i=1}^{k} \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sum_{v,u \in S_i} \text{sim}(v,u)}$ (4)
$\mathcal{G}_1'$	minimize $\sum_{i=1}^{k} n_i^2 \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sum_{v,u \in S_i} \text{sim}(v,u)}$ (5)

Fig. 4. Cluto criterion functions adapted from [7]

Table 1. Clustering validation results with K = 20 using DBI (less is better)

	Max-I_1	**Max-I_2**	**Min-E_1**	**Min-G_1**	**Min-G_1'**
RB	1,8387	1,8855	1,9169	1,8844	1,9088
RBR	1,8648	1,8850	1,8869	1,8583	1,8860
RB (POS)	1,9190	1,9176	1,9221	1,9232	1,9202
RBR (POS)	1,9177	1,8885	1,8864	1,9045	1,8799

K-Means: 1,9257 +/- 0.007; **K-Means (POS)**: 1,9258 +/- 0.012

Table 2. Clustering validation results with K = 20 using Inter similarity/Intra similarity (more is better)

	Max-I_1	**Max-I_2**	**Min-E_1**	**Min-G_1**	**Min-G_1'**
RB	1,1024	1,1010	1,0881	1,0929	1,0910
RBR	1,1101	1,1085	1,0971	1,1091	1,0997
RB (POS)	1,1130	1,1142	1,1033	1,1044	1,1063
RBR (POS)	1,1249	**1,1253**	1,1177	1,1198	1,1194

K-Means: 1,1024 +/- 0.002; **K-Means (POS)**: 1,1218 +/- 0.003

K-means performance was useful, but this algorithm creates groups with similar amount of objects and for some of them it is difficult to identify tags. Repeated bisections algorithm creates several groups with different length, and some of them were well separated and easy to tag.

4 Topic Detection

It is intended to find clusters tags for groups found through categorization techniques. Only grouping is not enough in lyrics categorization. It is necessary to show this to the user, for example with tags. In order to achieve this, the majority of the techniques use the set of most frequent words in each of the groups to make an approximation to the topic [1].

This set of words provides information that is useful to identify cluster's tags. In this case, the 20 most frequently words and the top 5 association rules were identified. For discovering association rules the FP-Growth algorithm was used with min length = 3 indicating support value for every rule (see table 3).

Table 3. Cluster results with K = 20 using RBR(POS) and I2 as criterion function

Cluster Id	**Top Words in TF-IDF**	**Top Itemsets**	**Tags**
Cluster 1 (Size: 731, Inter:0,2688, Intra:0,1039)	enamorar, amor, estar, corazón, querer, haber, decir, mujer, amar, tener, vida, poder, solo, ese, saber, ver, niño, dar, tanto, amigo.	enamorar,amor,querer (0.438) enamorar,querer,estar (0.409) enamorar,amor,estar (0.379) enamorar,querer,tener (0.372) enamorar,amor,corazon (0.360)	In love, Love, Heart
Cluster 2 (Size: 750, Inter:0,2629, Intra:0,1077)	adiós, doler, amor, decir, dolor, corazón, haber, amar, llorar, nuestro, aunque, tanto, querer, perder, saber, poder, alma, quedar, vida, dejar.	adios,amor,decir (0.375) adios,amor,haber (0.356) adios,amor,querer (0.332) adios,decir,haber (0.309) adios,amor,poder (0.304)	Good bye, Love, Pain
Cluster 3 (Size: 756, Inter:0,2437, Intra:0,0830)	vos, sos, tic, tac, estar, tenes, querer, co, queres, poder, corazón, solo, amor, ver, dar, saber, tener, siempre, pasar, ese.	vos,estar,querer (0.283) vos,estar,poder (0.279) vos,estar,tener (0.254) vos,querer,poder (0.238) vos,poder,tener (0.221)	Not identified
Cluster 4 (Size: 1259, Inter:0,2548, Intra:0,1143)	amar, amor, querer, decir, nadie, vida, corazón, saber, amarar, solo, haber, tanto, poder, tener, mujer, dar, dejar, ese, estar, siempre.	amar,amor,querer (0.459) amar,amor,haber (0.372) amar,amor,tener (0.357) amar,querer,haber (0.351) amar,amor,poder (0.345)	Love

Table 3. *(Continued)*

Cluster Id	**Top Words in TF-IDF**	**Top Itemsets**	**Tags**
Cluster 5 (Size: 852, Inter:0,2489, Intra:0,1114)	contigo, conmigo, querer, estar, amor, tener, vivir, solo, vida, decir, poder, ver, dar, haber, siempre, sentar, noche, saber, amar, quedar.	contigo,querer,estar (0.488) contigo,querer,amor (0.435) contigo,querer,tener (0.408) contigo,querer,poder (0.393) contigo,estar,amor (0.393)	Love, Declaration of love
Cluster 6 (Size: 1215, Inter:0,2447, Intra:0,1138)	olvidar, poder, amor, recordar, querer, amar, dejar, decir, haber, aunque, pensar, tanto, vida, corazón, saber, volver, recuerdo, estar, seguir, vivir.	olvidar,querer,amor (0.403) olvidar,querer,poder (0.390) olvidar,amor,poder (0.355) olvidar,querer,haber (0.347) olvidar,querer,estar (0.328)	Love, Obscurity, Memories
Cluster 7 (Size: 1049, Inter:0,2395, Intra:0,1041)	llorar, sufrir, amor, querer, pena, corazón, dolor, haber, solo, dejar, decir, saber, ver, lagrimar, estar, vida, amar, olvidar, pensar, morir.	llorar,querer,amor (0.365) llorar,querer,haber (0.350) llorar,amor,haber (0.333) llorar,querer,estar (0.285) llorar,querer,ver (0.284)	Pain, Tears, Love
Cluster 8 (Size: 977, Inter:0,2199, Intra:0,0626)	bailar, mover, ritmo, gozar, pa, menear, cumbia, baile, fiesta, gustar, pegao, cuerpo, colita, cheque, sua, poner, cintura, querer, ver, ese.	bailar,querer,ver (0.288) bailar,querer,tener (0.275) bailar,querer,dar (0.247) bailar,querer,ese (0.233) bailar,tener,ver (0.230)	Dance
Cluster 9 (Size: 1121, Inter:0,2369, Intra:0,1118)	volver, querer, ver, amor, poder, haber, estar, saber, dejar, vida, decir, vivir, esperar, perder, sentir, día, tener, solo, pensar, pasar.	volver,querer,poder (0.354) volver,querer,haber (0.344) volver,querer,estar (0.331) volver,querer,ver (0.329) volver,querer,tener (0.320)	Love, Reconciliation
Cluster 10 (Size: 706, Inter:0,2194, Intra:0,0817)	loco, gustar, na, querer, estar, ese, decir, tener, dar, volver, amor, zancudo, haber, saber, ver, nene, mirar, solo, corazón, dejar.	querer,gustar,tener (0.215) querer,loco,estar (0.201) querer,tener,haber (0.200) querer,tener,loco (0.194) querer,tener,decir (0.191)	Not identified
Cluster 11 (Size: 2098, Inter:0,2046, Intra:0,1102)	amor, corazón, querer, morir, vida, dar, solo, dolor, haber, poder, tener, vivir, saber, decir, sentar, estar, tanto, amar, nuestro, siempre.	amor,querer,corazon (0.380) amor,querer,haber (0.343) amor,querer,tener (0.331) amor,corazon,haber (0.326) amor,querer,poder (0.323)	Love, Heart
Cluster 12 (Size: 947, Inter:0,1835, Intra:0,0724)	dios, señor, usted, za, jesús, santo, cristo, gloria, don, gracia, haber, dar, poder, alabar, bendecir, vida, amor, padre, hijo, estar.	dios,haber,estar (0.189) dios,haber,tener (0.187) dios,haber,poder (0.183) dios,haber,querer (0.182) haber,querer,tener (0.172)	God, Religion

Table 3. *(Continued)*

Cluster Id	Top Words in TF-IDF	Top Itemsets	Tags
Cluster 13 (Size: 1130, Inter:0,1773, Intra:0,0815)	cantar, canción, luna, flor, maría, querer, noche, corazón, amor, estrella, canto, haber, sol, vida, voz, bello, dar, ese, tener, mirar.	cantar,querer,haber (0.197) cantar,querer,amor (0.184) cantar,querer,cancion (0.183) cantar,querer,tener (0.176) cantar,cancion,haber (0.168)	Sing
Cluster 14 (Size: 1552, Inter:0,1673, Intra:0,0660)	pa, mami, dar, gato, poner, eh, hey, ma, vamo, tra, tener, papi, nene, querer, ese, meter, ver, tirar, duro, gustar.	tener,querer,dar (0.385) tener,querer,ver (0.348) tener,querer,decir (0.350) tener,querer,ese (0.335) tener,querer,haber (0.303)	Not identified
Cluster 15 (Size: 892, Inter:0,1680, Intra:0,0679)	rap, mierda, os, puta, bla, rock, hop, hip, estilo, rima, perro, tener, ese, pa, mc, culo, roll, haber, dar, joder.	tener,haber,estar (0.352) tener,haber,ver (0.343) tener,haber,dar (0.339) tener,estar,dar (0.332) tener,haber,querer (0.330)	Rap
Cluster 16 (Size: 1974, Inter:0,1789, Intra:0,0953)	beso, cuerpo, besar, piel, noche, labio, boca, querer, mujer, amor, ah, dar, fuego, ojo, sentir, solo, tener, mio, ese, sentar.	querer,amor,tener (0.216) querer,amor,beso (0.209) querer,amor,haber (0.188) querer,amor,estar (0.188) querer,amor,dar (0.187)	Love, Kiss
Cluster 17 (Size: 2987, Inter:0,1817, Intra:0,1048)	decir, poder, querer, saber, pensar, haber, algo, nadie, estar, entender, solo, igual, ver, hablar, mejor, dejar, tener, importar, cambiar, mal.	querer,decir,poder (0.304) querer,decir,haber (0.291) querer,decir,estar (0.282) querer,decir,tener (0.281) querer,poder,haber (0.279)	Not identified
Cluster 18 (Size: 3252, Inter:0,1799, Intra:0,1066)	solo, estar, sueño, vida, quedar, poder, vivir, haber, encontrar, día, siempre, perder, esperar, tanto, ver, seguir, querer, sentar, pensar, junto.	estar,poder,querer (0.232) haber,estar,poder (0.229) haber,estar,querer (0.227) estar,poder,solo (0.222) haber,estar,solo (0.220)	Loneliness
Cluster 19 (Size: 2248, Inter:0,1654, Intra:0,0837)	volar, sol, mar, luz, cielo, azul, estrella, viento, noche, agua, ver, tierra, ojo, brillar, poder, amor, luna, haber, sueño, querer.	haber,ver,poder (0.118) querer,ver,poder (0.115) haber,querer,ver (0.112) haber,querer,poder (0.110) haber,querer,amor (0.109)	Firmament, Sky
Cluster 20 (Size: 3504, Inter:0,1323, Intra:0,0725)	haber, amigo, tener, bueno, gente, malo, ese, niño, calle, matar, dar, hombre, venir, estar, mucho, decir, pasar, hijo, madre, viejo.	haber,tener,querer (0.219) haber,tener,estar (0.210) haber,tener,ver (0.203) haber,tener,decir (0.203) haber,tener,poder (0.197)	Not identified

These results with top words and association rules help in the process of tagging but the tags depends on the person that performs tagging process. In this case, using 20 clusters was possible to identify tags for 15 of them in Spanish.

5 Conclusion and Future Work

Music categorization is an actual trend with wide researches in recent years. Those researches have focused their efforts on classifying datasets mainly composed by sound, but there are cases when they use lyrics too.

Efforts have focused primarily on two branches: genre and mood. When using genre, lyrics do not provide much information, whereas in the case of mood, the lyrics present larger contributions. In this work it was found that some specific topics, not related with mood or genre, could be discovered too.

The feature selection stands as one of the main points in the whole mining process. Dataset using POS reduces dimensionality and improves results.

Internal measures like Davies Bouldin Index (DBI) estimate clustering quality using averages of worst case scenarios. In the case of lyrics, some clusters are compact and well separated, but others are not. Inter similarity / Intra similarity offer better results for choosing the best option between different algorithms.

Repeated bisections with optimization (RBR) was the algorithm that offers better results in most cases. Additionally, with this algorithm it is possible to bisect clusters in which it is not possible to identify tags.

Our model takes into account all steps in lyrics clustering process, to finally discover topics and assign tags in some clusters. This model shows how it is possible to apply text mining techniques in a problem from the real world with a large amount of data.

Detecting the appropriate number of groups K is not an easy task. This happens because groups could be divided into several different topic groups (like genre, artist, mood, sentiment, topic, etc) with different boundaries. In our results some tags were genre others - moods and topics. This model helps in the process of tagging lyrics but has not given a complete set of tags for music.

The experiment results show how our analysis gives initial results in music tagging. Using 20 clusters allows to identify tags for 15 of them using top words and association rules. In other cases, it could be possible that those groups need more splits for performing tag annotation.

Future work could apply additional clustering techniques in depth like fuzzy techniques, test other datasets in other languages and improve K detection and tagging process.

References

1. Anaya-Sánchez, H., Pons-Porrata, A., Berlanga-Llavori, R.: A document clustering algorithm for discovering and describing topics. Pattern Recogn. Lett. 31(6), 502–510 (2010)
2. Barreira, L., Cavaco, S., Da Silva, J.: Unsupervised music genre classification with a model-based approach. In: Antunes, L., Pinto, H.S. (eds.) EPIA 2011. LNCS, vol. 7026, pp. 268–281. Springer, Heidelberg (2011)
3. Fu, Z., Lu, G., Ting, K.M., Zhang, D.: A survey of audio-based music classification and annotation. IEEE Transactions on Multimedia 13(2), 303–319 (2011)

4. Hu, X., Downie, J.S.: Improving mood classification in music digital libraries by combining lyrics and audio. In: Proceedings of the 10th Annual Joint Conference on Digital Libraries, JCDL 2010, New York, NY, USA, pp. 159–168 (2010)
5. Hu, Y., Chen, X., Yang, D.: Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. In: Proceedings of ISMIR 2009, pp. 123–128 (2009)
6. Inc., C.: Cisco visual networking index: Forecast and methodology, 2011-2016. Tech. rep., Cisco (2012)
7. Karypis, G.: Cluto a clustering toolkit. Tech. Rep. 02-017, Dept. of Computer Science, University of Minnesota (2003), `http://www.cs.umn.edu/~cluto`
8. Kleedorfer, F., Knees, P., Pohle, T.: Oh oh oh whoah! towards automatic topic detection in song lyrics. In: Bello, J.P., Chew, E., Turnbull, D. (eds.) ISMIR, pp. 287–292 (2008)
9. Laurier, C., Grivolla, J., Herrera, P.: Multimodal music mood classification using audio and lyrics. In: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, ICMLA 2008, pp. 688–693. IEEE Computer Society, Washington, DC (2008)
10. Li, T., Ogihara, M., Zhu, S.: Integrating features from different sources for music information retrieval. In: IEEE International Conference on Data Mining, pp. 372–381 (2006)
11. Mayer, R., Neumayer, R., Rauber, A.: Rhyme and style features for musical genre classification by song lyrics. In: Proceedings of the 9th International Conference on Music Information Retrieval (2008)
12. Nakatani, S.: Language detection library for java. Tech. rep., Cybozu Labs, Inc. (2011), `http://code.google.com/p/language-detection/`
13. Neumayer, R., Rauber, A.: Integration of text and audio features for genre classification in music information retrieval. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 724–727. Springer, Heidelberg (2007)
14. Neumayer, R., Rauber, A.: Multi-modal music information retrieval - visualisation and evaluation of clusterings by both audio and lyrics. In: Proceedings of the 8th Conference Recherche d'Information Assiste Par Ordinateur, RIAO 2007. ACM (2007)
15. Ozgur, A.: Supervised and Unsupervised Machine Learning Techniques For Text Document Categorization. Master's thesis, Department of Computer Engineering, Bogazici University, Istanbul, Turkey (2002)
16. Pachet, F., Cazaly, D.: A taxonomy of musical genres. In: Proc. Content-Based Multimedia Information Access, RIAO 2000 (2000)
17. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the Language Resources and Evaluation Conference, LREC 2012. ELRA, Istanbul (2012)
18. Pham, D.T., Dimov, S.S.N.C.D.: Selection of k in k -means clustering. In: Proceedings of the Institution of Mechanical Engineers, vol. 219, p. 103 (2005)
19. Russell, J.A.: A circumplex model of affect. Journal of Personality and Social Psychology 39, 1161–1178 (1980)
20. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24(5), 513–523 (1988), cited By (since 1996) 1952
21. Scaringella, N., Zoia, G., Mlynek, D.: Automatic genre classification of music content: a survey. IEEE Signal Processing Magazine 23(2), 133–141 (2006)

22. Shao, X., Xu, C., Kankanhalli, M.: Unsupervised classification of music genre using hidden markov model. In: 2004 IEEE International Conference on Multimedia and Expo, ICME 2004, vol. 3, pp. 2023–2026 (2004)
23. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, 1st edn. Addison Wesley (May 2005)
24. Ying, T.C., Doraisamy, S., Abdullah, L.: Genre and mood classification using lyric features. In: 2012 International Conference on Information Retrieval Knowledge Management, CAMP, pp. 260–263 (March 2012)
25. Zhao, Y., Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. Mach. Learn. 55(3), 311–331 (2004)

Feature Learning for Detection and Prediction of Freezing of Gait in Parkinson's Disease

Sinziana Mazilu[1], Alberto Calatroni[1], Eran Gazit[2], Daniel Roggen[1], Jeffrey M. Hausdorff[2], and Gerhard Tröster[1]

[1] Wearable Computing Laboratory, ETH Zürich, Switzerland
{sinziana.mazilu,alberto.calatroni,daniel.roggen,troester}@ife.ee.ethz.ch
[2] Laboratory of Gait and Neurodynamics, Tel Aviv Sourasky Medical Center
{erang,jhausdor}@tasmc.health.gov.il

Abstract. Freezing of gait (FoG) is a common gait impairment among patients with advanced Parkinson's disease. FoG is associated with falls and negatively impact the patient's quality of life. Wearable systems that detect FoG have been developed to help patients resume walking by means of auditory cueing. However, current methods for automated detection are not yet ideal. In this paper, we first compare feature learning approaches based on time-domain and statistical features to unsupervised ones based on principal components analysis. The latter systematically outperforms the former and also the standard in the field – Freezing Index by up to 8.1% in terms of F1-measure for FoG detection.

We go a step further by analyzing FoG prediction, i.e., identification of patterns (pre-FoG) occurring before FoG episodes, based only on motion data. Until now this was only attempted using electroencephalography. With respect to the three-class problem (FoG vs. pre-FoG vs. normal locomotion), we show that FoG prediction performance is highly patient-dependent, reaching an F1-measure of 56% in the pre-FoG class for patients who exhibit enough gait degradation before FoG.

Keywords: Unsupervised feature learning, Freezing of Gait, Parkinson's disease.

1 Introduction

Freezing of gait (FoG) is a common gait impairment among patients with Parkinson's disease (PD), defined as a "brief, episodic absence or marked reduction of forward progression of the feet despite the intention to walk" [18]. Patients describe FoG as the feeling of having the feet glued to the ground and being temporarily unable to re-initiate gait. According to a survey of 6620 PD patients, 47% of the subjects reported regular freezing and 28% experienced FoG daily [13]. FoG is associated with falls [12], has substantial clinical and social consequences [6, 15] and is often resistant to pharmacological treatment [2].

Rhythmic auditory stimulation (RAS) was introduced as an assistive tool for FoG treatment [8]. RAS can be applied to produce a rhythmic ticking sound upon

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 144–158, 2013.

detection of a FoG episode, to help the patient resume walking. Wearable systems based on motion sensors have been proposed for the detection and treatment of FoG with auditory stimulation [1, 11]. While RAS upon detection helps to shorten the duration of FoG episodes [1], it cannot avoid them altogether due to the latency of the detection, which is at best on the order of hundreds of milliseconds [11]. A step further is to predict when a patient is *about to* experience FoG, thus enabling preemptive RAS, with the goal of avoiding the FoG episodes. We call this *FoG prediction* as opposed to *FoG detection.*

There are some known specific properties that differentiate the sensor data during FoG episodes from normal walking (e.g., a large increase in the signal energy in the 3-8Hz frequency band [9,15]) and the gait of patients with FoG also differs between freezing episodes, compared to patients who do not experience FoG [10]. There are even suggestions of a characteristic change in the gait pattern just prior to the occurrence of a FoG episode; however, currently, there is no way of automatically identifying the prodromal state, when the normal gait pattern is about to transform into FoG.

The lack of physiological understanding of the gait deterioration preceding FoG makes it difficult to come up with a model or with problem-specific features based on expert knowledge. Moreover, walking styles of PD patients differ across subjects (including diverse motor anomalies) [17]. Thus eventual patterns in the data just before a FoG event will also likely be highly subject-specific. Nevertheless, previous work suggests that there is a deterioration of the normal gait before FoG, although this deterioration can be expressed in various ways [17–19].

In this work, we first formulate the *FoG detection* problem as a two-class classification problem: FoG versus normal locomotion. Similarly, we treat *FoG prediction* problem as a three class classification problem. Beside FoG and normal locomotion, we consider the walking periods before FoG episodes as a third class called *pre-FoG*. We hypothesize that there is a detectable deterioration of gait in this phase which precedes FoG. We assume different durations of the pre-FoG events, since these cannot be labeled by an expert, but can rather only be retrieved through data mining from segments of data preceding FoG events. We focus on the analysis of different feature extraction approaches that lead to a meaningful representation of both the FoG and the new pre-FoG class. The feature extraction approaches that we investigated are the following:

(a) Extraction of standard frequency-based features, namely Freezing Index and total energy in the frequency band 0.5-8 Hz. This is the current standard in the field and serves as a baseline [15].

(b) Extraction of various *hand-crafted* time-domain and statistical features, which are used in pattern recognition problems involving motion or human activity recognition.

(c) Unsupervised feature learning [20]. This method involves extraction of information from the raw data, without relying on domain specific knowledge, or on the availability of ground truth annotations. We evaluate the use of principal component analysis for extracting a compact representation of the structure of the signals.

The contributions of this work are summarized as follows:

1. We go beyond state of the art by explicitly introducing and performing a first step toward FoG prediction, as opposed to mere detection, thereby potentially allowing for the possibility of applying preemptive RAS;
2. We compare three methods for feature extraction in the FoG detection and FoG prediction problems and show that unsupervised feature learning outperforms on average standard feature extraction schemes in our real-life dataset;
3. We show that removal of pre-FoG sequences from the training data for FoG detection improves classification performance;
4. For FoG prediction, we show that, for some patients, gait anomalies associated with the upcoming onset of FoG can be detected, thereby allowing for an early intervention with RAS.

2 Related Work

FoG Detection. Several research groups have proposed wearable systems for the detection of FoG episodes [1, 4, 5, 7, 11, 14–16, 21, 23]. Most sensor setups involve accelerometers and/or gyroscopes [1, 5, 11, 16, 21], extended with electroencephalography (EEG) [7] or electromyography (EMG) [4]. One standard feature which is extracted from the raw signals is the Freezing Index (FI), defined as the ratio between the power contained in the so-called *freezing* and *locomotion* frequency bands (3-8 Hz and 0.5-3 Hz respectively) [1, 11, 15]. This feature is convenient since it requires only FFT-computation. Other feature extraction approaches involve mixed time-frequency features [23] and entropy [21]. In [14], the authors investigated the use of time-domain and statistical features, together with FFT-features. Various classifiers have been used for the two-class classification problem (FoG versus no-FoG), including Decision Trees, Random Trees/Forests, Naive Bayes [21] as well as rule-based classifiers [5] and simple thresholds on the FI [1]. Overall, the different proposed approaches reach detection sensitivities that often exceed 80%, but the detection is performed with at best a latency of a few hundred milliseconds. Handojoseno and colleagues [7] make use of wavelet decomposition to analyze the dynamics of EEG signals during the onset and the freezing periods. Their aim was to achieve an early detection of FoG from brain activity that could, potentially, help patients to avoid an impending FoG episode. To our best knowledge, no attempts have been yet made at tackling the FoG prediction problem using just motion sensors. We therefore perform a first analysis in this direction.

Unsupervised Feature Learning. Automatic (unsupervised) feature extraction has been proposed in the context of human activity recognition based on motion sensors. Plötz et al. [20] argued that instead of using the explicit knowledge to select specific features, one can extract the core signal characteristics by means of principal components analysis. This allows one to uncover meaningful, low-dimensional representations of raw data without relying on domain-specific knowledge. The results on public activity recognition datasets showed that the features learned in this unsupervised manner are more discriminative than state of the art representations based on time- and frequency-domain features. We propose to apply this method for detection and prediction of FoG, since the properties of the FoG and pre-FoG signals are subject-dependent and difficult to model.

3 Feature Extraction for FoG Detection and Prediction

The general process that we adopt for signal processing and classification is depicted in Figure 1. The set of operations is standard in pattern recognition problems involving motion data from on-body 3-dimensional accelerometers: sensor signals are sampled and sliced into partially overlapping windows. In each window, features are extracted and the resulting vectors are classified according to a pretrained model. In this work, we empirically set the window length to $1s$ (64 samples) with $0.25s$ of overlap (16 samples). We choose a Decision Tree classifier, because of its low computational cost when deployed. In this work, we focus on the selection of the appropriate features for detection and prediction of FoG, so the optimization of the classifier parameters is out of our scope. In the training phase of the system, we use feature ranking based on Mutual Information (MI) to rank the top discriminant time-based and statistical features [3]. We denote with N_F the number of top-ranked features retained in the classification process.

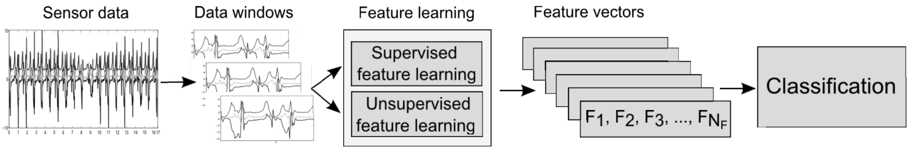


Fig. 1. Signal processing and classification for the detection and prediction of FoG

3.1 Feature Extraction Schemes

We choose three groups of features, the first of which has been already used in the context of FoG detection and is used here as a baseline. We call *supervised* the first two feature extraction approaches, since they involve features manually selected due to expert knowledge. The features are computed for each window.

Supervised: Domain-specific Feature Extraction. The first feature group contains the Freezing Index and the sum of energy in the freezing (3-8 Hz) and locomotory (0.5-3 Hz) frequency bands. These features are obtained by computing the FFT, followed by binning, in order to compute the spectral distribution of the energy in the desired bands.

Supervised: Feature Extraction of Time-domain and Statistical Features. The second group of features is often used in activity recognition [22]. Until now, only a small subset of these has been also applied to FoG detection [14]. We list the used features in Table 1. We extracted 18 features for each of the three accelerometer axes (x, y, z) and six features using data from all three axes. **Unsupervised Feature Learning.** For learning the implicit structure of the data, each data window containing 64 samples for the three accelerometer axes is arranged into a 192-dimensional vector (the first three entries correspond to the first samples from the x, y and z axes, and so on). In the training phase, principal component analysis (PCA) is then applied to the whole training data matrix, obtained by stacking all the 192-dimensional vectors in the training set and disregarding class labels. This yields a projection matrix, which is then used in the testing phase to project the single data frames.

Table 1. Computed statistical features and their brief descriptions

	Axis Features	
No.	**Feature**	**Description**
1,2	Min, Max	Minimum and maximum of the signal
3	Median	Median signal value
4,5	Mean, ArmMean	Average value, and the harmonic average of the signal
6	Root Mean Square (RMS)	Quadratic mean value of the signal
7	GeoMean	Geometric average of the signal
8	Variance	Square of the standard deviation
9	Standard Deviation (STD)	Mean deviation of the signal compared to the average
10	Kurtosis	The degree of peakedness of the sensor signal distribution
11	Skewness	The degree of asymmetry of the sensor signal distribution
12	Mode	The number that appears most often in the signal
13	TrimMean	Trimmed mean of the signal in the window
14	Entropy	Measure of the distribution of frequency components
15	Asymmetry coefficient	The first moment of the data in the window divided by STD over the window
16	Range	The difference between the largest and smallest values of the signal
17	Zero Crossing Rate (ZCR)	Total number of times the signal changes from positive to negative or back, normalized by the window length
18	Mean Crossing Rate (MCR)	Total number of times the signal changes from below average to above average, normalized by the window length

	Sensor Features	
No.	**Feature**	**Description**
55	Signal Magnitude Vector (SMV)	Sum of the euclidean norm over the three axis over the entire window normalized by the window length
56	Normalized Signal Magnitude Area (SMA)	Acceleration magnitude summed over three axes normalized by the window length
57,58,59	Eigenvalues of Dominant Directions (EVA)	Eigenvalues of the covariance matrix of the acceleration data along x, y, and z axis
60	Averaged Acceleration Energy (AAE)	Mean value of the energy over three acceleration axes

3.2 Assumptions about pre-FoG Events: From FoG Detection to Prediction

We assume that the gait cannot enter in the FoG state directly from walking state. Rather, we assume that prior to FoG, there is a gait deterioration that eventually leads to the FoG. This is represented by a transition period of variable duration T_{Prefog} that we refer to as the pre-FoG state. An example of FoG episode, with supposed pre-FoG, is shown in Figure 2. The optimal value of T_{Prefog} will be patient-dependent. The identification of segments of pre-FoG data is valuable both for FoG detection and prediction. **Making Detection More Robust.** For the detection problem, we set up a two-class classification problem. We name the two classes WALK (which includes instances of normal locomotion, including walking, standing, turning, etc.) and FoG, which represents the freezing episodes. In the training phase, we remove data for a duration of T_{Prefog} before each FoG event contained in the training set. This aims at having a more precise classifier model for the FoG and for the WALK classes.

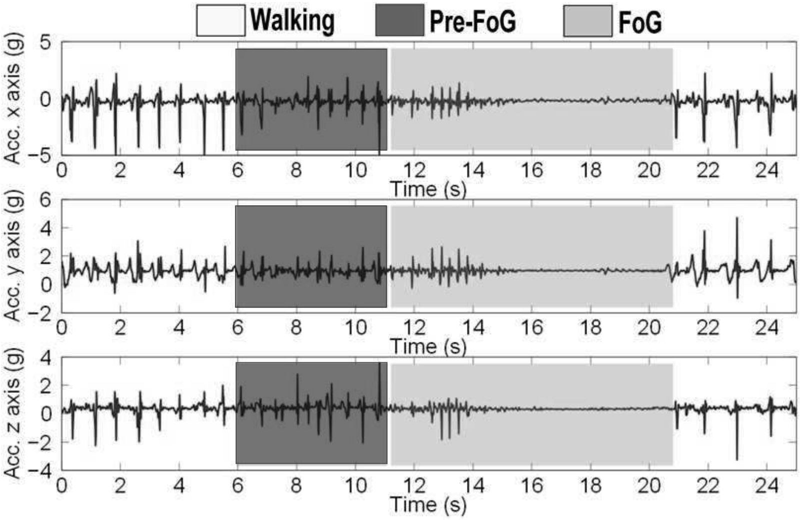


Fig. 2. An example of an accelerometer signal, on the three acceleration axes, that captures the motor variations in the gait of a patient with Parkinson's disease. The sequence contains normal gait, a FoG episode, preceded by a assumed pre-FoG period.

Towards FoG Prediction. For prediction, we set up a three-class classification problem. Besides the two classes described above (WALK and FoG), we use the segments assumed to be in a pre-FoG state to build the model for the third class.

4 Dataset

We validated the proposed approach on the public available DAPHNet dataset[1] [1], which contains data collected from eight PD patients that experienced regular FoG in daily life. Data were recorded using three 3D accelerometers attached to the shank (above the ankle), the thigh (above the knee) and to the lower back of each subject. For our experiments here we focused on movement data recorded from the ankle, as the data from the other two sensors generally behave similarly. Subjects completed sessions of 20-30 minutes each, consisting of three walking tasks: (1) Walking back and forth in a straight line, including several 180-degrees turns; (2) Random walking with a series of initiated stops and 360 degrees turns; (3) Walking simulating activities of daily living, which included entering and leaving rooms, walking to the kitchen, getting something to drink and returning to the starting room with a cup of water.

Motor performances varied strongly among the participants. While some subjects maintained regular gait during nonfreezing episodes, others walked slowly and were very unstable. The DAPHnet dataset contains 237 FoG episodes; the duration of FoG episodes is between 0.5s and 40.5s (7.3±6.7s). 50% of the FoGs lasted for less than 5.4s and 93.2% were shorter than 20s. FoGs were labeled by physiotherapists using synchronized video recordings. The start of a FoG event was defined as the point when the gait pattern (i.e., alternating left-right stepping) was arrested, and the end of a FoG was defined as the point in time when the pattern was resumed.

5 Experiments and Evaluation

We performed two sets of experiments using the DAPHnet dataset described in Section 4: one for *FoG detection* and one for *FoG prediction*. For FoG-detection we ignored the pre-FoG sequences. For both sets of experiments and for two of the three groups of features introduced in Section 3.1, we varied the number of selected features N_F from 5 to 60 in steps of 5. This cannot be done for the domain-specific features, since they are only two - FI and total energy. We further characterized the influence of different choices of the pre-FoG duration on both the two-class and three-class problem, by sweeping the assumed pre-FoG duration in the range $T_{Prefog} \in \{1s, 2s, ..., 11s\}$.

The evaluation was performed on a patient-dependent basis. Since in each patient dataset the WALK class was over-represented compared to the FoG class, we chose to balance the data by having $size(\text{WALK}) = X * size(\text{FoG})$, where $X \in \{1.5, 2, ..., 10\}$. We performed an $N = 10$-fold cross validation, in which the training data contains N-1 parts from the FoG data, N-1 parts from normal locomotion data, and the testing data the rest. The data were split for each fold in such a way as to avoid having time-correlated chunks of the same FoG, WALK, or pre-FoG events in the training and testing data.

[1] `www.wearable.ethz.ch/resources/Dataset`

We report results in terms of overall patient datasets average sensitivity and average specificity of the FoG class, and F1-measures for FoG, WALK and pre-FoG classes, in a window-to-window comparison.

6 Results

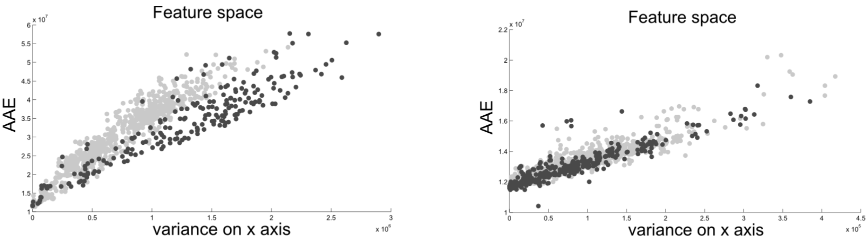
In the following, we analyze the performance of the different feature extraction strategies for the FoG-detection and FoG-prediction problems.

6.1 Time-Domain and Statistical Features

The top ranked features based on MI for FoG detection are AAE, eigenvalues of dominant directions, range, variance, root mean square, and standard deviation (some features, like standard deviation and variance are of course strictly related). The top ranked features according to their MI are those computed from the entire data window (all axes), followed by those on x-axis. Table 2 shows the top k ranked features, for k ranging from 5 to 20. Note that selecting features that are ranked highly by the MI does not automatically guarantee that they are also discriminative enough. Figure 3 contains an example of distribution of the top ranked features AAE and variance on x-axis. For some patients these features are enough to distinguish between FoG and WALK – the two classes

Table 2. Average top ranked features with Mutual Information

Top k	x axis	y axis	z axis	sensor
Top 5	variance	-	variance	EVA (2 directions), AAE
Top 10	RMS, variance, range	variance, range	variance	EVA (3 directions), AAE
Top 15	variance, range, RMS, min, STD	variance, range, RMS	variance, range, RMS	EVA (3 directions), AAE
Top 20	max, RMS, variance, STD, min, range	variance, range, RMS, max, min, STD	RMS, variance, range, min	EVA (3 directions), AAE



(a) Patient 2

(b) Patient 8

Fig. 3. AAE vs. variance on x-axis for (a) Patient 2 data and (b) Patient 8 data

form two distinct clusters when represented by these two features. Still, this does not work for all the patients. For example, in the case of Patient 8, even if the top ranked features with MI are the same, their discriminative power is lower – FoG is easily confused with WALK, when represented by the same two features.

6.2 FoG Detection

We compare the different feature learning approaches in terms of producing discriminative feature sets for distinguishing FoG and normal gait. In Table 3 the sensitivity, specificity and F1 measures for two feature-extraction methods are depicted as a function of number of features N_F used for classification, for a fixed duration of ignored data $T_{Prefog} = 3s$ before each FoG. The classification results are significantly improved when performing unsupervised feature learning compared to the results for the standard feature set, for values of $N_F < 30$. Classification based on PCA features also outperforms the one using FFT-based features, when using small number of PCA features (see Figure 4). In Figure 4, we observe that for larger values of N_F, the classification performances tend to decrease for unsupervised extracted features. PCA concentrates the variability and the useful information from the raw data in the first features. The usefulness of a feature decreases with its rank. However, our target is to use as few features as possible, as noted above.

Table 3. Average of the sensitivity, specificity and F1-measure for the FoG class for supervised and unsupervised feature extraction methods, in the two-class classification problem. The pre-FoG duration is fixed as $T_{Prefog} = 3s$

Sensitivity (%)

Features	5	10	15	20	25
Unsupervised	77.15	77.7	76.29	76.86	76.86
Supervised	67.8	68.53	69.42	66.65	67.58

F1 (FoG)(%)

Features	5	10	15	20	25
Unsupervised	78.2	79.09	77.53	77.62	76.29
Supervised	70.94	72.54	73.79	72.33	73.02

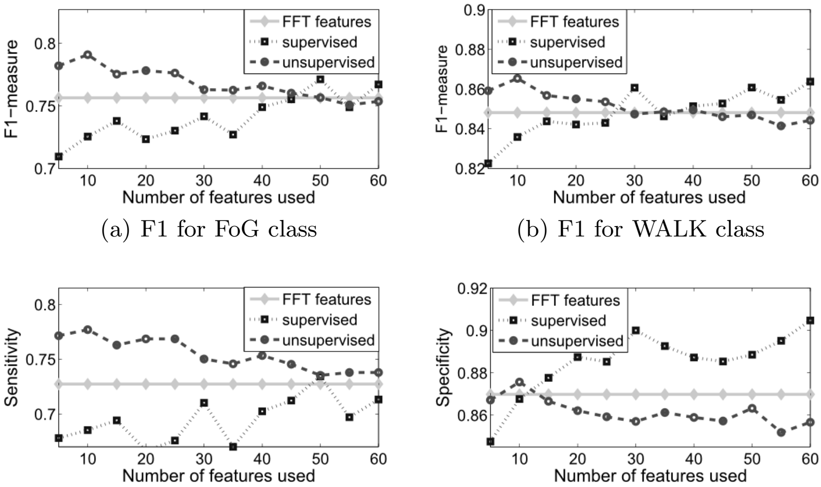
Specificity (%)

Features	5	10	15	20	25
Unsupervised	86.71	87.56	86.65	86.21	85.52
Supervised	84.75	86.76	87.76	88.74	88.52

F1 (WALK) (%)

Features	5	10	15	20	25
Unsupervised	85.91	86.53	85.67	85.5	85.35
Supervised	82.25	83.58	84.37	84.21	84.29

In Figure 5, we present the classification results with $N_F = 10$ features, when varying the amount of discarded data before each FoG episode in the range $T_{Prefog} \in [1s, 11s]$, in steps of 1s. We observe that for both supervised and unsupervised features, FoG detection performance increases with the increase of T_{Prefog}, until reaching a plateau value at $T_{Prefog} = 5 - 6s$. This suggests that these discarded portions of data could contain properties that are different both from FoG and normal locomotion. In the next set of experiments, we analyze whether this dataset has specific properties that will lead to prediction of FoG episodes.



(a) F1 for FoG class

(b) F1 for WALK class

(c) Sensitivity

(d) Specificity

Fig. 4. Sensitivity, Specificity and F1 measures for FoG detection when using different values for N_F unsupervised and supervised extracted features. The amount of discarded data is fixed to $T_{Prefog} = 3s$ before each FoG episode.

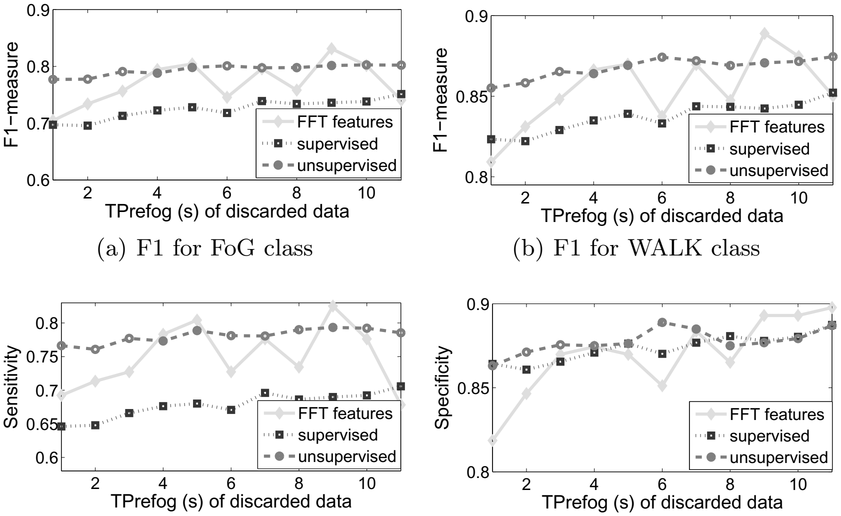
6.3 Towards FoG Prediction

In the previous experiments, we observed that discarding T_{Prefog} data preceding each FoG episode improved the FoG detection results for all types of feature extraction. We now present the results for the three-class classification problem, where we use the discarded chunks as examples of the pre-FoG class. As a first step, we analyzed the impact of the addition of this third class to the mutual information between the various features and the classes.

Mutual Information. We compare the mutual information of the features in the FoG-detection and FoG-prediction problems. Figure 6 shows an example of MI values computed for both supervised and unsupervised features, on the same Patient 2 dataset, in case of FoG detection and FoG prediction problems. We observe that all MI values improve for top ranked features, when adding the third class. This suggests that the pre-FoG data can indeed be representative.

Performance of FoG Prediction. In the next experiments, we set $N_F = 10$, and we varied $T_{pre-FoG}$ from 1s to 6s, in steps of 1s. We stopped at $6s$ because a further increase did not improve the FoG prediction results.

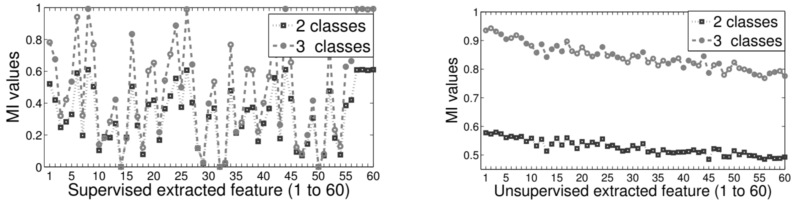
Figure 7 shows the variation of F1-measures for all the three classes versus the value of T_{Prefog}. We first observe that, like in the two-class classification problem (FoG detection), the unsupervised features perform better than the supervised ones. Second, the F1-measures for the FoG and WALK classes are



(a) F1 for FoG class (b) F1 for WALK class

(c) Sensitivity (d) Specificity

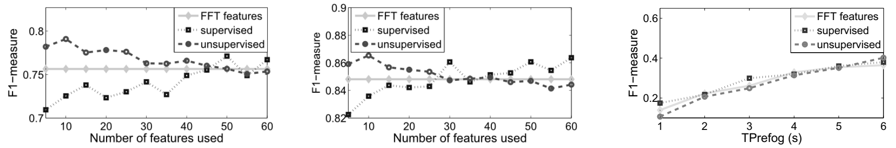
Fig. 5. Sensitivity, Specificity and F1-measures for FoG detection when using $N_F = 10$ unsupervised and supervised extracted features. The amount of discarded data varies between $T_{Prefog} = 1s$ and $T_{Prefog} = 11s$.



(a) Supervised (b) Unsupervised

Fig. 6. MI values for all $N_{total} = 60$ computed features, both with supervised and unsupervised methods, for FoG detection and FoG prediction. $T_{Prefog} = 5s$

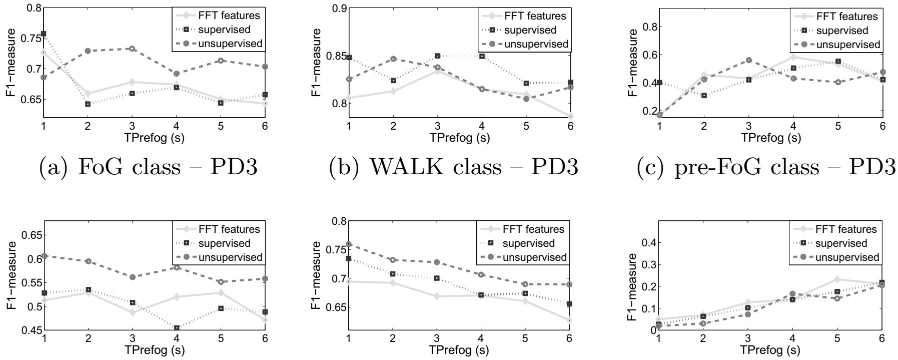
smaller than in the two-class problem. This is expected since we are trying to solve a classification task with one extra class – pre-FoG – which is identified using an assumption on its presence and duration, which leads inevitably to a noisy training. Instances of the pre-FoG class will indeed not always be radically different from WALK or FoG instances, which will introduce confusion. Nevertheless, we identify a trade-off: we can use the unsupervised feature extraction to perform FoG prediction at the expense of performance on detection. Too small values of T_{Prefog} lead to poor results on the F1-measures, because the pre-FoG class is not representative enough. On the contrary, indefinitely increasing T_{Prefog} improves the F1-measure only for this class, but dramatically



(a) F1 for FoG class (b) F1 for WALK class (c) F1 for pre-FoG class

Fig. 7. F1-measures for FoG prediction when using $N_F = 10$ unsupervised and supervised extracted features, with $T_{Prefog} \in [1s, 6s]$

decreases that of the other classes. This is due to the fact that the pre-FoG and WALK classes become more and more similar.



(a) FoG class – PD3 (b) WALK class – PD3 (c) pre-FoG class – PD3

(d) FoG class – PD8 (e) WALK class – PD8 (f) pre-FoG class – PD8

Fig. 8. F1-measures for FoG prediction on Patient 3 and Patient 8 data when using $N_F = 10$ unsupervised and supervised extracted features, with $T_{Prefog} \in [1s, 6s]$.

Figure 8 displays the F1-measure variations for the datasets of Patient 3 (PD3) and Patient 8 (PD8). In the case of PD3, when using unsupervised extracted features, for T_{Prefog} periods of $2s$ and $3s$, the F1-measures increase for all the classes. The F1-measures for the pre-FoG class are 0.42 for $T_{Prefog} = 2s$ and 0.56 for $T_{Prefog} = 3s$. So there are common patterns in the 2s or 3s before FoG episodes that are distinct from WALK and FoG. The same behavior of F1-measures is observed for supervised extracted features, but with a delay compared to using unsupervised features. For $T_{Prefog} = 1s$ supervised features even outperform the unsupervised ones. The likely reason is that with such short pre-FoG durations, PCA in unable to capture the structure of that class. On the other hand, for PD8, an increase of T_{Prefog} leads to a constant decrease in performance for the detection of the FoG and WALK classes, while having a small increase for pre-FoG (along with a decrease of the WALK F1-measure). That shows that WALK and pre-FoG are similar, thus using two distinct classes just leads to confusion in the classification. So, for this patient, we cannot extract specific patterns that could differentiate pre-FoG from the global WALK class.

6.4 Discussion

The classification performance for FoG detection is not as high as that reported in other works. We believe this is mainly due to a less optimistic evaluation scheme, where we selected the training and testing data in each fold to avoid having training and testing data chunks coming from the same FoG episodes at once. This should lead to a more realistic estimate of the performance of a real-world deployed system. Furthermore, FoG-prediction performances vary considerably across subjects. We can claim that for some PD patients, like Patient 3, there are patterns, visible in the accelerometer data, that are characteristic of the pre-FoG class, making it different from the normal locomotion class. These patients exhibit a deterioration of the walk just before FoG episodes.

There are some limitations related to the assumtpions on the pre-FoG class:

- Different nature of the FoG episodes. Some FoG events occur when the subject starts walking, meaning there is no gait before the FoG, so that the gait deterioration that we assume to exist in the pre-FoG phase does not exist, rendering FoG prediction especially challenging for those cases.
- The duration of the pre-FoG class T_{Prefog} is considered to be fixed for each patient. Nevertheless, the pre-FoG pattern duration will probably vary even for different FoG episodes for the same patient. This means that an optimal training set for the pre-FoG class for a single patient might need to contain segments having different values for T_{Prefog}. In order to determine the correct value for each single instance, an approach could involve a direct monitoring of the variation of the features, to detect when they start changing from the normal status to the FoG status.

7 Conclusion

In this work, we analyzed the performance of three feature extraction approaches for detecting freezing of gait in patients with Parkinson's disease. Features based on time-domain and statistical features where compared to unsupervised ones based on principal components analysis, while Freezing Index (FI) was used as a baseline reference. We tested the approaches on acceleration data collected at the ankle from patients that experienced FoG in daily-life. Unsupervised feature learning outperformed FI by up to 7.1% and the time-domain and statistical features by up to 8.1% in terms of F1-measure for FoG detection.

We went a step further by analyzing FoG prediction, i.e. identification of patterns (pre-FoG) occurring before FoG episodes, based only on acceleration data. The purpose is to predict FoG so to assist patients in avoiding freezing periods altogether. For this, we assume that walking sequences of a fixed length T_{Prefog} just previous to a FoG episode have different characteristics compared to normal locomotion patterns and to FoG. On the three-class problem (FoG vs. pre-FoG vs. normal locomotion) we obtained results highly patient-dependent, reaching an F1-measure of 56% in the pre-FoG class for one patient. The identification of pre-FoG patterns is also beneficial for the simple FoG detection: when pre-FoG data are discarded from the training set, performance on FoG detection increases for all the feature extraction methods.

The use of unsupervised features is a promising avenue, since these capture important variations in the data, without the bias of an expert choosing features manually and without any prior knowledge of the class labels. In order to improve the results, other, more complex unsupervised methods for feature learning will be tested (PCA using nonlinear kernels, deep learning). Furthermore, additional unobtrusive sensing modalities could be considered (e.g. gyroscopes). Finally, our assumption on a fixed duration of the pre-FoG class for all FoG events might need to be revised. To this end, methods monitoring directly changes in the extracted features could be beneficial for identifying the actual start of the pre-FoG phases, where present.

Acknowledgments. The research leading to these results has received funding from the European Union - Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 288516 (CuPiD project).

References

1. Bächlin, M., Plotnik, M., Roggen, D., Maidan, I., Hausdorff, J.M., Giladi, N., Tröster, G.: Wearable Assistant for Parkinson's Disease Patients with the Freezing of Gait Symptom. IEEE Transactions on Information Technology in Biomedicine 14, 436–446 (2010)
2. Bloem, B.R., Hausdorff, J.M., Visser, J.E., Giladi, N.: Falls and Freezing of Gait in Parkinson's Disease: A Review of Two Interconnected, Episodic Phenomena. Movement Disorders 19, 871–884 (2004)
3. Cilibrasi, R., Vitányi, P.M.B.: Clustering by Compression. IEEE Transactions on Information Theory 51, 1523–1545 (2005)
4. Cole, B.T., Roy, S.H., Nawab, S.H.: Detecting Freezing of Gait During Unscripted and Unconstrained Activity. In: Engineering in Medicine and Biology Society (EMBC), pp. 5649–5652 (2011)
5. Djurić-Jovičić, M., Jovičić, N.S., Milovanović, I., Radovanović, S., Kresojević, N., Popović, M.B.: Classification of Walking Patterns in Parkinson's Disease Patients Based on Inertial Sensor Data. In: 10th Symposium on Neural Network Applications in Electrical Engineering, pp. 3–6 (2010)
6. Giladi, N., Hausdorff, J.M.: The Role of Mental Function in the Pathogenesis of Freezing of Gait in Parkinson's Disease. Journal of the Neurological Sciences 248, 173–176 (2006)
7. Handojoseno, A.M.A., Shine, J.M., Nguyen, T.N., Tran, Y., Lewis, S.J.G., Nguyen, H.T.: The Detection of Freezing of Gait in Parkinson's Disease Patients Using EEG Signals Based on Wavelet Decomposition. In: Engineering in Medicine and Biology Society (EMBC), pp. 69–72 (2012)
8. Hashimoto, T.: Speculation on the Responsible Sites and Pathophysiology of Freezing of Gait. Parkinsonism & Related Disorders 12, 55–62 (2006)
9. Hausdorff, J.M., Balash, Y., Giladi, N.: Time Series Analysis of Leg Movements During Freezing of Gait in Parkinsons Disease: Akinesia, Rhyme or Reason? Physica A: Stat. Mechanics & Appl. 321, 565–570 (2003)

10. Hausdorff, J.M., Schaafsma, J., Balash, Y., Bartels, A., Gurevich, T., Giladi, N.: Impaired Regulation of Stride Variability in Parkinsons Disease Subjects with Freezing of Gait. Experimental Brain Research 149, 187–194 (2003)
11. Jovanov, E., Wang, E., Verhagen, L., Fredrickson, M., Fratangelo, R.: deFog - a Real Time System for Detection and Unfreezing of Gait of Parkinson's Patients. In: 31th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2009)
12. Latt, M., Lord, S., Morris, J., Fung, V.: Clinical and Physiological Assessments for Elucidating Falls Risk in Parkinson's Disease. Movement Disorders 24, 1280–1289 (2009)
13. Macht, M., Kaussner, Y., Möller, J.C., Stiasny-Kolster, K., Eggert, K.M., Krüger, H.-P., Ellgring, H.: Predictors of Freezing in Parkinson's Disease: A Survey of 6,620 Patients. Movement Disorders 22, 953–956 (2007)
14. Mazilu, S., Hardegger, M., Zhu, Z., Roggen, D., Tröster, G., Hausdorff, J.M.: Online Detection of Freezing of Gait with Smartphones and Machine Learning Techniques. In: 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 123–130 (2012)
15. Moore, O., Peretz, C., Giladi, N.: Freezing of Gait Affects Quality of Life of Peoples with Parkinson's Disease Beyond its Relationships with Mobility and Gait. Movement Disorders 22, 2192–2195 (2007)
16. Niazmand, K., Tonn, K., Zhao, Y., Fietzek, U.M., Schroeteler, F., Ziegler, K., Ceballos-Baumann, A.O., Lueth, T.C.: Freezing of Gait Detection in Parkinson's Disease Using Accelerometer based Smart Clothes. In: Biomedical Circuits and Systems Conference (BioCAS), pp. 201–204 (2011)
17. Nieuwboer, A., Dom, R., De Weerdt, W., Desloovere, K., Janssens, L., Stijn, V.: Electromyographic Profiles of Gait Prior to Onset of Freezing Episodes in Patients with Parkinson's Disease. Brain 127, 1650–1660 (2004)
18. Nutt, J.G., Bloem, B.R., Giladi, N., Hallett, M., Horak, F.B., Nieuwboer, A.: Freezing of Gait: Moving Forward on a Mysterious Clinical Phenomenon. The Lancet Neurology 10, 734–744 (2011)
19. Plotnik, M., Giladi, N., Hausdorff, J.M.: Is Freezing of Gait in Parkinsons Disease a Result of Multiple Gait Impairments? Implications for Treatment. Parkinson's Disease (2012), doi:10.1155/2012/459321
20. Plötz, T., Hammerla, N.Y., Olivier, P.: Feature Learning for Activity Recognition in Ubiquitous Computing. In: 22nd International Joint Conference on Artificial Intelligence (IJCAI) (2011)
21. Tripoliti, E., Tzallas, A., Tsipouras, M., Rigas, G., Bougia, P., Leontiou, M., Konitsiotis, S., Chondrogiorgi, M., Tsouli, S., Fotiadis, D.: Automatic Detection of Freezing of Gait Events in Patients with Parkinson's Disease. Computer Methods and Programs in Biomedicine 110(1), 12–26 (2013)
22. Zhang, M., Sawchuk, A.: A Feature Selection-Based Framework for Human Activity Recognition Using Wearable Multimodal Sensors. In: International Conference on Body Area Networks (BodyNets) (2011)
23. Zhao, Y., Tonn, K., Niazmand, K., Fietzek, U.M., D'Angelo, L.T., Ceballos-Baumann, A., Lueth, T.C.: Online FOG Identification in Parkinson's Disease with a Time-Frequency Combined Algorithm. In: IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 192–195 (2012)

Multi-document Text Summarization Using Topic Model and Fuzzy Logic

Sanghoon Lee, Saeid Belkasim, and Yanqing Zhang

Georgia State University, Computer Science,
Atlanta, Georgia, USA
slee172@student.gsu.edu,
{sbelkasim,yzhang}@cs.gsu.edu

Abstract. The automation of the process of summarizing documents plays a major rule in many applications. Automatic Text Summarization has been focused on retaining the essential information without affecting the document quality. This paper proposes a new multi-document summarization method that combines topic model and fuzzy logic model. The proposed method extracts some relevant topic words from source documents. The extracted words are used as elements of fuzzy sets. Meanwhile, each sentence on the source document is used to generate a fuzzy relevance rule that measures the importance of each sentence. A fuzzy inference system is used to generate the final summarization. Our summarization results are evaluated against some well-known summary systems and performed well in divergences and similarities.

Keywords: Fuzzy Systems, Data mining, Text Analysis, and Semantics.

1 Introduction

Summarization is the creation of a concise document that represents a meaningful content of a given set of documents. As the tremendous increase in information on the internet is overwhelming manual processing, it is necessary to employ computers to automate the process of summarizing documents. Summarization produces condensed information that maintains document coherence and avoids redundancy. Automatic Text Summarization has been used in many areas, such as news and media to become suitable display on a small-sized mobile device. The document processing research community has been trying to generate appropriate metrics for measuring the quality of summarizer since the early 90s several articles dealing with this issue have published. Document Understanding Conference (DUC) [2], Text Retrieval Conferences (TREC) [1], Text Summarization Challenge (TSC) [4], and Text Analysis Conference (TAC)[3] are some of these conferences that have been held from 1992 to present.

The fuzzy set theory [5] which has an advantage of modeling uncertainty can reduce the gap between different meanings in a document. R. Witte et al. [6] presented how the uncertainty in natural language can be modeled with a fuzzy-theory based representation using fuzzy heuristics and fuzzy co-reference

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 159–168, 2013.

chains. Ladda et al. [8] proposed 8 sentence features and applied those features to the fuzzy set theory. Ravindra et. al. [9] describes the automatic evaluation method using Fuzzy set model. Our approach is motivated by [8]. Ladda et al. calculated a score from each sentence in a document as the vector of their features making rules for their fuzzy set, and then extracted important sentences to generate a summary. Their summary was evaluated with other summarizers using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [7]. Our approach is different from their approach: First, we borrow the concept of the Latent Dirichlet Allocation (LDA) model [13] and use this model to generate topic words in all input documents with the help of a graphical user interface tool [14]. Second, we reduced the number of features to only four features: TF-ISF score, title score, length score, and position score. Last, our summary results were evaluated using computer generated evaluation tool instead of human metrics which requires human made summary to be available. This evaluation tool is very important because computer generated summary may not have an equivalent human made summary. The computer generated summaries can still benefit from metrics like ROUGE.

In this paper, we focus on extracting vital sentences from multi-documents using the topic model. For this purpose, a fuzzy technique is used to extract the important sentences present in multi-documents. Because the weights of sentences in various documents are different, the formation of the summaries may have vagueness. Fuzzy set theory and fuzzy logic are ideally suited for dealing with this type of uncertainty. Thus, we propose the fuzzy logic for automatic text summarization and the effectiveness of our proposed method is being evaluated against similar approaches. Section 2 describes our system design that includes a pre-processing stage, the scoring of sentences, and the fuzzy logic approach. Section 3 evaluates our method to other approaches. Finally, conclusions and plans for future extension of our study are given in Section 4.

2 Topic Summary System Design

Generally, two approaches in the automatic text summarization have been studied: extractive summary and abstractive summary. The first one is a summary which generates a summary taken exactly as they appear in the documents whereas the abstractive summary generates paraphrased sentences in the input documents. In this paper, our approach is relevant to the extractive summary. Based on the extractive summary, we use both the topic modeling method and the Fuzzy System to extract a summary from the source documents. Blei et al. proposed the LDA which is a generative probabilistic model of a corpus [13]. LDA generates a set of topics providing a document representation. For generating a topic in the source document, we borrow the concept of LDA topic modeling. Since the weights of sentences in documents are different, the computation of the extractive summary may have vagueness. We used a fuzzy technique to reduce the vagueness. Fig. 1 shows our system design. As a first step, source documents are divided into the sentences for determining each sentence score.

The preprocessing stages: the sentence boundary detection, stop-word remover, and stemming produce a clean sentence. Second, scores are determined for all topics and their associated sentences. Then, these scores are used in the Fuzzy System in order to decide the important sentences in the source document. Last, Fuzzy System selects sentences that can be extracted into the summary document. This procedure is described in Fig. 1.

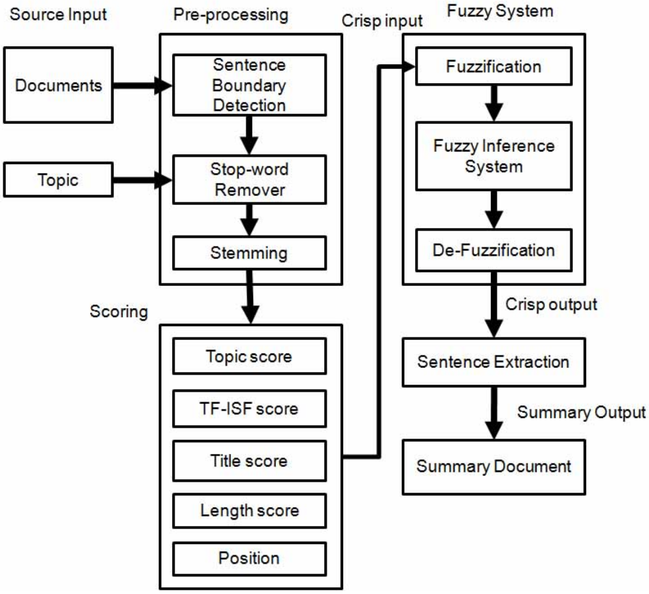


Fig. 1. Design of the Topic Summary System

2.1 Pre-processing

Sentence Boundary Detection. The raw source documents need to be segmented into sentences for processing by other components of the system. The sentence segmentation is achieved with the help of a supervised system that classifies sentence boundaries without any heuristics or hand-generated lists. We used Wall Street Journal news combined with the Brown Corpus to train our system. The trained system produced low error rates on test news data of nearly 0.25%. [10] This model uses the same training data as mxTerminator [11], and allows for Navie Bayes or SVM model. In section 4, we evaluate our produced summary with input from various news documents.

Stop Words Remover. In order to compare the weights between words, we need to remove the stop words which are generally filtered out prior to, or after, processing natural language data. In fact, there is no clear definition for the list of stop words which all tools use. In this paper, we used the most common stop words list that includes words such as a and the. The stop words are removed from the source documents to increase search performance.

Stemming. Stemming is a process for reducing inflected words to their stem, base or root from. Algorithms for stemming have been studied in computer science since 1968. We used Porter Stemmer [12], a very widely used algorithm which became the standard algorithm used for English word stemming.

2.2 Topic Scoring

David Newman [14] developed a graphical user interface tool for the LDA topic modeling.We used the topic modeling toolbox for extracting three topics from source documents. The Newsblaster systems archives [18] are trained as the input documents and each topic is set to 20 topic words.

Topic Score. Each topic word extracted from the source documents is used to in determining a score for each sentence in the document. The sentence score S_T is calculated by the equation below:

$$S_T = numTopics/Max(numTopics) \tag{1}$$

, where $numTopics$ is the number of topic terms in a sentence. $Max(numTopics)$ is the maximum number of topic terms in a sentence. The maximum number of topic terms is computed by counting the number of topic terms in all sentences.

2.3 Feature Scoring

Term Frequency · Inverse Sentence Frequency (TF · ISF). The Term Frequency (TF) is the frequency of term within a document and the Inverse Document Frequency (IDF) is a well-known measure of a words importance. TF · ISF [29] is an adaptation of the TF · IDF [26]. We defined the TF · ISF score S_1 as below:

$$TF \cdot ISF = TF(t, s) \times ISF(t) \tag{2}$$

$$ISF = 1 + log\frac{totalSen}{numSen + 1} \tag{3}$$

$$S_1 = \frac{\sum_{i=1}^{n} TF_i \cdot ISF_i}{Max(\sum_{i=1}^{n} TF_i^k \cdot ISF_i^k)}(k = 1, 2, \ldots, m), \tag{4}$$

where $TF(t, s)$ is the number of term t in a sentence s. $ISF(t)$ is the inverse sentence frequency of the term t. k is the number of sentences. $totalSen$ is the total number of sentences in a document. $numSen$ is the number of sentences that the term t occurs.

Title Score. Jaccard similarity coefficient [27] which measures similarity between two sets is used for the title core. The title score S_2 is based on how many of the title terms are found in a sentence. Thus, the more a sentence contains the title terms, the higher the title score in the sentence is.

$$S_2 = \frac{S_t \cap T_t}{S_t \cup T_t}, \tag{5}$$

where S_t is terms in a sentence. T_t is terms in a title.

Length Score. The length of a sentence is scored. The length score S_3 is computed by normalizing the value given the total number of terms in a sentence. The longer a sentence is, the less the length score in the sentence is. Generally, it returns smaller values when the number of terms is large and returns larger values when the number of terms is small.

$$S_3 = 1 - \frac{L}{Max(L)}, \tag{6}$$

where L is the number of terms in a sentence. $Max(L)$ is the maximum number of terms in a sentence.

Position Score. Our position score is based on the hypothesis in which sentences which occur under certain headings are positively relevant and topic terms tend to occur very early or very late in a document. Position score S_4 is computed as below:

$$S_4 = first5sentences|last5sentences, \tag{7}$$

where $first5sentences$ is the first five sentences scores and $last5sentences$ is the last five sentences scores. Each score in the sentences ranges from 0 to 1.

2.4 Fuzzy System

Fuzzy technique is widely used in various machine control system. In general, an input variable in a fuzzy system are mapped by a set of membership functions called fuzzy sets and the crisp input value is converted into a fuzzy value during the process called fuzzification. Then, the fuzzy system makes decisions generating output values for the input variables based on a set of rules. We used the fuzzy technique to extract the important sentences present in multi-documents. For the fuzzy input membership function, we use five scoring features; topic score, TF-ISF score, title score, length score, and position score. Each feature has fuzzy sets below:

$$S = \{(w, \mu_s(w))|w \in W\}, \tag{8}$$

where S is a fuzzy set. W is a set of degrees for the scores. $\mu_s(w) \in [0, 1]$ is a membership function.

Fig.2 shows the membership function of the TF · ISF score. For the fuzzy inference, we use the logic of the minimum t-norm below:

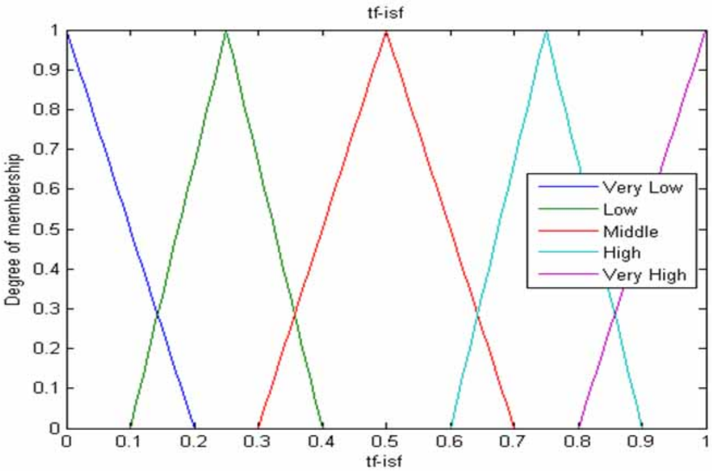


Fig. 2. The mumbership function of TF-ISF

$$\mu_{s1} AND \mu_{s2} = min\{\mu_{s1}, \mu_{s2}\}, \tag{9}$$

where μ_{s1} and μ_{s2} are fuzzy sets.

The rules are performed by using the intersection operation and produce the output fuzzy set. For example, if topic score is high AND TF · ISF score is high AND title score is high AND length score is high AND position score is high, $THEN$ the importance is high. For the de-fuzzification, we use the centroid defuzzification using max-min inferencing below:

$$CenterofGravity = \frac{\sum_{w=1}^{n}(\mu_s(w) \times L_w)}{\sum_{w=1}^{n}\mu_s(w)}, \tag{10}$$

where $\mu_s(w)$ is the mass of the scores and L the location of the mass

3 Evaluation

Many evaluation methods for text document summary are based on human made summary. Generally, researchers compare the human made summary with their computed summary in order to evaluate their summaries. Even though human summaries are still beneficial for evaluating summaries using metrics like ROUGE, we may not be able to evaluate our summaries when we cannot find the reference human summaries. In order to resolve this problem, Annie Louis and Ani Nenkova [17] proposed an evaluation method comparing input documents with candidate summaries without the need for human summary. They analyzed various features for comparing input documents with some summaries and produced very good correlations between their work and humans one. For the sake of evaluating our summary, in this paper, we implemented their evaluation method which uses various features obtained directly from the document.

3.1 Data Sets

A tool SIMetricx-v1 [21] built to compare a generated summary with the input documents is used in the evaluation process. This tool predicts the summary quality producing divergences between vocabulary distributions of the input document and the summary. The input documents consist of input 1, input 2, and input 3 from the Newsblaster systems archives [18]. All input documents and summaries are in plain text and contain no HTML tags etc. In order to compare our summaries, we use two base line summaries and two other summaries:

- base1: A baseline system. The first sentences of each document starting from most recent are added up to the length limit.
- base2: Another baseline where the lead sentences only from the most recent document are added up to the 150 word limit
- News blaster: Summary from the news blaster system
- MEAD: Summary produced by the MEAD summarizer [19]

3.2 Results

Our results are based on the features of [17] which are defined as:

- KLInputSummary - Kullback Leibler divergence between input and summary
- KLSummaryInput - Kullback Leibler divergence between summary and input
- unsmoothedJSD - Jensen Shannon divergence between input and summary
- smoothedJSD -Smoothed Jensen Shannon divergence.
- cosineSimilarity - Cosine similarity between all words in input documents and a summary.
- topicTerms- Proportion of topic terms in a summary that are topic words of the input.
- unigramProb - Unigram probability of a summary
- multinomialProb - Multinomial probability of a summary

Table 1 shows the results of the features for summaries. For establishing the effectiveness of our system, we used three document sets: input 1, input 2, and input 3, which include 20, 15, and 19 documents respectively. Kullback Leibler (KL) divergence [30] is a non-symmetric method of measuring the difference between two probability distributions. The divergence is defined as the average number of words wasted by the samples that have two probability distributions: the input document and the summary. In this paper, we computed the divergence by two ways, input-summary and summary-input because KL divergence is not symmetric. Jensen Shannon (JS) divergence is a well-known method of measuring

Table 1. Comparision our system with the baselines

ID	SysID	KL-IS	KL-SI	JSD-unS	JSD-S	Cosine	Topics	uni-Prob	multi-Prob
input1	Base1	1.92115	1.61838	0.45389	0.32542	0.60395	0.47297	-201.19195	-97.63694
	Base2	2.03388	1.65006	0.4537	0.33553	0.88069	0.50588	-228.03998	-108.28518
	MEAD	1.88596	1.38862	0.4276	0.30887	0.6537	0.62651	-209.58353	-94.62087
	NewsB	2.18943	1.8889	0.48795	0.36337	0.78131	0.4125	-230.99905	-115.90475
	TFuzzy	**1.85217**	**1.44667**	**0.45941**	**0.30961**	**0.8256**	**0.66102**	**-147.10274**	**-73.18053**
Input2	Base1	1.90647	1.44921	0.40439	0.30928	0.6541	0.41975	-207.71389	-92.9944
	Base2	2.65826	2.00215	0.5026	0.40801	0.28824	0.21591	-252.33676	-123.7654
	MEAD	1.84536	1.30667	0.38211	0.29166	0.79683	0.54651	-212.6833	-91.04515
	NewsB	1.96263	1.4549	0.40172	0.31111	0.7185	0.49425	-222.87001	-98.74827
	TFuzzy	**1.64461**	**1.51851**	**0.50393**	**0.2885**	**0.62851**	**0.69565**	**-53.60434**	**-33.47514**
Input3	Base1	1.74798	1.49149	0.43173	0.30163	0.63495	0.45946	-195.50509	-92.72824
	Base2	1.97371	1.77928	0.4775	0.33862	0.80364	0.37143	-191.40065	-97.54206
	MEAD	1.87949	1.46613	0.42816	0.31038	0.75478	0.44828	-225.95963	- 102.96823
	NewsB	1.88447	1.45429	0.42111	0.3091	0.76432	0.4086	-244.32748	-109.48751
	TFuzzy	**1.62052**	**1.55735**	**0.52489**	**0.28983**	**0.71508**	**0.5**	**-69.93805**	**-43.33927**

the similarity between two probability distributions [28][26]. JS divergence is a natural extension of the KD divergence to a set of distributions. We borrowed the concept of smoothing method [17] to avoid the undefined divergence problem in which the second probability of the summary-input document is zero. In table 1, higher divergence scores indicate poor quality summaries and higher scores of other features indicate better summaries. Our system generates the lowest divergence scores in the KL input-summary divergence of input 1, input 2, and input 3 and smoothed JS divergence of input 2 and input 3. This is because the average number of words wasted in both the KL input-summary divergence and smoothed JS divergence is the smallest meaning that the similarity between input documents and the summary is the highest. On the other hand, our system obtained high divergence scores both in the KL summary-input divergence and unsmoothed JS divergence. In order to compare the similarity between input documents and summary, we computed the cosine similarity. The score of our summary in the cosine similarity is lower than others comparatively. However, high scored summaries in the cosine similarity do not mean that the summaries are more important than others. This is because the cosine similarity is only computed with all words in the input documents. We computed the proportion of the topic words in summaries. In all document sets, our summary has a higher proportion of topic words to other summaries. We also computed the unigram probability and multinomial probability of summaries. These probabilities represent the log likelihood of a summary given the input documents. The results show our summary has more weights than other summaries in those features.

4 Conclusion

We proposed a multi-document summary system using both a topic model and a fuzzy method, which reduces the error rates of divergence between the input document and the summary. For extracting the topics on the source documents, the Latent Dirichlet Allocation (LDA) method is used, and each sentence in the input documents is scored by using the topic words. Also, other scores were generated by adding different weights for sentences. With the fuzzy system, these scores produce a rule based decision that determines whether a sentence is important or not. We also compared our system with other systems by evaluating their divergences and similarity. Our summary have showed a good quality in KL input-summary divergence and smoothed JS divergence between input documents and summary. However, our summary does not have good qualities in other divergences and cosign similarities. To solve this problem, we propose to the type2-fuzzy set system that can handle larger uncertainty [24] and continue to reduce the similarity gap between the input document and the summary. Recently, many summarization models have been proposed for medical domain [22][23], but the medical articles still have vague terms, making it difficult to extract the information. We will need to use medical data as input documents generating a summary, and this work will be very beneficial for extracting important meaning of the documents.

References

1. Voorhees, E.M., Harman, D.K.: The Eight Text Retrieval Conference (TREC-8). In: National Institute of Standards and Technology (NIST) (1999)
2. DUC.: The Document Understanding Conference (2001-2007), http://duc.nist.gov
3. TAC.: Text Analysis Conference (2008-present), http://www.nist.gov/tac/
4. Fukushima, T., Okumura, M.: Text summarization challenge: text summarization in Japan. In: NAACL 2001 Workshop Automatic Summarization, pp. 51–59 (2001)
5. Zadeh, L.A.: Fuzzy sets. In: Yager, R.R., Ovchinnikov, S., Tong, R.M., Nguyen, H.T. (eds.) Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh, pp. 29–44. Wiley and Sons (1987); Originally published in Information and Control, vol. 8, pp. 338–353. Academic Press, New York (1965)
6. Witte, R., Bergler, S.: Fuzzy coreference resolution for Summarization. In: 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS), pp. 43–50. Universit Ca Foscari, Venice (2003)
7. Lin, C.-Y.: ROUGE: a Package for Automatic Evaluation of Summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25-26 (2004)
8. Suanmali, L., Salim, N., Binwahlan, M.S.: Fuzzy Logic Based Method for Improving Text Summarization. International Journal of Computer Science and Information Security (IJCSIS) 2(1) (2009)
9. Ravindra, G., Balakrishnan, N., Ramakrishnan, K.R.: Automatic Evaluation of Extract Summaries Using Fuzzy F-score Measure. In: 5th International Conference on Knowledge Based Computer Systems, December 19-22, pp. 487–497 (2004)
10. Gillick, D.: Sentence Boundary Detection and the Problem with the U.S. The Association for Computational Linguistics, 241–244 (2009)
11. Reynar, J.C., Ratnaparkhi, A.: A Maximum Entropy Approach to Identifying Sentence Boundaries. In: 5th Conference on Applied Natural Language Processing, Washington, D.C., March 31-April 3 (1997)

12. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)
13. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
14. Newman, D.: Topic modeling tool, `http://code.google.com/p/topic-modeling-tool`
15. Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
16. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. In: Fuzzy Sets and Systems, pp. 9–34. Elsevier, Amsterdam (1999)
17. Louis, A., Nenkova, A.: Summary Evaluation without Human Models. In: Text Analysis Conference (TAC) (2008)
18. McKeown, K., Barzilay, R., Chen, J., Elson, D.K., Evans, D.K., Klavans, J., Nenkova, A., Schiffman, B., Sigelman, S.: Columbia's Newsblaster: New Features and Future Directions. In: HLT-NAACL (2003)
19. Timothy, D.R., Allison, T., Blair-goldensohn, S., Blitzer, J., Elebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Winkel, A., Zhang, Z.: MEAD a platform for multidocument multilingual text summarization. In: International Conference on Language Resources and Evaluation (2004)
20. Conroy, J.M., Schlesinger, J.D., O'Leary, D.P.: Topic-Focused Multi-Document Summarization Using an Approximate Oracle Score. In: The ACL 2006 / COLING 2006 (2006)
21. SIMetrix: Summary Input similarity Metrics, `http://www.cis.upenn.edu/~lannie/IEval2.html`
22. Summerscales, R.L., Argamon, S., Bai, S., Huperff, J., Schwartzff, A.: Automatic Summarization of Results from Clinical Trials. In: BIBM, pp. 372–377 (2011)
23. Kiritchenko, S., Bruijn, B., Carini, S., Martin, J., Sim, I.: Exact: automatic extraction of clinical trial characteristics from journal publications. BMC Med. Inform. Decis. Mak. 10(1), 56 (2010)
24. Zadeh, L.A.: The Concept of a Linguistic Variable and Its Application to Approximate Reasoning1. Information Sciences 8, 199–249 (1975)
25. Neto, L., Santos, A.D., Kaestner, C.A.A., Freitas, A.A.: Document Clustering and Text Summarization. In: 4th Int. Conf. Practical Applications of Knowledge Discovery and Data Mining (PADD 2000), pp. 41–55. The Practical Application Company, London (2000)
26. Salton, G., Buckley, C.: Term-weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24, 513–523 (1988); Reprinted in: Sparck Jones, K., Willet, P.: Readings in Information Retrieval, pp. 323–328. Morgan Kaufmann (1997)
27. Jaccard, P.: Etude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Soci. Vaudoise des Sciences Naturelles 37, 547–579 (1901)
28. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991)
29. Dhillon, I., Mallela, S., Kumar, R.: Enhanced word clustering for hierarchical classfication. In: Proc. of 8th ACM Intl. Conf. on Knowledge Discovery and Data Mining (2002)
30. Kullback, S., Leibler, R.A.: On Information and Sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)

A Pattern Based Two-Stage Text Classifier

Moch Arif Bijaksana[1,2], Yuefeng Li[1], and Abdulmohsen Algarni[3]

[1] School of Electrical Engineering and Computer Science,
Queensland University of Technology,
Brisbane, QLD 4000, Australia
[2] Informatics Faculty, Telkom Institute of Technology,
Bandung 40257, Indonesia
[3] College of Computer Science, King Khalid University,
P.O. Box 394, Abha 61411, Saudi Arabia

Abstract. In a classification problem typically we face two challenging issues, the diverse characteristic of negative documents and sometimes a lot of negative documents that are closed to positive documents. Therefore, it is hard for a single classifier to clearly classify incoming documents into classes. This paper proposes a novel gradual problem solving to create a two-stage classifier. The first stage identifies reliable negatives (negative documents with weak positive characteristics). It concentrates on minimizing the number of false negative documents (recall-oriented). We use Rocchio, an existing recall based classifier, for this stage. The second stage is a precision-oriented "fine tuning", concentrates on minimizing the number of false positive documents by applying pattern (a statistical phrase) mining techniques. In this stage a pattern-based scoring is followed by threshold setting (thresholding). Experiment shows that our statistical phrase based two-stage classifier is promising.

Keywords: Two-stage classification, Text classification, Pattern mining, Scoring, Thresholding.

1 Introduction

In real life, many classification problems are multi-class and multi-label. Multi-class and multi-label classification is popularly solved by splitting into several binary classifications. Support Vector Machine (SVM) and Rocchio classifiers usually apply this approach. Binary classification theoretically is more generic than multi-class classification or multi-label classification [15]. In this paper we use binary dataset for experiments.

In a classification problem typically we face two challenging issues, the diverse characteristic of negative documents and sometimes a lot of negative documents that are closed to positive documents. Therefore, it is hard for a single classifier to clearly classify incoming documents into classes. Most of existing popular text classifier such as SVM, Rocchio and k Nearest Neighbours (kNN) are single stage classifiers.

Term [1] is the most common type of feature in document representation. A complex natural language document is transformed into a set of simple independent terms. Using

[1] Terms are normalized words. Word normalization handles superficial differences, such as accents and diacritics, capitalization/case-folding, and other issues in a language e.g. color vs. colour in English [10].

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 169–182, 2013.

simple term feature makes the classification efficient. However, relation information among terms is lost [16].

A topic might have clues (good indicators) to represent the topic. For example in TREC-11 RCV1 corpus [17]; topic “Economic espionage” (e.g. “spy”, “espionage”, “industry”) has less number of good indicators than topic “Progress in treatment of schizophrenia” (a lot of treatments jargon). In topics with a lot number of clues, term-based might not be able to catch the theme of document, so the effectiveness is low [14]. A solution to this problem, we can use terms co-occurrence approach. A new approach for document representation is using termset (pattern), a “statistical phrase”. Pattern Taxonomy Model (PTM) [18,6] used intra-document based frequent closed sequential pattern with paragraph as the transactional unit.

The main evaluation metric for text classification is F measure which is based on recall and precision [15]. The problem of text classification can be divided into recall-oriented classifier and precision-oriented classifier separately. Our approach combine Rocchio as a recall-oriented stage, and a novel pattern based scoring-thresholding stage as precision-oriented stage.

Typically, scoring process is conducted by the classifier and thresholding is a post-processing. For classification, thresholding is often considered as a trivial process and is not important; therefore it is under-investigated. However, Yiming Yang [20] proved that thresholding is important and not simple. She proved that an effective thresholding strategy produces significantly better classification than other thresholding strategies.

We propose a new precision-oriented classifier which set both score and threshold values explicitly. In scoring phase, we focus on describing positive feature by using patterns, statistical semantic features that captures semantic relations between terms. While in thresholding phase we use an effective training-based model.

We conducted substantial experiments on popular text classification corpus based on Reuters Corpus Volume 1 (RCV1), compared with SVM, Rocchio and another two-stage classifier to evaluate the proposed model. The results show that our pattern based two-stage classifier is promising.

The rest of this paper is structured as follows: Section 2 discusses related work. Section 3 proposes our two-stage text classification approach. The experiment design is described in Section 4, whereas the results are discussed in Section 5. Finally, Section 6 gives concluding remarks.

2 Related Work

SVM and Rocchio are among the most popular learning algorithms for text classification [15,10]. SVM is an outstanding text categorization method because of its capability to overcome text properties [4]. The properties of text are high dimensional (more than 1000), few irrelevant features (dense concept vector), sparse document vectors (most feature in document vector are zero), and most text categorization problems are linearly separable [4]. Even if all available features were used (no dimensional reduction), SVM had good effectiveness, and difficult to be beaten [11].

Classification can be done in two ways, new documents is directly predicted the class, or performed in two-stages scoring/ranking and thresholding [15]. Scoring process conducted by the classifier and thresholding is a post-processing. Information retrieval models are the basis of ranking algorithm that is used in search engines to produce the ranked list of documents [2].

Scoring for ranking is the main problem in information filtering field, the objective is to effectively score incoming documents to rank. Some recent work in information filtering are including [6].

Different to the ranking task, classification is the task of assigning documents to predefined categories. A comprehensive review of text classification methods can be found in [15]. To date, many classification methods, such as Naive Bayes, Rocchio, kNN and SVM have been developed in IR [10].

For classification, thresholding is often considered a trivial process and is not important; therefore it is under-investigated. However, Yiming Yang [20] proved that thresholding is important and not simple. She proved, using kNN, an effective thresholding strategy produces significantly better classfication effectiveness significantly than other thresholding strategies.

Existing works on thresholding strategy are generally in the context of post-procesing classification or for multi-label classification problems such as [3,20]. However, in principle, these thresholding strategies can be used for tranforming ranking to binary decision classification. To our knowledge, only a few number of works focused on thresholding strategies for ranking into binary decision, among others [7,21].

There are two steps in rank to binary decision transformation, firstly scoring documents $score(d_j, c_i)$, then thresholding [15]. These classifier usually use default threshold, for example threshold score is zero for SVM and probability is 0.5 for Bayes classifier [3]

There are at least three popular thresholding strategies, namely ranking-based, score-based, and proportional-based. Yiming Yang [19,20] respectively named them as $RCut$ (rank-based thresholding), $SCut$ (score-based local optimization), and $PCut$ (proportion-based assignment). Ranking-based thresholding is referred as *fixed thresholding* [15] or *"k-per-doc" thresholding* [5]; and score-based thresholding as *CSV thresholding* [13,15].

A two-stage method for information filtering was introduced in [8,9] to significantly improve the performance.

3 A Pattern Based Two-Stage Text Classifier

The set of negative documents have a variety of topics. Simply put, negative documents N divided into two parts, the N_1 and N_2. N_1 (near negative documents) are documents that have close similarity with positive documents P (see Figure 1).

Figure 2 shows the classification of two-stage global framework. By using the same training set, each stage produces a classification model. In classifying phase, the classification model on stage one concentrates to identify negative documents. At this stage, documents that are predicted as negative documents are grouped into TN_1 (true negative group one) if the document is a true negative, or FN_1 (false negative group one) if the documents actually are positive documents. At this stage (stage one), the priority is to minimize FN rate, with acceptable FP (false positive, i.e. negative documents

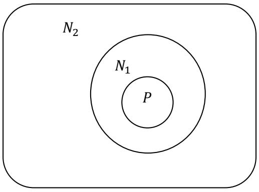


Fig. 1. Positive P, and negative N_1 (near positive), N_2 in a binary class

falsely predicted as postive) rate. Then, classification model two, whic is produced in stage two, is used to identify documents that positively predicted in stage one. In our two-stage model, true negative $TN = TN_1 + TN_2$, false negative $FN = FN_1 + FN_2$, true positive TP, and false positive FP. This stage two is a fine tuning process.

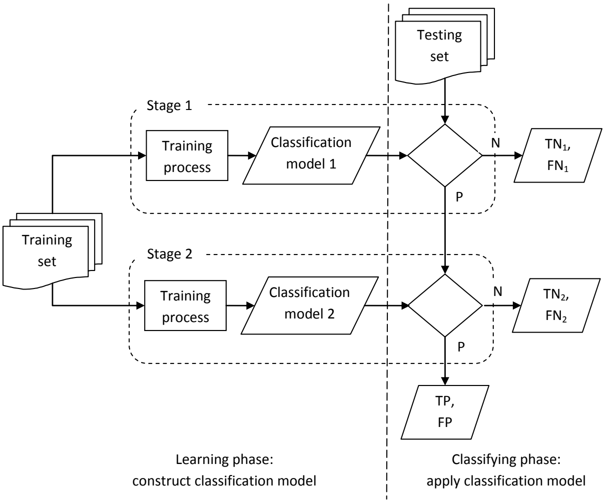


Fig. 2. Two-stage framework

In our two-stage model, at classifying phase the first stage is a recall-oriented focused on documents with low scores, while the second stage is a precision-oriented focused on documents with high scores (see Figure 3).

Our two-stage classifier (*TSC*) use Rocchio classifier for stage one and a pattern-based RFD$_\tau$ classifier for stage two. Our proposed classifier RFD$_\tau$ is decribed in next subsection. Learning and classifying phases of TSC algorithms are outlined in Algorithm 1 and Algorithm 2.

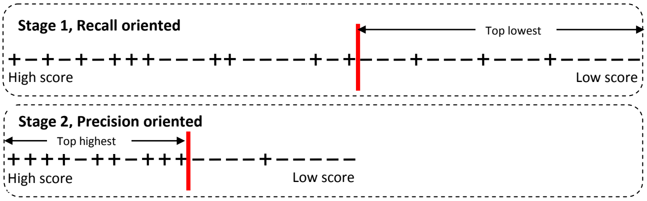


Fig. 3. Two-stage framework: classifying phase

Algorithm 1. TSCLearning

```
Input  : A training set, $D = D^+ \cup D^-$.
Output: Rocchio classification model; and
        threshold value, $\tau$.
// Learn training dataset using Rocchio classifier, get
   Rocchio model.
$Model_{Rocchio} = Classifier_{Rocchio}(D)$ ;
// Calculate the score of training documents using $RFD$.
$D_{score} = RFD(D, min\_sup, \theta_1, \theta_2)$ ;
// Calculate the threshold value, $\tau$.
$\tau = Thresholding(D_{score})$ ;
```

Algorithm 2. TSCClassifying

```
Input  : A new unlabel document;
        Rocchio classification model; and
        threshold value, $\tau$.
Output: Class label for unlabel document.
// Pedict label the new documents $d_{unlabeled}$ by using Rocchio
   model.
$d_{labeled} = Rocchio(d_{unlabeled}, Model_{Rocchio})$ ;
// If Rocchio label it as negative, so the final label of
   the documen is negative
if $d_{labeled}$ is negative then label of $d$ is negative ;
// If Rocchio label it as positive, so the final label of
   the documen is depend on $RFD_\tau$
else $d_{labeled} = RFD_\tau(d_{unlabeled}, Model_{Rocchio})$ ;
```

3.1 Pattern-Based Scoring Model

In this paper, we assume that all documents are split into paragraphs. So a given document d_i yields a set of paragraphs $PS(d_i)$. In our model, we use sequential closed patterns. The definition of sequential closed pattern can be found in [6].

Table 1. Pattern based document representation

Doc	Patterns
d_1	$\langle carbon\rangle_4, \langle carbon, emiss\rangle_2, \langle air, pollut\rangle_2$
d_2	$\langle greenhous, global\rangle_3, \langle emiss, global\rangle_2$
d_3	$\langle greenhous\rangle_2, \langle global, emiss\rangle_2$
d_4	$\langle carbon\rangle_3, \langle air\rangle_3, \langle air, antarct\rangle_2$
d_5	$\langle emiss, global, pollut\rangle_2$

Table 1 illustrates document representation in our pattern-based model, which is based on Relevance Feature Discovery (RFD) [6] and Pattern Taxonomy Model (PTM) [18]. In this figure d_1 has three pattern features $\langle carbon\rangle_4$, $\langle carbon, emiss\rangle_3$, and $\langle air, pollut\rangle_2$. Subscripted values are support values which represents weight. It means that in d_1 there are four paragraphs contain pattern $\langle carbon\rangle$, three paragraphs contain pattern $\langle carbon, emiss\rangle$, and two paragraphs contain pattern $\langle air, pollut\rangle$. A termset X is called a frequent sequential pattern if its relative support $supp_r(X)$ is greater than or equal to a predefined minimum support, that is, $supp_r(X) \geq min\_sup$.

In pattern-based, the class representation is in the form of a set of weighted terms. The number of terms in class representation is relatively small compared to the size of the vocabulary in the class. The terms weight in class representation is calculated from the appearance of terms in the document representation (patterns). There are several ways to calculate the weight of term [6,18]. The basic weight of term t in dataset D^+ is

$$weight(t, D^+) = \sum_{i=1}^{|D^+|} \frac{|\{p|p \in SP_i, t \in p\}|}{\sum_{p \in SP_i} |p|}$$

where SP_i is pattern set of pattern p in document d_i, and $|p|$ is the number of term in pattern p. For example in Table 1, $D^+ = \{d_1, d_2, \ldots, d_5\}$, term *global* (which appears in document d_2, $d_3, \ldots d_5$), has $weight(global, D^+) = \frac{2}{4} + \frac{1}{3} + \frac{1}{3} = \frac{7}{6}$.

Algorithm 3 describes RFD document scoring model.

3.2 Thresholding Model

The threshold value (τ) in our model is based on score of document. Score in document d_i, $Score(d_i)$, is its weight in RFD model. For thresholding, score can be based on

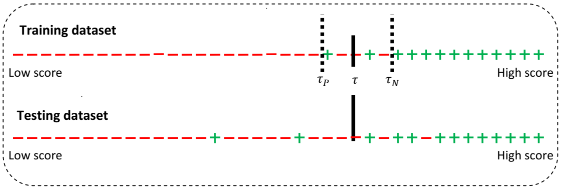


Fig. 4. Threshold setting

Algorithm 3. RFD

```
Input  : A training set, $D = D^+ \cup D^-$;
         parameter minimum support, $min\_sup$; and
         experimental parameters, $\theta_1$ and $\theta_2$.
Output: Score of a document, $score(d)$.
// Extract positive patterns $P^+$ from $D^+$. Detail of
   algorithm SPMining is in [18].
$P^+$ =SPMining$(D^+, min\_sup)$ ;
// Extract terms $T^{P^+}$ from $P^+$.
$T^{P^+} = \{t \mid t \in p, p \in P^+\}$ ;
// Calculate initial term weight of $T^{P^+}$.
$weight_i(t) = support(t, D^+), t \in T^{P^+}$ ;
// Select top rank negative samples (offender), $D_o^-$.
$D_o^- = \{d_i \mid d_i \in D^-, score(d_i) > 0, i < \frac{\lceil D^+ \rceil}{2}\}$ ;
// Extract negative patterns $P_o^-$ from $D_o^-$.
$P_o^-$ =SPMining$(D_o^-, min\_sup)$ ;
// Extract terms $T^{P_o^-}$ from $P_o^-$.
$T^{P_o^-} = \{t \mid t \in p, p \in P_o^-\}$ ;
// Calculate initial term weight for terms that just only
   appear in $D_o^-$.
$weight_i(t) = -support(t, D^+), t \in \{T^{P_o^-} - T^{P^+}\}$ ;
// Get the set of terms $T$ as union of $T^{P^+}$ and $T^{P_o^-}$.
$T = T \cup T_o$ ;
// Classify $T$ into three categories based on their
   specificity: positive specific $T^{spe+}$, negative specific
   $T^{spe-}$, and general $T^{gen}$ .
$\begin{pmatrix} T^{spe+} & = \{t \in T \mid spe(t) > \theta_2\}, \\ T^{gen} & = \{t \in T \mid \theta_1 \leq spe(t) \leq \theta_2\}, and \\ T^{spe-} & = \{t \in T \mid spe(t) < \theta_1\} \end{pmatrix}$,
where $spe(t) = \frac{1}{n} \times (|\{d \mid d \in D^+, t \in d\}| - |\{d \mid d \in D_o^-, t \in d\}|)$;
// Revise term weight function, based on their term
   specificity.
$weight(t) = \begin{cases} weigth_i(t) + (weight_i(t) \times spe(t)), & \text{if } t \in T^{spe+} \\ weigth_i(t), & \text{if } t \in T^{gen} \\ weigth_i(t) - (weigth_i(t) \times spe(t)), & \text{if } t \in T^{spe-} \end{cases}$
// Calculate document score
$score(d) = \sum_{t \in T, t \in d} weight(t)$
```

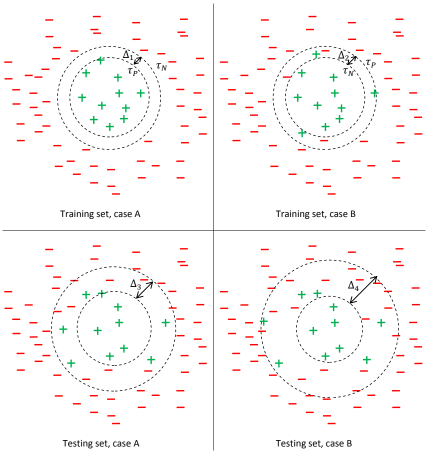


Fig. 5. Training and testing cases. Case A is a non-overlap training score $\tau_P > \tau_N$, case B is an overlap training $\tau_P < \tau_N$. In both case A and case B testing score are overlap, and usually $\varDelta_3 < \varDelta_4$.

training set, validation set, or testing set. For a dataset with a small number of training set, it is difficult to get a representative validation set. Using testing set for thresholding makes thresholding model is not suitable for online learning. Our model generate threshold based on score of training set D, which consists of a set of positive documents, D^+, and a set of negative documents, D^-.

Algorithm 4. RFD$_\tau$Learning

```
Input  : A new unlabel documents;
         Rocchio classification model; and
         threshold value, $\tau$.
Output: Class label for unlabel document.

// Calculate score of training document
$D_{score} = RFD(D, min\_sup, \theta_1, \theta_2)$ ;
// Calculate the threshold value, $\tau$.
$\tau = Thresholding(D_{score})$ ;
```

Our τ is based on minimum score of positive training document (τ_P) and minimum score of negative training document (τ_N) (see Figure 4).

$$\tau_P = \min_{d_i \in D^+} \{Score(d_i)\}$$

$$\tau_N = \max_{d_i \in D^-} \{Score(d_i)\}$$

A multi-class classification can run in several binary classification. These binary classifications usually have imbalance classification problem, where the number of negative

Algorithm 5. RFD$_\tau$Classifying

```
Input  : A new unlabel documents;
         Label Rocchio classification model; and
         threshold value, $\tau$.
Output: Class label for unlabel document.
// Classify document based on $\tau$ and its $score$
if $score(d) > \tau$ then d is labeled as positive documents ;
else d is labeled as negative documents
```

much more than of positive. The performance of imbalance classification problem typically has peak on the left side of data distribution (see Figure 6).

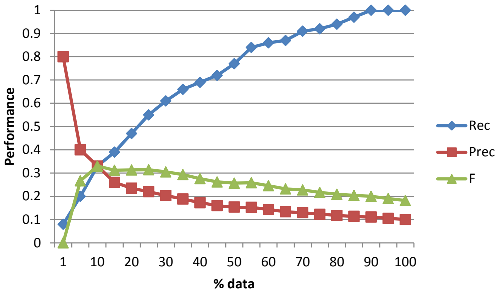


Fig. 6. Typical performance for binary classification

Based on [7] with only score of positive training documents D^+ available, the optimal threshold is τ_P. In a real dataset, in most cases the maximum score of negative testing document are more than the minimum score of positive testing document (see Figure 5). Therefore,

$$\tau = \tau_P - \alpha$$

where $0 \leq \alpha \leq (\tau_P - \tau_N)$. With simplification version

$$\tau_P \leq \tau \leq \tau_N$$

With both D^+ and D^- available, we found that

$$\tau = \min(\tau_P, \tau_N)$$

that is

$$\tau = \begin{cases} \tau_P, & \text{if } \tau_P < \tau_N \\ \tau_N, & \text{if } \tau_P > \tau_N \end{cases}$$

Algorithm 4 and Algorithm 5 describe RFD$_\tau$ in learning and classifying phase respectively.

4 Evaluation

In this section, we first discuss the data collection used for our experiments. We also describe the baseline models and their implementation.

In this study we use the 50 assessor topics of TREC-11 Filtering Track RCV1[2] dataset contains 21,605 documents. According to Buckley and others [1], 50 topics are stable and enough for high quality experiments. RCV1 corpus consists of all and only English language stories produced by Reuter's journalists. Each topic in the dataset is binary class with its own positive and negative set. Documents in TREC-11 are from RCV1, has developed and provided 100 topics for the filtering track aiming at building a robust filtering system. The first 50 topics of TREC-11 RCV1 documents were categorised by humans; the other 50 topics were categorised by intersecting two Reuters topic categories. The assessor topics are more reliable and the quality of the intersection topics is not quite good [17].

The documents are treated as plain text documents by preprocessing the documents. The tasks of removing stop-words according to a given stop-words list and stemming terms by applying the Porter Stemming algorithm are conducted.

Two popular baseline classifiers are used: Rocchio, and SVM. In this paper, our proposed model is called TSC (Two-stage Classifier).

The Rocchio algorithm [12] has been widely adopted in the area of text categorization. It can be used to build the profile for representing the concept of a topic which consists of a set of relevant (positive) and irrelevant (negative) documents. Two centroids $\boldsymbol{c}_+$ for positive and $\boldsymbol{c}_-$ for negative are generated

$$\boldsymbol{c}_+ = \frac{1}{|D^+|} \sum_{\boldsymbol{d} \in D^+} \frac{\boldsymbol{d}}{||\boldsymbol{d}||}$$

$$\boldsymbol{c}_- = \frac{1}{|D^-|} \sum_{\boldsymbol{d} \in D^-} \frac{\boldsymbol{d}}{||\boldsymbol{d}||}$$

For predicting new documents, we use cosine similarity between centroid $\boldsymbol{c}_j$ and document $\boldsymbol{d}_i$

$$sim(\boldsymbol{c}_j, \boldsymbol{d}_i) = \frac{\boldsymbol{c}_i \cdot \boldsymbol{d}_i}{||\boldsymbol{c}_i|| \times ||\boldsymbol{d}_i||}$$

New documents will be predicted as positive if they are more similar to positive centroid, otherwise it will be predicted as negative.

SVM is a statistical method that can be used to find a hyperplane that best separates two classes. SVM is one of state of the art of text classifier. It achieved the best performance on the Reuters-21578 data collection for document classification [15]. For experiments in this paper, we used SVM^{Light} package [3].

Effectiveness for text classification was measured by two different means F_β and Accuracy (Acc). F_β is the most important metric [15]. F_β is a unification value of Recall (R) and Precision (P):

[2] http://trec.nist.gov/data/t2002_filtering.html

[3] http://svmlight.joachims.org/

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

The parameter $\beta = 1$ is used in this paper, which means that recall and precision is weighed equally.

$$F_1 = \frac{2PR}{P + R}$$

To get the final result of several topics, two different ways may be adopted, microaveraging (F_1^μ) and macroaveraging (F_1^M) [15]:

$$F_1^\mu = \frac{2P^\mu R^\mu}{(P^\mu + R^\mu)}$$

where

$$P^\mu = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FP_i)}$$

$$R^\mu = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FN_i)}$$

TP (True Positive) is the number of documents the system correctly identifies as positives; TN (True Negative) is the number of documents the system correctly identifies as negatives; FP (False Positive) is the number of documents the system falsely identifies as positives; FN (False Negative) is the number of positive documents the system fails to identify; and $|\mathcal{C}|$ is the number of topics.

$$F_1^M = \frac{\sum_{i=1}^{|\mathcal{C}|} F_{1,i}}{|\mathcal{C}|}$$

where $F_{1,i}$ is the F_1 for topic i.

For accuracy,

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Acc^\mu = \frac{\sum_{i=1}^{|\mathcal{C}|}(TP_i + TN_i)}{\sum_{i=1}^{|\mathcal{C}|}(TP + FP + TN + FN)}$$

$$Acc^M = \frac{\sum_{i=1}^{|\mathcal{C}|} Acc_i}{|\mathcal{C}|}$$

The statistical method, paired two-tailed *t-test*, is also used to analyze the experimental results. If the associated *p*-value is low ($<$ 0.05), that shows the difference in means across the paired observations is significant.

Table 2. Comparison of TSC and popular text classifers

Model	Macro-average		Micro-average	
	F_1	Acc	F_1	Acc
TSC	**0.44**	**0.69**	**0.57**	**0.71**
SVM	0.19	0.56	0.50	0.61
Rocchio	0.37	0.66	0.42	0.68
$\%change$	19.6%	4.4 %	12.8%	3.7%

5 Results and Discussion

In this section we present the experimental results for comparing the proposed model TSC with the baseline models. The results listed in Table 2 and Table 3 show that our model outperforms baseline models and other two- stage model.

In accuracy results, with using original testing for dataset with imbalanced number of positive and negative, accuracy produce misleading measurements. For example in our TREC-11 RCV1 dataset if we predict all of testing documents as negative, we get Recall = 0, Precision=0, F_1 = 0, Accuracy = 81% (macro average), 82% (micro average). If all testing documents are predicted as positive, we get Recall = 1, Precision = 0, F_1 = 0, Accuracy = 19% (macro average) 18% (micro average). Therefore, we use average of five random balance of testing (the number of positive and the number of negative is the same). In balance testing set, if we predict all testing document as negative, we get Recall = 0%, Precision = 0%, F_1 = 0, Accuracy $\approx$ 50% (macro average) $\approx$ 50% (micro average). If all testing documents are predicted as positive, we get Recall = 1, Precision=0, F_1 = 0, Accuracy $\approx$ 50% (macro average) $\approx$ 50% (micro average).

Table 3. Comparison of two-stage classifiers

Model	Macro-average		Micro-average	
	F_1	Acc	F_1	Acc
TSC	**0.44**	**0.69**	**0.57**	**0.71**
Two-stage: Rocchio-SVM	0.19	0.56	0.46	0.61
$\%change$	129.2%	22.3 %	22.4%	17.2%

The *t-test p* values in Table 4, indicate that the difference in scores is statistically significant.

Table 4. p-values for baseline models comparing to TSM in all accessing topics

	F_1	Acc
SVM	8.516E-12	9.569E-15
Rocchio	0.0317	0.045
TS Rocchio-SVM	4.323E-12	1.1E-09

Table 5. Experiment Result in Scoring Phase [6]

	top-20	*MAP*	F_1	*b/p*	*IAP*
SVM	0.45	0.41	0.42	0.41	0.44
$Rocchio$	0.47	0.43	0.43	0.42	0.45
RFD	**0.56**	**0.49**	**0.47**	**0.47**	**0.51**

Table 6. Recall-oriented stage one and precision-oriented stage two

Model	Macro-average		Micro-average	
	Rec	$Prec$	Rec	$Prec$
Stage 1: Rocchio	**0.82**	0.27	**0.85**	0.28
Stage 2: RFD$_\tau$	0.59	**0.39**	0.70	**0.42**

Some classifiers, such as SVM and Rocchio, use scoring then thresholding stages. The final performance of a classifier is based on both stage. In scoring stage, RFD scoring has better result than SVM and Rocchio (see Table 5), where *top-20* is the average precision of the top 20 documents, *MAP* is Mean Average Precision, *b/p* is the break-even point, and *IAP* is Interpolated Average Precision. With an efficient thresholding, scoring methods can maintain overall classifier's performance. Table 6 shows that our effective two-stage classifier is recall-oriented in stage one and precision-oriented in stage two.

6 Conclusions

Effective text classification is not a trivial process. In this paper, we proposed a new approach for text classification by using a pattern based two-stage classifier. We proposed a novel pattern based scoring and thresholding for stage two. The scoring model is based on pattern mining which capture semantic content of the text; and the thresholding model is based on training set that makes it efficient. Experiment shows that our proposed model is comparable to existing state of the art text classifiers.

References

1. Buckley, C., Voorhees, E.: Evaluating evaluation measure stability. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 33–40. ACM (2000)
2. Croft, W., Metzler, D., Strohman, T.: Search engines: Information retrieval in practice. Addison-Wesley (2010)
3. Gopal, S., Yang, Y.: Multilabel classification with meta-level features. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 315–322. ACM (2010)
4. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)

5. Lewis, D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 37–50. ACM (1992)
6. Li, Y., Algarni, A., Zhong, N.: Mining positive and negative patterns for relevance feature discovery. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 753–762. ACM (2010)
7. Li, Y., Zhong, N.: Mining ontology for automatically acquiring web user information needs. IEEE Transactions on Knowledge and Data Engineering 18(4), 554–568 (2006)
8. Li, Y., Zhou, X., Bruza, P., Xu, Y., Lau, R.Y.: A two-stage text mining model for information filtering. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1023–1032. ACM (2008)
9. Li, Y., Zhou, X., Bruza, P., Xu, Y., Lau, R.Y.: A two-stage decision model for information filtering. Decision Support Systems (2011)
10. Manning, C., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
11. Qiu, L., Zhao, R., Zhou, G., Yi, S.: An extensive empirical study of feature selection for text categorization. In: Proceedings of the Seventh IEEE/ACIS International Conference onComputer and Information Science, ICIS 2008, pp. 312–315. IEEE (2008)
12. Rocchio, J.: Relevance feedback in information retrieval. In: SMART Retrieval System Experimens in Automatic Document Processing, pp. 313–323 (1971)
13. Schapire, R., Singer, Y., Singhal, A.: Boosting and rocchio applied to text filtering. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 215–223. ACM (1998)
14. Schütze, H., Hull, D., Pedersen, J.: A comparison of classifiers and document representations for the routing problem. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 229–237. ACM (1995)
15. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34(1), 1–47 (2002)
16. Shen, D., Sun, J., Yang, Q., Zhao, H., Chen, Z.: Text classification improved through automatically extracted sequences. In: Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, pp. 121–121. IEEE (2006)
17. Soboroff, I., Robertson, S.: Building a filtering test collection for trec 2002. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 243–250. ACM (2003)
18. Wu, S., Li, Y., Xu, Y.: Deploying approaches for pattern refinement in text mining. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 1157–1161. Springer, Heidelberg (2006)
19. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval 1(1), 69–90 (1999)
20. Yang, Y.: A study of thresholding strategies for text categorization. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 137–145. ACM (2001)
21. Zhang, Y., Callan, J.: Maximum likelihood estimation for filtering thresholds. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 294–302. ACM (2001)

Applying a Lightweight Iterative Merging Chinese Segmentation in Web Image Annotation

Chuen-Min Huang and Yen-Jia Chang

Department of Information Management
National Yunlin University of Science & Technology, Taiwan, R.O.C.
{huangcm,g9923712}@yuntech.edu.tw

Abstract. Traditional CBIR method relies on visual features to identify objects in an image and uses predefined terms to annotate images, thus it fails to depict the implicit meanings. Recent textual content analysis methods applied to image annotation were blamed for their complexity of computation. In this research, we propose a corpus-free, relatively light computation of term segmentation method, namely "Iterative Merging Chinese Segmentation (IMCS) ," to identify representative terms from a single web page to obtain anecdotes as a semantic enrichment of the target image. It requires minimum computation needs that allows to share characters/words and facilitate their use at fine granularities without prohibitive cost. In the experiment, this method achieves a precision rate of 86.02%, and gains acceptance from expert rating and user rating of 75% and 68%, respectively. In performance testing, it only takes 0.006 second to process each image in a collection of 1,728 testing data set.

Keywords: Automatic Image Annotation, Iterative Merging Chinese Segmentation, N-Gram, Lexical Chain.

1 Introduction

With the rapid development of image technology and digital devices, more and more web images are stored and displayed on the Internet. As a result, the way to process these images to be retrieved efficiently has become a crucial issue. Content-Based Image Retrieval (CBIR) with its focus on rapid application of voluminous low-level visual features such as color, texture, shape, etc. gains popularity to support efficient searching and browsing images. Due to the visual features explain less semantics, and the annotation relies on limited predefined terms, thereby the results usually are not satisfactory.

A glimpse of related studies would reveal that a couple of supervised learning approach and clustering techniques have been applied to image annotation including Support Vector Machine (SVM) [1, 2], Bayesian [3] and Self-Organizing Feature Map (SOM) [4, 5]. In addition, various text processing techniques that support content identification to analyze textual content based on word co-occurrence, location, and lexical-chained concepts have been elaborated in [6-8]. However, the aforementioned techniques suffer the same disadvantages of heavy computation for

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 183–194, 2013.

multi-document processing, which will consume lots of memory and may incur run-time overhead. In this study, we propose a corpus-free, relatively light computation of term segmentation for single document processing, namely "Iterative Merging Chinese Segmentation (IMCS) method." The IMCS method is to identify representative terms from a string in a single document with minimum computation needs that allows to share characters/words and facilitate their use at fine granularities without prohibitive cost. The results showed that applying our method in a single document is enough to generate content descriptor for image annotation. The precision rate achieves 86.02%, and the acceptance from expert and user rating reach 75% and 68%, respectively.The performance testing is also very promising.

The rest of the paper is organized as follows. Section 2 presents the related work including automatic image annotation, text processing, and lexical semantics. Then, we address the design of our experiment in Section 3. The experimental result and evaluation method are described in Section 4. Finally, in Section 5, we draw some conclusions and propose future work.

2 Related Work

2.1 Automatic Image Annotation

There have been a number of models applied for image annotation. In general, image annotation can be categorized into three types: retrieval-based, classification-based, and probabilistic-based [6]. The basic notion behind retrieval-based annotation is that semantic-relevant images are composed of similar visual features. CBIR have been proposed in 1992 [9]. Since then, more and more studies annotated the images based on this method [10]. CBIR is applied by the use of images features, including shape, color and texture. However, this method is limited by the training data set and the hidden semantic or abstract concepts can't be extracted because the keywords are confined to pre-defined terms. Consequently, the results of CBIR are usually not satisfactory. The second type, also known as the supervised learning approach, treats annotation as classification using multiple classifiers. The images are classified based on the extracted features. This method processes each semantic concept as an independent class, and assigns each concept as one classifier. Bayesian [3] and SVM [11] are the most often used approaches. The third type is constructed by estimating the correlations between images and concepts with a particular emphasis on the term-term relationship and intends to solve the problem of "homograph" and "homophony." Frequent used approaches include co-occurrence model [12], LSA [5], PLSA [13] and HMM [14]. Notwithstanding the efforts made on the enhancement of annotation quality, the aforementioned approaches suffer the same disadvantages of complex processing of contents and semantics.

2.2 Text Processing

In the field of Information Retrieval, text processing technique was conducted to extract keywords or features that can represent document content. One of the most prominent problems in processing Chinese texts is the identification of valid words in a sentence, since there are no delimiters to separate words in a sentence, identifying words/phrases is difficult because of segmentation ambiguities and the occurrences of newly formed words. In general, Chinese texts can be parsed using dictionary lookup, statistical or hybrid approaches [15]. The dictionary lookup approach identifies key words of a string by mapping well-established corpus. The statistical technique extracts elements by using n-gram computation. The hybrid method conducts dictionary mapping to process the major task of word extraction and handle the remains through n-gram computation. For example, the Chinese string *Tsai ying wen bei jian tan bian hui mian yi xiao shi* 蔡英文北監探扁會面一小時 'Ying-wen Tsai went to Taipei prison to visit Shui-bian Chen for an hour' will be parsed as: ' [*Tsai ying wen*] [蔡英文], ' ' [*bei jian*] [北監],' '[*tan*] [探],' '[*bian*] [扁],' '[*hui mian*] [會面],' and '[*yi xiao shi*] [一小時]' by a dictionary-based CKIP Chinese word segmentation system (CKIP for short). This method is very efficient while it fails to identify newly-formed or out-of-the-vocabulary words and it is also blamed for the triviality of the list of the extracted words. The statistical technique is conducted by extracting each n-gram (bi-gram, tri-gram, … etc.) from the input string. This method relies on the frequency of each n-gram and a threshold to determine the validity of each word. The above string through n-gram segmentation will produce: '[蔡英], [英文], [文北], [北監], [監探], [探扁], [扁會], [會面], [面一], [一小], and [小時];' '蔡英文, 英文北, ..., 一小時; ' and so on. The application of this method has the benefit of corpus-free and the capability of extracting newly-formed or out-of-the-vocabulary words while at the expense of huge computations and the follow-up filtering efforts. Recently, a number of studies proposed substring [9], significant estimation [16], and relational normalization [17, 18] to identify words based on statistical calculations.

The following task after word extraction is to assign each word a weight, or value, reflecting its presumed importance for purposes of content identification. Probably the most popular and well-implemented algorithm in traditional computing for term weighting is TFIDF (Term Frequency - Inverse Document Frequency) [19]. TFIDF regards the importance of a word is proportional to the standard occurrence frequency of each word k in each document i (that is, FREQik) and inversely proportional to the total number of documents to which each word is assigned. The term discrimination value can be used to compute a weight for each word in each document of a collection by combining the term frequency factor with the discrimination value. Some studies [20] [21] proposed methods to assign different weights to words by location.

2.3 Lexical Semantics

Lexical cohesion can be interpreted as the state of cohering for making the sentences of a text, indicated by the use of semantically related vocabulary. Lexical chains (LCs) are sequences of words which are in lexical cohesion relation with each other and they tend to indicate portions of a text that form semantic units; they could serve further as a basis for a segmentation [22]. This method is usually applied in a summarization generation [23]. For instance, the string *xiang liang kong jian mo xing*

向量空間模型 'Vector space model' may be parsed as '[*xiang liang*] [向量] ,' '[*kong jian*] [空間] ,' and '[*mo xing*] [模型]' if there is no further merging process undergoing. Thereby the most informative word *xiang liang kong jian mo xing* 向量空間模型 will be left out. Usually LCs are constructed in a bottom-up manner by taking each candidate word of a text, and finding an appropriate semantic relation offered by a thesaurus. Instead, the paper [24] proposes a top-down approach of linear text segmentation based on lexical cohesion of a text. Some scholars suggested to use machine learning approaches to create a set of rules to calculate the rate of forming a new word by characters for entity recognition including the maximum entropy model (MEM) and hidden Markov model (HMM) and claimed this method was able to achieve reasonable performance with minimal training data [25, 26].

3 Research Design

3.1 Research Model Overview

This study covers three tasks: text processing, image annotation, and image evaluation. The image-resided web pages were collected from Taiwan news website and the framework of our research is depicted as Fig. 1. We used the news title and the text as input data; the image captions as ground truth labels. Then, we conducted IMCS and term weighting for the input data. After that, we generated a list of three representative words for each image. From the list, we assigned the word with the highest weight as the primary annotation and the rest as secondary annotations. Finally, we evaluated the primary and secondary annotations by using the image caption and human judgment, respectively. In the following section, we will introduce the process of IMCS and the way of term weighting and the word annotation.

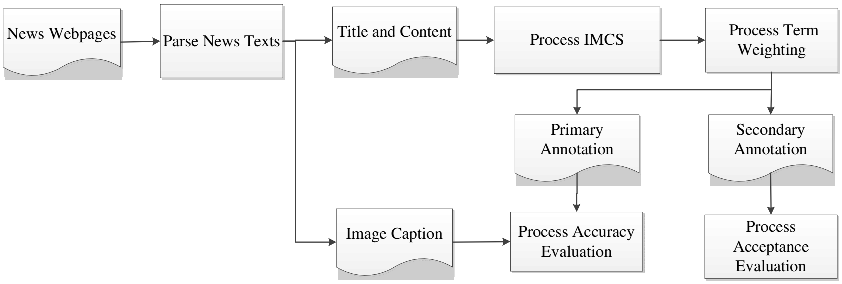


Fig. 1. Research Model

3.2 Iterative Merging Chinese Segmentation, IMCS

The fundamental idea of the IMCS was a hybrid of n-gram and LCs processing. Each gram is regarded as a Chinese character. We will examine the probability distribution rate of each character from a string with its chained characters. A chain-like iterative merging to form compounds for lexical consideration will be employed. Physically, the same character may appear in different locations in the text structure. Logically there is only one character for every occurrence of a given character in the text. Because the number of different characters processed is less than the number of characters in the document, the total number of characters is substantially less than what an implementation would use by the merging. The iterative merging process is discussed with an example as stated below.

3.2.1 Step 1: Build up a Directed Graph

The string: *Tsai fan chi: yu chang an pu shih pi an, tse mo yi chih wen, yu chang an kuo fa chi chin wei he yi tsai fang chi chuan li* 蔡反擊：宇昌案不是弊案，怎麼一直問，宇昌案國發基金為何一再放棄權利, 'Tsai fired back against the comments and stated that Yu Chang is not a scandal, why did you keep asking for and why the National Development Fund abandons its rights repeatedly?' will produce a directed graph as Fig. 2. After removing duplicate characters and replacing a punctuation with a new line, each character will form a vertex and an edge with an arrow extending from itself to its chained characters (vertices). In this example, it is clear that the character "一" appears twice in the string, while there is only one vertex to present it in the graph.

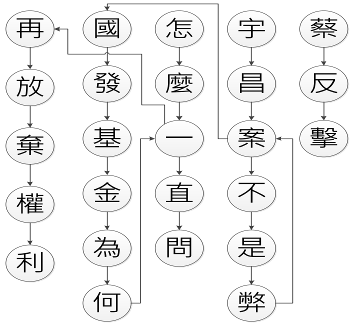


Fig. 2. The graph of Step 1 of IMCS

3.2.2 Step 2: Calculate the Probability Distribution of Edges

In a graph, two vertices form an edge are said to be the endpoints of it and the edge is said to be incident to the vertices. This step is to calculate the degree of a vertex in a graph. The degree of a vertex in a graph is the number of edges incident to it. In this example, the vertex "案" points to both the vertex "不" and the vertex "國" therefore the probability of the edges "案→不" and "案→國" is 50% (or 0.5) for each, shown as Fig. 3. In the figure, the expression “percent” in [percent, digit] represents the average probability distribution of a vertex to the other vertices and “digit” means the number of links of a vertex to all its connected vertices in the string.

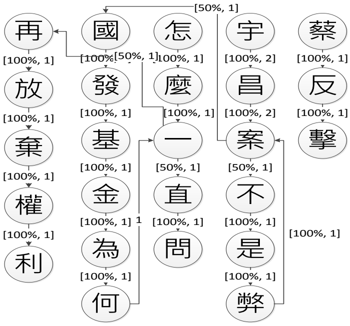


Fig. 3. The graph of Step 2 of IMCS

3.2.3 Step 3: Concatenate Vertices

To determine whether two vertices can be concatenated, it depends on the criteria listed in (1).

$$(\Pr(i,j) \geq DISC) \wedge (|LC(i,o)|) > 1$$

and (1)

$$DISC = \frac{1}{\log SN}, \; \mathit{if} \;\; SN > 10 \,; \quad DISC = 1 \;\; , \; \mathit{otherwise}$$

where $\Pr(i,j)$ represents the probability of the concatenation from a vertex i to a vertex j; the discriminator $DISC$ is the threshold value; $|LC(i,o)|$ means the number of lexical chains extending from a vertex i to all its connected vertices; SN represents the number of segments of the string. The discriminator is used as a way of normalizing the length of the string, thereby reducing the divergence terms in a lengthy document. For example, since there are four segments of the string "[蔡反擊]：[宇昌案不是弊案]，[怎麼一直問]，[宇昌案國發基金為何一再放棄權利]," the value of the threshold will be 1. The probability of "宇→昌" and "昌→案" is 100% (or 1) for each and that meets the requirement; thereby these two chains will be concatenated as the vertices [宇昌] and [昌案], respectively as shown in Fig. 4. Our data sets showed that the average number of segments in a news document is around 60-70. The concatenation process performs better when we set the threshold value near 0.5. Each concatenated vertex is regarded as a compound unit which will proceed to its next links and establish new edges and recalculate the probabilities of the vertices "宇昌→案," "昌案→不," and "昌案→國."

3.2.4 Step 4: Run Iteration

The above steps will be iterated until no concatenation can be found, and this iteration process will generate a series of short and long LCs from the string. A long LC is believed to be more content representative than a short LC could possibly be. In this example, it is obvious that "宇昌案" is a better content indicator than either "宇昌" or

"昌案." To reduce the possibility of extracting less representative LCs from the concatenation, we will take a post-processing as the last step to finalize the IMCS.

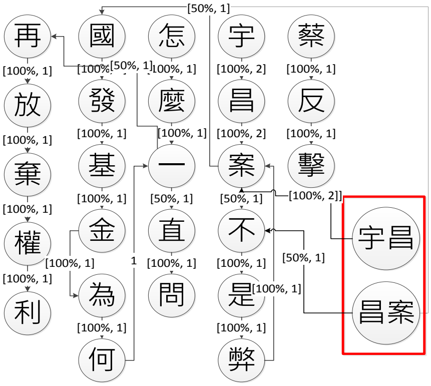


Fig. 4. The graph of Step 3 of IMCS

3.2.5 Step 5: Execute Post-processing

In the final step, significant words are determined by observing the information mutually shared by two overlapped LCs using the following significance estimation (SE) function as (2).

$$SE_i = \frac{f_i}{f_a + f_b - f_i} \tag{2}$$

where i denotes the LC_i to be estimated, i.e., $i = i_1 i_2 \ldots i_n$; a and b represent the two longest compound substrings of LC_i with the length n-1, i.e., $a = i_1 i_2 \ldots i_{n-1}$ and $b = i_2 i_3 \ldots i_n$ As for f_a, f_b and f_i are the frequencies of a, b, and i, respectively. For example, the term i "宇昌案" shall gain the SE value of 0.83 with its frequency 6 of substring a “與∨” as well as the frequency 6 of its substring "宇昌 (f_a)" and the value 5 of the other substring "昌案 (f_b). " We will retain both term i and remove term a if $f_i = f_a$, because the full term is more representative than its substring. Likewise, we will remove b if $f_i = f_b$. In the example above, since the frequency of "宇昌案 (f_i) "is equal to "宇昌 (f_a)," so we remove "宇昌 (f_a)"; but "宇昌案 (f_i)" is less than "昌案 (f_b)," so we retain both terms.

3.3 Term Weighting

It is suggested that the obvious place where appropriate content identifiers might be found in a news is the title and the first paragraph. In addition, we also considered frequency and the length of a word as the indicators of word significance in a document. Given a word LC_i, the term weighting algorithm may be defined as (3).

$$Weight_i = tf_i \times (val_1 + val_2 + length_i)$$

and (3)

$$val1 = \begin{cases} 2, word_i \in title \\ 0, otherwise \end{cases} \quad val2 = \begin{cases} 2, word_i \in firstparagraph \\ 0, otherwise \end{cases}$$

where tf_i represents the frequency of LC_i; val_1 and val_2 express the extra weight of LC_i based on its position in the news report. If LC_i appears in the title or the first paragraph, it gains an extra weight of 2 respectively.

4 Evaluation

In this experiment, we collected 1,738 image-resided web pages from Taiwan news website as the data sets. To verify our proposed IMCS method can successfully identify the content representation for image annotation, we used image captions served as ground truth labels to see whether the generated primary annotation is included in the list. We invited three experts to assess the appropriateness of the secondary annotations (the second and the third high scores of a word list) with respect to the image content. In the end, we also measured the performance of the IMCS method in a real-time mode.

4.1 Evaluation of Primary Annotation

Since an image caption is written by a journalist, it is assumed that a man-made caption would be faithful to an image scenario. Therefore, we considered image captions as the ground truth labels which will be used to evaluate the accuracy rate of the generated image annotations from title and text. If our generated primary annotation matches a substring of the image caption, we will be confident to assure that the IMCS method works well as anticipated.

Due to the problem of semantic ambiguity in part of Chinese words, where the interpretation of image annotations may vary from user to user, therefore the exact number of correct annotations of an image will not be clearly identified. For example, the string *zong tong ma ying jiu* 總統馬英九 'President Ma Ying-jiu' is meant to be regarded as a single lexical word, therefore it may not be appropriate to segment it into [總統 'President'] and [馬英九 'Ma Ying-jiu'] even though these two words are valid. Consequently, the recall measurement did not apply to this study. A precision measurement was used to understand the proportion of primary annotations actually matched the image captions as (4).

$$p = \frac{number \quad of \quad matched \quad PAs \quad in \quad captions}{total \quad number \quad of \quad PAs} \tag{4}$$

where PAs represent the generated primary annotations from 1,738 documents. After the IMCA processing, the total number of matched PAs in captions are 1,495. We obtained a precision rate of 86.06%.

4.2 Evaluation of Secondary Annotation by Experts

To evaluate the validity of the secondary annotations, we invited three experts to participate in the assessment. Sixty pieces of news were randomly selected from which we extracted the second and the third high scores LCs and produced 120 annotations. To reduce ambiguous judgements, each annotation was evaluated based on a method of dichotomic classification to which the annotation represents the image content. Each expert could only select either "agree" or "disagree" for each annotation. The result showed that the number of check marks of consent is 270 out of 360. It implies that the agreement of the appropriateness of the secondary annotations to the images achieves 75%.

4.3 Evaluation of Secondary Annotation by Users

To evaluate the appropriateness of the secondary annotation, we conducted another survey to understand the differences between the image annotation and the users' expectation. Sixty-three graduate and undergraduate students were recruited from the National Yunlin university of Science & Technology, Taiwan to participate in the assessment. Thirty pieces of news were randomly selected from which we extracted the second and the third high scores LCs and produced 60 annotations. To assist the assessment, we provided news title, texts and caption for references. Each annotation was evaluated by 63 subjects to understand the degree to which the annotations appropriately address the image content. The result in Table 1 shows that the number of check marks of agreement is much higher than that of disagreement. The agreement rate of user evaluation reaches 68.2%.

Table 1. Statistics of Results of user satisfaction

	Highly Agree	Agree	Average	Disagree	Highly Disagree	Total
Number	1490	1088	748	288	166	3780

4.4 Performance Testing

After the validity and acceptance evaluation, we conducted a performance testing with respect to the time spent of processing from an event trigger to system response. Often real-time response times are understood to be in milliseconds and sometimes microseconds. Our testing data sets consist of 1,738 pieces of news, the processing time is 10.45 seconds in total with 0.006 seconds on average for each news.

5 Conclusion

In this paper, we propose a corpus-free, relatively light computation of term segmentation for single document processing, namely "Iterative Merging Chinese Segmentation (IMCS) method" to identify representative terms for image annotation. The IMCS method is based on a hybrid of n-gram and lexical chain processing for image annotation. Unlike previous techniques suffer the problems of heavy computation for multi-document processing, which will consume lots of memory and may incur unacceptable run-time overhead. Our method considers an input document as a string composed of a series of lexical chains with varied lengths. This purpose of this study is to extract these lexical chains in a precise way with minimum computation needs. Each iteration automatically calculates the probability of lexical chain between two vertices and creates concatenate compound units. This processing allows to share characters/words and facilitate their use at fine granularities without prohibitive cost. Results showed that this method achieves a precision rate of 86.02%, and gains acceptance from expert rating and user rating of 75% and 68%, respectively. In performance testing, it only takes 0.006 second to process each image in a collection of 1,728 testing data set.

Even though the IMCS method is only applied to single-document processing in the current study, the results showed that it is enough to generate content descriptor for image annotation. With the benefit of corpus-free and lightweight features, it can also be easily embedded in any text processing application. Furthermore, it would be interesting to implement this algorithm to multi-document processing and test its performance. Other researchers may verify our study using a larger data set or compare with the state of the art algorithms. It is hoped that our research will invite more perspectives on automatic image annotation.

References

[1] Gao, S., et al.: Automatic image annotation through multi-topic text categorization. Presented at the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (2006)

[2] Lei, Z., Jun, M.: Image annotation by incorporating word correlations into multi-class SVM. Soft Computing 15, 917–927 (2011)

[3] Luong-Dong, N., et al.: A Bayesian approach integrating regional and global features for image semantic learning. In: Proceedings of the IEEE International Conference On Multimedia, pp. 546–549 (2009)

[4] Chow, T.W.S., Rahman, M.K.M.: A new image classification technique using tree-structured regional features. Advanced Neurocomputing Theory and Methodology 70, 1040–1050 (2007)

[5] Huang, C.M., et al.: Automatic image annotation by incorporating weighting strategy with CSOM classifier. Presented at the The 2011 International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV 2011), Monte Carlo Resort, Las Vegas, Nevada, USA (2011)

[6] Su, J.H., et al.: Effective image semantic annotation by discovering visual-concept associations from image-concept distribution model. In: Proceedings of the IEEE International Conference On Multimedia, pp. 42–47 (2010)

[7] Barnard, K., et al.: Matching words and pictures. The Journal of Machine Learning Research 3, 1107–1135 (2003)

[8] Zhu, S., Liu, Y.: Semi-supervised learning model based efficient image annotation. IEEE Signal Processing Letter 16, 989–992 (2009)

[9] Kato, T.: Database architecture for content-based image retrieval. In: Proc. SPIE 1662, Image Storage and Retrieval Systems, pp. 112–123 (1992)

[10] Jing, L., et al.: Automatic image annotation based-on model space. In: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering 2005, pp. 455–460 (2005)

[11] Gao, Y., et al.: Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers. Presented at the Proceedings of the 14th Annual ACM International Conference on Multimedia, Santa Barbara, CA, USA (2006)

[12] Mori, Y., et al.: Image-to-word transformation based on dividing and vector quantizing images with words. Presented at the First International Workshop on Multimedia Intelligent Storage and Retrieval Manegement (1999)

[13] Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. Presented at the Proceedings of the 12th Annual ACM International Conference on Multimedia, New York, NY, USA (2004)

[14] Carneiro, G., Vasconcelos, N.: Formulating semantic image annotation as a supervised learning problem. Presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2005)

[15] Wang, Z., et al.: Word segmentation of Chinese text with multiple hybrid methods. Presented at the 2009 International Conference on Computational Intelligence and Software Engineering (2009)

[16] Horng, J.T., Yeh, C.C.: Applying genetic algorithms to query optimization in document retrieval. Information Processing & Management 36, 737–759 (2000)

[17] Kim, M.S., et al.: Structural optimization of a full-text n-gram index using relational normalization. The VLDB Journal 17, 1485–1507 (2008)

[18] Fuketa, M., et al.: A retrieval method of similar strings using substrings. Presented at the 2010 Second International Conference on Computer Engineering and Applications (2010)

[19] Teng, C., et al.: A behavioural mode research on user-focus summarization. Mathematical and Computer Modelling 51, 985–994 (2010)

[20] Yan, H., et al.: Compressing term positions in web indexes. Presented at the Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA (2009)

[21] Troy, A.D., Zhang, G.-Q.: Enhancing relevance scoring with chronological term rank. Presented at the Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands (2007)

[22] Tatar, D., et al.: Text Segments as Constrained Formal Concepts. In: 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 223–228 (2010)

[23] Shanthi, V., Lalitha, S.: Lexical chaining process for text generations. Presented at the International Conference on Process Automation, Control and Computing (PACC) (2011)

[24] Tatar, D., et al.: Lexical Chains Segmentation in Summarization. In: 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 95–101 (2008)

[25] Chiong, R., Wang, W.: Named entity recognition using hybrid machine learning approach. Presented at the The 5th IEEE International Conference on Cognitive Informatics (2006)

[26] Ageishi, R., Miura, T.: Named entity recognition based on a Hidden Markov Model in part-of-speech tagging. In: Presented at the First International Conference on the Applications of Digital Information and Web Technologies (2008)

Relevance as a Metric for Evaluating Machine Learning Algorithms

Aravind Kota Gopalakrishna[1], Tanir Ozcelebi[1],
Antonio Liotta[1,2], and Johan J. Lukkien[1]

[1] System Architecture and Networking (SAN),
Department of Mathematics and Computer Science,
[2] Electro-Optical Communications,
Department of Electrical Engineering,
Eindhoven University of Technology, The Netherlands
{a.kota.gopalakrishna,t.ozcelebi,a.liotta,j.j.lukkien}@tue.nl

Abstract. In machine learning, the choice of a learning algorithm that is suitable for the application domain is critical. The performance metric used to compare different algorithms must also reflect the concerns of users in the application domain under consideration. In this paper, we propose a novel probability-based performance metric called *Relevance Score* for evaluating supervised learning algorithms. We evaluate the proposed metric through empirical analysis on a dataset gathered from an intelligent lighting pilot installation. In comparison to the commonly used Classification Accuracy metric, the Relevance Score proves to be more appropriate for a certain class of applications.

Keywords: Machine learning algorithms, performance metric, probabilistic approach.

1 Introduction

One of the general goals of a machine learning (ML) algorithm is to capture complex relationships and patterns within a dataset and to apply such knowledge on new data that is believed to have the same (or a similar) pattern. The success criteria for the ML algorithm depend on many factors, most importantly, on the user concerns that are specific to the application domain.

Consider a car navigation system that utilizes a specific ML algorithm to select the best route from a source location to a destination. There are many ways to measure the performance of this algorithm objectively. Useful metrics can be, for instance, the actual time to reach the destination, the amount of gas consumption and the amount of tolls paid. However, it is often the case that a combination of these metrics cannot directly be mapped to the real satisfaction of the driver from the navigation experience, whereas this is what matters in the end. Assume that there are only two home-office routes possible and the driver *generally* likes to take the first route, which is shorter, faster and safer. On the other hand, *sometimes*, the driver likes to take the second route that goes along

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 195–208, 2013.

the sea shore, especially when it is sunny and the driver is in the mood. The ML algorithm can be expected to learn to pick the desired route among the two only if weather is a parameter that can be and is monitored by the navigation device. Monitoring the mood of a person is very difficult nonetheless. Alternatively, the ML algorithm will simply start looking for patterns within other data that is available to it, e.g. the time of the day, which may or may not have any actual impact. In this case, the desired route may also depend on those parameters, i.e. features, which are relevant yet not considered by the navigation device.

It is a difficult task to identify *all* the features that are relevant for the ML algorithm employed by an application. The features that are relevant for the application define a *context*. Even when the context is known, it may be technically very challenging or very intrusive to monitor some features especially when users are involved, e.g. the driver's mood in the previous example. The subset of relevant features that are monitored define an *observed context*. Depending on the application under consideration and the observed context, sometimes evaluating an output, i.e. a prediction made by the ML algorithm, as either *right* or *wrong* might not be sufficient as there are gray areas in between. For example, the intelligent environment described in [1] offers lighting services (light settings) depending on several contextual factors such as user identity, user activity and time of the day. When the dataset collected from this intelligent environment is examined, there are two important remarks to be made. The first remark is that the desired light settings are not fixed for a given instance of observed context. Since perception of lighting also depends on subjective factors that cannot be monitored, even the same user, the same activity and the same time of the day may result in different desired light settings. The second remark is that, for a given observed context, there are acceptable (desired at least once) and unacceptable light settings (never desired).

This means that the overall performance of the ML algorithm may be hindered by the fact that the observed context is not entirely representative of the actual context of an application. When this is the case, it is a challenge to evaluate and compare the performances of different ML algorithms in the application domain. In the literature, different evaluation metrics assess different characteristics of ML algorithms [2]. Classification problems in ML are broadly categorized into multi-class classification and multi-label classification problems. Multi-class classification algorithms [3] are for those categories of problems where, for a given input instance, there is exactly one correct output class, which is selected out of several classes. Examples of multi-class classification include image recognition and diagnosis [4]. The commonly used evaluation metric for multi-class classification problems are accuracy, precision and recall. Multi-label classification algorithms [5] address that category of problems where multiple output classes must be selected for each input instance. Examples of multi-label classification include text [6] and music categorization [7]. The commonly used evaluation metrics for multi-label classification problems are hamming-loss, precision and recall.

As illustrated by the examples of car navigation and intelligent lighting, there also exists another class of applications. In these applications, the ML prediction model has to select a unique output class for a given input instance (as in multi-class classification), and the output for a given input instance may fall into multiple acceptable output classes (as in multi-label classification). An interesting property of such an application is randomness in the output, i.e. there is no single acceptable outcome for a given observed context but there is a statistical regularity associated with different possible and acceptable outcomes. In such cases, even though the prediction models that are used to solve multi-class classification problems suits the need, the evaluation metrics such as accuracy, precision and recall are not suitable. For instance, the use of accuracy as a metric would lead to a score of zero for a misclassified sample, which should not be the case. The bad choice of performance metrics would produce results that are meaningless to achieve the design goal of an application [8] [9].

In this paper, we devise a new evaluation scheme to evaluate the performance of supervised learning algorithms for this class of applications and propose a new evaluation metric called *Relevance Score (RS)*. We consider an intelligent lighting scenario to describe and formulate the problem. *Relevance Score* is defined as a percentage measure of how much relevant an output or outcome is, as predicted by the prediction model for a given observed context. The score is based on the probabilistic distances among the different outcomes, i.e. predicted and actual outcome. We use this metric to evaluate several rule based prediction models and then compare it with the commonly used metric *Prediction Accuracy or Classification Accuracy (CA)* to validate the significance of the proposed metric.

The paper is organized as follows. Section 2 discusses the drawbacks of using the commonly used *CA* metric with an example of intelligent lighting application and identifies improvement directions. Section 3 presents the proposed relevance metric. Section 4 provides an evaluation of the proposed metric. Finally, Section 5 concludes the paper.

2 Problem Definition

In this section, we discuss the drawbacks of *Classification Accuracy (CA)* as a metric for evaluating supervised learning algorithms, which motivates the need for a new metric using an intelligent lighting application scenario.

Let $x_i \in X$ denote the i^{th} sample in a dataset, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and X represents an n-dimensional input feature space given by $X = X_1 \times X_2 \times X_3 \times \ldots \times X_n$. Let us denote the output class label as y where $y \in Y = \{y_1, y_2, \ldots, y_K\}$ and K denotes the number of possible outcomes. In the intelligent lighting application under consideration [1], there are 8 possible presets for output lighting conditions (K=8) in a breakout area, as shown in the three dimensional space in Fig. 1. Each octant represents a possible light combination (e.g. static-warm-dim).

A user is provided with a lighting condition based on six input features ($n = 6$) namely user-identity, type of activity, area of activity, number of users, time of the day and the external light influence.

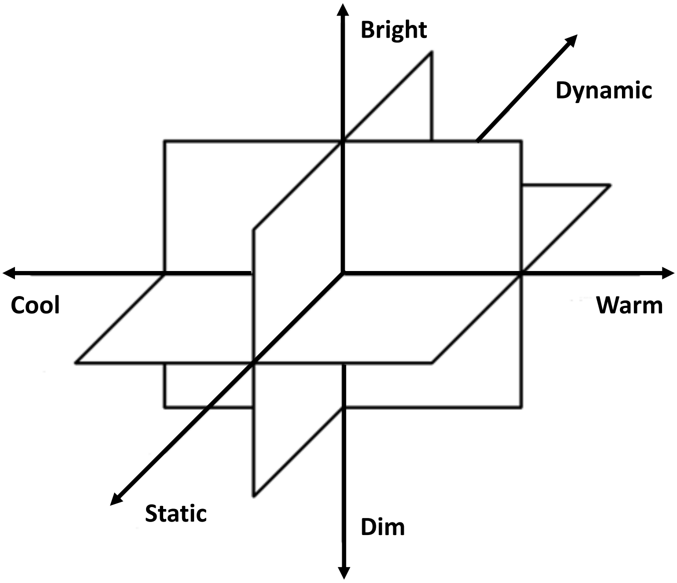


Fig. 1. Possible Output Light Combinations

In [1], the supervised learning approach is explored where a considered classification model is trained on the breakout dataset to predict a unique light output $y_i \in Y = \{y_1, y_2, \ldots, y_8\}$ for a given input $x_i = (x_{i1}, x_{i2}, \ldots, x_{i6})$. The success of an intelligent lighting application depends on selecting the best suitable prediction model that selects the desired lighting condition for a given observed context. The performance of the selected models is evaluated using CA as a metric.

Let us assume that a classification model h is trained using the breakout dataset. The breakout dataset contains a sample x 10 times and the output light condition selected by a user is light A *(LA)* four times, light B *(LB)* four times and light C *(LC)* two times. This means that the user for a given input instance x has selected LA, LB and LC with probabilities 0.4, 0.4 and 0.2 respectively, i.e. a sample x has multiple output classes. Assume that the classifier h selects LA based on what it has learnt from the training data, whereas the user actually desires LB as the lighting condition. Here the classifier h is not right with the prediction, but not entirely wrong either. If CA is used as a metric, then the selection made by the classifier h would be assessed as completely wrong. From the application point-of-view the CA metric is not very representative as it is important to measure how relevant the lighting condition is for a given observed context, rather than to measure how accurate it is. It can never be 100% accurate as the user is not consistent in choosing the desired lighting condition, based on the observed context.

From the example above it is evident that CA is not a relevant metric for similar applications as it fails to capture the degree of relevance when there is a

Table 1. Sample comparison of metrics

Actual Output	Predicted Output	CA Metric	Alternative Metric
LA	*LA*	100	Good match (100)
LB	*LC*	0	Decent match (75)
LC	*LA*	0	Irrelevant match (0)
LA	*LB*	0	Relevant match (50)
LB	*LB*	100	Good match (100)
		200/5 = 40%	325/5 = 65%

"non-ideal" prediction. Hence, it is necessary to have an evaluation mechanism that is able to compute the distance between actual and predicted outcome when they are not equal. An example of such a metric for a given classifier is shown in Table 1.

As shown in Table 1, by virtue of the problem, *CA* gives a low accuracy of 40%, whereas the user is given relevant outputs 80% of the time and irrelevant outputs only 20% of the time according to the *Alternative Metric*, resulting in an average score of 65%. Here the *Alternative Metric* computes the relevance of the outcome i.e. how good is the output selection made by a prediction model through the data statistics available from the observed environment, thereby making it more suitable than the *CA* metric.

Formally, we propose an evaluation metric named as *Relevance Score*, which is more suitable for the third class of applications mentioned in Section 1 than the commonly used CA metric. In particular, we seek to find a performance function $f : (X, Y) \to R_{+}^{100}$ where $R_{+}^{100} = \{RS \in R | 0 \leq RS \leq 100\}$.

3 Proposed Metric

In this section, we propose a new mechanism to evaluate supervised prediction models used in the presented class of applications, e.g. in intelligent lighting. Here, we formulate a function that quantifies the relevance of the predicted outcome when there is a mismatch between the predicted and actual outcome.

The well known CA metric is computed as an average of the sum of individual accuracies for k test set samples as in equation 8.

$$CA = \frac{1}{k} \sum_{i=1}^{k} Acc_i \tag{1}$$

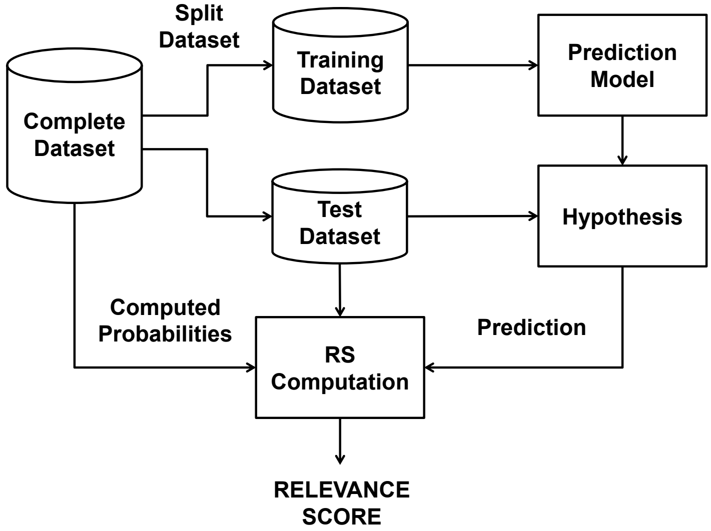


Fig. 2. Relevance Score computation procedure

We inherit the same mechanism to compute the *Relevance Score* on a test set for a given prediction model with a minor modification where the accuracy *Acc* for a sample is replaced by *Score* as in equation 2 and computed as in equation 8.

$$RS = \frac{1}{k}\sum_{i=1}^{k} Score_i \tag{2}$$

We divide *RS* computation into two phases: 1. Probability Computation Phase, and 2. Evaluation Phase. The entire process of computing *RS* is shown in Fig. 2.

3.1 Probability Computation Phase

An *ErrScore* is a real number that is calculated when there is an error in prediction by the prediction model. It is computed based on five parameters, the predicted outcome (O_P), the actual outcome (O_A), the probability of occurrence of the predicted outcome ($P(O_P)$), the probability of occurrence of the actual outcome ($P(O_A)$) and the probability of the most frequently selected outcome ($P(O_H)$). These probabilities are the values that are computed from the entire dataset. The probability values are computed for the different output class labels ($y \in Y$) for every input instance (x_i) in the breakout dataset. Formally, compute $P(y|x_i)$ for $i = 1, 2, 3, ..., m$ where m is the number of samples in the entire dataset. In this phase, the values of the most probable feature that is responsible for the randomness in output (e.g. user identity feature in intelligent lighting application) is removed for generalization.

Table 2. Qualitative Relevance

CASE	OUTCOME	PROBABILITIES	QUALITATIVE RELEVANCE
1	$O_P = O_A$	-	Highly Relevant
2	$O_P \neq O_A$	$P(O_H) = P(O_P); P(O_P) > P(O_A)$	Moderately Relevant
3	$O_P \neq O_A$	$P(O_H) > P(O_P) > P(O_A)$	Relevant
4	$O_P \neq O_A$	$P(O_H) > P(O_A) > P(O_P)$	Less Relevant
5	$O_P \neq O_A$	$P(O_H) = P(O_A); P(O_A) > P(O_P)$	Irrelevant

3.2 Evaluation Phase

In this phase, the predictions on the test data are evaluated using the considered classification models. If d denotes the probabilistic distances in general, then the distances between the probabilities are given by,

$$d_{HP} = |P(O_H) - P(O_P)| \tag{3}$$

$$d_{PA} = |P(O_P) - P(O_A)| \tag{4}$$

$$d_{HA} = |P(O_H) - P(O_A)| \tag{5}$$

Several cases are possible for a given input instance based on the predicted and the actual outcomes and their computed probabilities. The cases may be classified qualitatively in terms of relevancy of the outputs provided. The decreasing order of relevance we consider for different cases is summarized in Table 2. The ordering is based on the consequence of individual cases as explained later. Thus, *ErrScore* is a score that is obtained by quantifying these cases.

Case 1: $O_P = O_A$

In this case, for a given input instance in the test set, the predicted outcome and the actual outcome are equal. Thus, there is no error in the predicted outcome.

Case 2: $O_P \neq O_A; P(O_H) = P(O_P) > P(O_A)$

In this case, for a given input instance the predicted outcome and the highest probable outcome are equal but a different output (actual outcome) is selected. Since the predicted outcome is equal to the highest probable outcome ($d_{PH} = 0$) and the classification model has not been able to capture the switch in output, the error is kept minimum and is equal to $\beta \cdot d_{PA}$ where β is a positive real constant, whose value depends on the application.

Case 3: $O_P \neq O_A; P(O_H) > P(O_P) > P(O_A)$

In this case, for a given input instance the predicted outcome, the actual outcome and the highest probable outcome are not equal. Since the probability of the predicted outcome lies in between that of the highest probable outcome and the actual outcome, the error value is higher than the previous case, and is equal to $(\alpha \cdot d_{HP} + \beta \cdot d_{PA})$ where α and β denote positive real constants.

Case 4: $O_P \neq O_A; P(O_H) > P(O_A) > P(O_P)$

In this case, for a given input instance the predicted outcome, the actual outcome and the highest probable outcome are not equal. Since the probability of the predicted outcome lies farther away than that of the highest probable outcome and the actual outcome, the error value is much higher, and equal to $(\alpha \cdot d_{HP} + \beta \cdot d_{PA})$ as in Case 3.

Case 5: $O_P \neq O_A; P(O_H) = P(O_A) > P(O_P)$

In this case, for a given input instance the actual outcome and the highest probable outcome are equal but the classification model selects a different output (predicted outcome). Since the actual outcome is equal to the highest probable outcome and the performance of the classification model was poor to select a different output, the error rate is much higher than that of the previous cases. The error is equal to $(\alpha + \beta) \cdot d_{HP}$.

Combining above equations and normalizing over $(\alpha+\beta)$, we find the following error score function

$$ErrScore = \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta}. \tag{6}$$

Therefore, the RS for a sample is computed as below

$$Score = (1 - ErrScore) \times 100, \tag{7}$$

$$Score = (1 - \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta}) \times 100. \tag{8}$$

4 Experimentation and Results

In this section, we study the performance of the proposed RS metric and compare it to the CA metric for different classifiers used in the breakout dataset. We further investigate the influence of α and β parameters of the RS metric function.

The performance of RS metric is studied through various experimentations using the Weka [10] simulator on the following rule-based prediction models: DecisionTable [11], JRip [12], Nearest Neighbor with generalization (NNge) [13], PART [14], ConjunctiveRule [15] and Ridor [16] on the breakout dataset [1]. The breakout dataset consists of 236 samples of data from different users gathered from the breakout area. Here 70% of the dataset is used for training and the remaining 30% is used as test set. We consider 10 randomly shuffled training and test sets for experimentation and then mean the results to avoid biased results. We perform the following investigations:

1. Comparison of RS and CA Metrics
2. Significance of α and β
3. Lower and Upper Bounds of RS with changing α and β
4. Testing of RS on Random Output Data

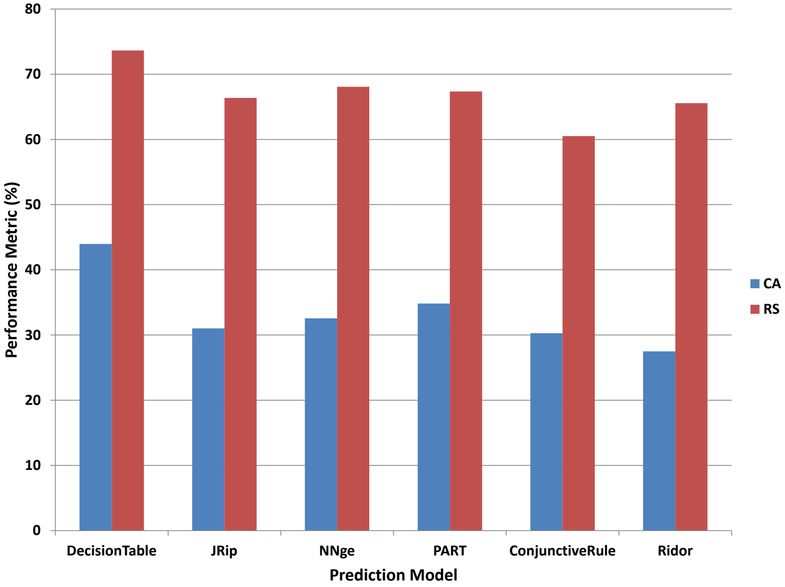


Fig. 3. Performance Metric vs. Prediction Model

4.1 Comparison of RS and CA Metrics

Fig. 3 shows the graph of Performance Metric vs. Prediction Model for two different metrics: *CA* and *RS*. The RS values are computed with $\alpha = 2$ and $\beta = 1$. As discussed earlier, low accuracy values are achieved with the *CA* metric. This means that no prediction models provide exact lighting conditions based on the input instances in more than 50% of the test samples. However, with RS as a metric, a maximum of 73% relevant lighting condition is achieved with DecisionTable prediction model.

An interesting observation is that the prediction model Ridor has lower *CA* than Conjunctive Rule, whereas Ridor has a better *RS* value than Conjunctive Rule. This is interpreted as follows. For a portion of the data samples, the most desired outcome (the best match) is calculated for each prediction model and ConjunctiveRule performs better than Ridor, i.e. it has higher classification accuracy. However, when it comes to the relevance of the predicted outcomes, the Ridor is more successful than ConjunctiveRule, i.e. it predicts the most relevant outcomes on average. Even though ConjunctiveRule finds the best match slightly more often than Ridor, when it cannot find the best match, it comes up with less relevant predictions. Therefore, Ridor is preferable for applications where relevance of the predicted outcome is more critical than accuracy. As a consequence, for such applications, *RS* as a performance metric is more relevant.

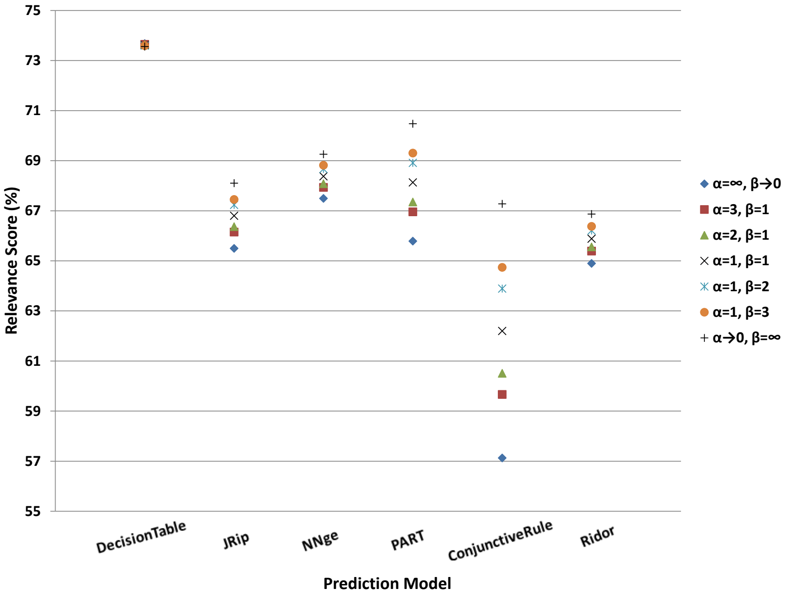


Fig. 4. Relevance Score vs. Prediction Model

4.2 Significance of α and β

In this experiment, we investigate the influence of varying values of α and β on RS. Fig. 4 shows the plot of RS vs. Prediction Model for different values of α and β. The graph shows that the prediction model DecisionTable is more consistent with the changing (α, β) whereas the RS values improve with α decreasing and β increasing for other prediction models. The parameters α and β are used as weights for probabilistic distances d_{HP} and d_{PA}, respectively. Thus, when α is higher the product $\alpha \cdot (d_{HP})$ is emphasized more, i.e. the probabilistic distance between the predicted and actual outcome is given less importance. When β is higher, the product $\beta \cdot (d_{PA})$ is emphasized more, i.e. the probabilistic distance between the most probable outcome and the predicted outcome is given less importance.

Fig. 5 shows the plot of RS vs. α, β for different prediction models. From the graph, we find that the predictions made by ConjunctiveRule model are closer to the actual outcome and hence varying α, β has more influence on the RS values. The low RS values for the ConjunctiveRule are due to the fact that the model is not good in consistently predicting relevant outcomes for the given scenarios.

4.3 Lower and Upper Bounds of RS with changing α and β

Maximum or minimum RS that a prediction model can reach is found by taking either α or β to be very large.

When α is very large compared to β, the error is computed is as follows:

$$ErrScore = \lim_{\alpha \to \infty} \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta} = \lim_{\alpha \to \infty} \frac{\alpha(d_{HP})}{\alpha} = d_{HP}. \quad (9)$$

When β is very large compared to α, the error is computed is as follows:

$$ErrScore = \lim_{\beta \to \infty} \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta} = \lim_{\beta \to \infty} \frac{\beta(d_{PA})}{\beta} = d_{PA}. \quad (10)$$

The lower and the upper bounds on RS for the considered prediction models are summarized in Table 3.

Table 3. Summary of maximum and minimum *RS* values for various prediction models on the breakout dataset

	$\alpha \to \infty$	$\beta \to \infty$
DecisionTable	73.68	73.56
JRip	65.50	68.10
NNge	67.50	69.26
PART	65.79	70.48
ConjunctiveRule	57.13	67.28
Ridor	64.90	66.87

4.4 Testing of RS on Random Output Data

In this test, we study the *RS* results on the same dataset with randomly generated output. One of the main motivations for this research was the fact that the performance of the prediction model should not be punished for inconsistencies in the data pattern that are caused by factors outside the observed context. For example, a prediction model whose goal is to find the desired lighting conditions for the user can be expected to perform at best as good as the user himself. If the user is generating inconsistent data, i.e. if an observed context maps to different outcomes at different times, then the student (the prediction model) will be learning from an inconsistent teacher (the user), whose reasons for changing mind are completely hidden from the student, i.e. the reasons either don't exist or they are beyond the observed context. Consider the extreme case where there is no relation between the input instances and output lighting conditions, i.e. the user throws an 8-sided dice to select the desired lighting condition and the dice is not part of the observed context. In this case, the CA metric would be equal to 1/8 as there are 8 possible lighting conditions. We also expect to achieve a lower RS in comparison to the scores on the dataset with real output. On the other hand, note that the user makes no distinction between lighting conditions. For a given observed context, even though there is exactly one desired lighting

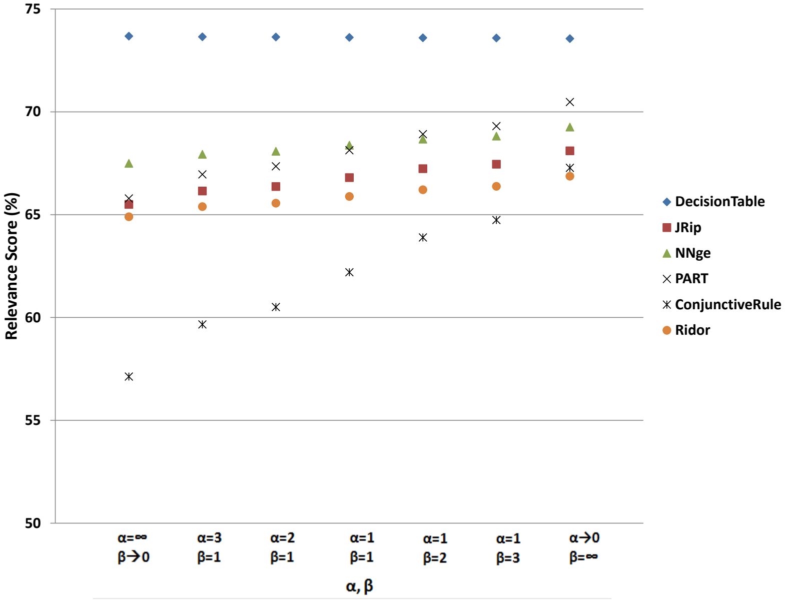


Fig. 5. Relevance Score vs. (α, β)

condition, all possible lighting conditions are indeed somewhat relevant. Therefore, we expect that the RS metric will be higher than the CA metric. Fig. 6 shows the plot of RS vs. prediction model for random output data and for the real output data from the breakout dataset.

In the random output dataset, since one of the eight light conditions is chosen at random i.e. with probability 0.125, the output probability distributions are almost uniform for the input instances. This makes it difficult for the prediction models to select the right output choice. Thus all the considered prediction models lie between the range 50 and 60%. But in case of real output dataset, there are some relationships between the input and output causing the output probability distribution to be non-uniform. Thus the prediction model DecisionTable performs the best with a RS of 74%.

From the performed studies, we find that RS is a more suitable metric than the commonly used metrics for the mentioned class of applications. Furthermore, the RS metric provides a provision to select the prediction models that have different characteristics in predicting the outcomes. This depends on the application requirement, and is done by varying parameters α and β. For example, a higher value of α is chosen as compared to β (say $\alpha = 10$ and $\beta = 1$) for applications where capturing inconsistencies in the outcome for an observed context is not so critical. A small value of α is chosen as compared to β (say $\alpha = 1$ and $\beta = 100$) for applications where it is necessary to select a prediction model that is highly sensitive to the inconsistencies in the outcome.

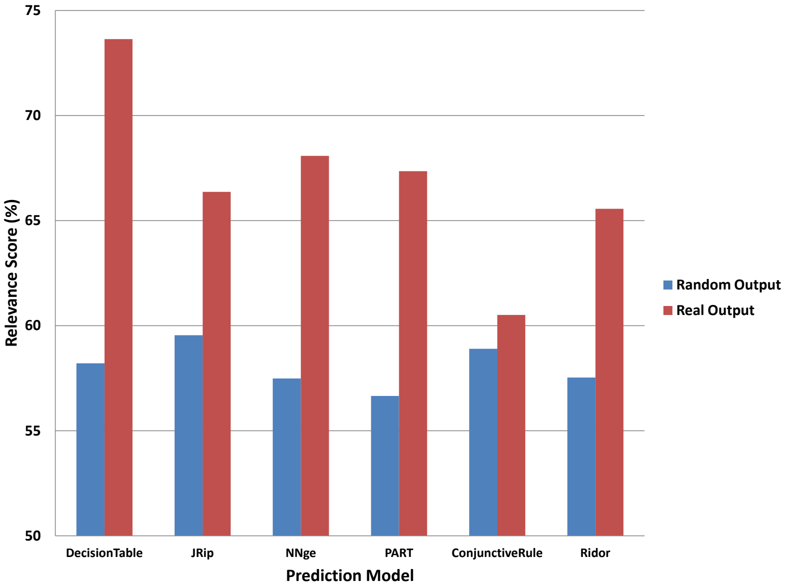


Fig. 6. Relevance Score vs. Prediction Model

5 Conclusion

The choice of a suitable performance metric is critical to select a machine learning algorithm for intelligent applications. In this direction, we presented the drawback of the commonly used evaluation metric *CA* with an illustration of intelligent lighting and similar class of applications. We further proposed a new metric named as *Relevance Score* to evaluate machine learning algorithms when there is more than one choice of output class labels suitable for an input instance. The metric is based on the probability values of the outcomes computed from the dataset for a given observed context. We also presented a detailed analysis on the performance and relevance of the proposed metric *RS* through various experiments on the breakout dataset. From the obtained results, we conclude that *RS* is a better performance metric for the class of applications where the relevance of predictions is critical rather than the prediction accuracy. Moreover, variants of RS computation can be used by carefully selecting the parameters of the RS computation function based on the application requirement.

Acknowledgments. We would like to acknowledge Smart Context-aware Services (SmaCS) project through the Point One grant No. 10012172, for supporting this work. We would also like to thank Serge Offermans, Intelligent Lighting Institute (ILI), TU/e for providing the breakout dataset.

References

1. Gopalakrishna, A.K., Ozcelebi, T., Liotta, A., Lukkien, J.J.: Exploiting Machine Learning for Intelligent Room Lighting Applications. In: The 6th International Conference on Intelligent Systems, Sofia, pp. 406–411 (2012)
2. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006)
3. Aly, M.: Survey of Multiclass Classification Methods. Technical Report, California Institute of Technology (2005)
4. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, ch. 1, pp. 21–26 (2005)
5. Tsoumakas, G., Katakis, I.: Multi-Label Classification: An Overview. J. Data Warehousing and Mining 3(3), 1–13 (2007)
6. McCallum, A.K.: Multi-label Text Classification with a Mixture Model trained by EM. In: AAAI 1999 Workshop on Text Learning (1999)
7. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-Label Classification of Music into Emotions. In: Proceedings of The International Society for Music Information Retrieval, pp. 325–330 (2008)
8. Japkowicz, N.: Why question machine learning evaluation methods? In: AAAI 2006 Workshop on Evaluation Methods for Machine Learning, pp. 6–11 (2006)
9. Jain, R.: The Art of Computer System Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling. John Wiley (1991)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
11. Kohavi, R.: The Power of Decision Tables. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 174–189. Springer, Heidelberg (1995)
12. Cohen, W.H.: Fast Effective Rule Induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123. Morgan Kaufmann (1995)
13. Martin, B.: Instance-based Learning: Nearest Neighbor with Generalization. Technical Report, University of Waikato (1995)
14. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 144–151 (1998)
15. Clarke, P., Niblett, T.: The CN2 Rule Induction Algorithm. In: Machine Learning, pp. 261–283 (1989)
16. Gaines, B.R., Compton, P.: Induction of ripple-down rules applied to modeling large databases. Journal of Intelligent Information Systems 5(3), 211–228 (1995)

Preceding Rule Induction with Instance Reduction Methods

Osama Othman[1] and Christopher H. Bryant[2]

[1] King Abdullah II School for Information Technology,
Jordan University, Amman Jordan
Othmano@gi-group.net
[2] School of Computing, Scienc and Engineering, Newton Building,
The University of Salford, Greater Manchester, M5 4WT, England, UK
C.H.Bryant@salford.ac.uk

Abstract. A new prepruning technique for rule induction is presented which applies instance reduction before rule induction. An empirical evaluation records the predictive accuracy and size of rule-sets generated from 24 datasets from the UCI Machine Learning Repository. Three instance reduction algorithms (Edited Nearest Neighbour, AllKnn and DROP5) are compared. Each one is used to reduce the size of the training set, prior to inducing a set of rules using Clark and Boswell's modification of CN2. A hybrid instance reduction algorithm (comprised of AllKnn and DROP5) is also tested. For most of the datasets, pruning the training set using ENN, AllKnn or the hybrid significantly reduces the number of rules generated by CN2, without adversely affecting the predictive performance. The hybrid achieves the highest average predictive accuracy.

Keywords: Rule Induction, Overfitting, Noise Filtering, Instance Reduction.

1 Introduction

Our work concerns the use of pruning to solve one of the most important problems in the field of machine learning, namely, overfitting which affects the predictive accuracy. We say the produced classifier overfits the data if we can find a different classifier with more error over training examples but smaller error over test data. Overfitting occurs in two situations: when the training set contains noisy instances and when the training set is not a representative sample from the instance space [19]. Both of these situations are common in real world applications.

The aim of our work is to investigate whether overfitting can be reduced by preceding rule induction with instance reduction. We focus on instance reduction methods which have proved capable of reducing the size of training set and resulted in the smallest reduction in predictive accuracy [29], [28]. More specifically, we will apply algorithms that try to remove the border instances, which tend to be noisy instances or hard-to-learn, untypical instances [10].

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 209–218, 2013.

The paper is organized as follows. Section 2 reviews the typical methods for rule pruning. Section 3 reviews the instance reduction techniques we use in this work. In Section 4, we discuss the results of pre pruning for rule induction using CN2 in terms of predictive accuracy and number of generated rules. Section 5 presents our conclusions.

2 Rule Pruning

A variety of methods has been proposed to prune the produced rule sets, and can be categorized into:

- Prepruning: These algorithms either use heuristics (i.e., stopping criteria.) to relax the constraint of completely satisfying the training instances, or reduce the number of training examples before generating the classifier[9], in the hope that using fewer training examples will produce fewer rules.

- Post Pruning: Initially introduces a rule set that is consistent with training instances, and then the rule set is examined to remove rules and conditions that do not reflect true regularities of the domain. Examples of post pruning algorithms include REP (Reduced Error Pruning algorithm) [2], GROW [5], SSRR [20] and hybrid and incremental post pruning techniques [24].

- Integration Pre Pruning and Post Pruning: Instead of learning the entire rule set and then applying the pruning, this category prunes each rule immediately after it has been learned. Examples of such algorithms are IREP (Incremental Reduced Error Pruning) [11], RIPPER [6], SLIPPER [7] and IREP++ [8].

The aim of our work is to empirically investigate whether pre pruning for rule induction can eliminate some of the produced classification rules while retaining the same level of predictive accuracy.

Pre pruning for rule induction can be achieved in two ways:

1- Condition reductions: pruning each rule independently in the course of learning by using a heuristic to determine when to stop adding conditions to the rule.

2- Rule Pruning: trying to reduce the number of produced rules by either

    a. Removing the most specific produced rules (hopefully that cover the noisy instances from typing or measurement errors).
    b. Reducing the instances used to build the rules.

Previous research on pre pruning focused on simplifying the rules during induction. There is a case study which investigated the effect of a new noisy instance detection method before induction on specific dataset (i.e early diagnosis of rheumatic diseases) [9], and the suggested method is suitable for datasets with just two classes. Grudzinski et al. concentrated on the EkP system [14] as instance reduction method before rule induction, and they illustrated it is possible to extract simpler sets of rules from reduced datasets [13]. However no one has investigated the effect of preceding rule induction with instance reduction methods, in terms of predictive accuracy and number of generated rules.

We will apply some instance pruning methods that have been proven to maintain the predictive accuracy and reduce the size of training set. We will investigate whether applying the rule induction methods to the pruned training set reduces the number of classification rules without adversely affecting the predictive accuracy.

3 Instance Reduction Technique

Instance pruning tries to prune the original training set to get a smaller subset of it. Searching for a subset S of instances to keep instead of the original training set T can proceed in variety of directions, including: incremental, decremental and batch [28].

Incremental methods begin with empty subset S, and add instances (from training set T) to subset S if it fulfills some criteria. Thus if new instances are made available later (after training is completed) they can continue to be added to S according to the same criteria. Incremental methods are sensitive to the order of presentation of the instances. Condensed Nearest Neighbor (CNN) [15] and Selective Nearest Neighbor (SNN) [22] are examples of Incremental methods. On the other hand, decremental methods begin with all the instances in the training set (i.e. T=S), and search for instances to remove; they are often computationally more expensive than incremental methods. Reduced Nearest Neighbor (RNN) [12] and Decremental Reduction Optimization Procedure (DROP1-5) [29] represent examples of decremental methods. Finally, batch methods, as decremental methods, begin with all instances in training set, but before they remove any, they find all of the instances that meet the removal criteria and then they remove them all at once [25]. Batch methods also suffer from increased time complexity compared with incremental methods. In our experiments, we will use decremental and batch methods because, in comparison to incremental methods, they have been shown to give rise to higher predictive accuracies [29].

Instance reduction methods can be categorized as retaining either internal or border instances:

- Border instances: the intuition for retaining border instances is that internal instances do not affect the decision boundaries and thus can be removed with relatively little effect on classification.
- Internal instances: seek to remove border instances, and hopefully removes instances that are noisy.

In our experiments, we focus on three reduction algorithms that performed well in reducing the number of instances [28], and provided good results before applying Neural Network learning [10]. These algorithms eliminate border instances which tend to be noisy instances or hard to learn untypical instances.

3.1 The Edited Nearest Neighbor Algorithm

Edited Nearest Neighbor ENN [27] is decremental algorithm which removes an instance if it does not agree with the majority of its k nearest neighbor (with k= 3).

This removes noisy instances as well as near border instances and retains all internal instances. Figure 1 shows the pseudo code for ENN algorithm.

3.2 AllKnn

AllKnn [25] is batch algorithm which makes k iteration, at the ith iteration; it flags as bad any instance that is not classified correctly by its i nearest neighbors. After completing all iterations, the algorithm removes all instances flagged as bad. Figure 2 shows the pseudo code for AllKnn algorithm.

```
For each instance (i)
        If (the class of instance (i) <> the majority class of k neighbors)
            Remove the Instance
```

Fig. 1. Pseudo-code for ENN algorithm

```
oldk = k
For each instance (i)
    For k=1 till oldk
        If (the class of instance (i) <> the majority class of k neighbors)
              Flag the Instance for pruning
Remove each flagged instance
```

Fig. 2. Pseudo-code for AllKnn algorithm

```
Let T be the initial set of instances
Measure the distance of each instance in T from its nearest enemy (instance with different class). Sort the instances in T by their distance, in ascending order.
Let S = T.
 For each instance P in S:
         Find $P.N_{1..k+1}$, the k+1 nearest neighbors of P in S.
         Add P to each of its neighbors' lists of associates.
 For each instance P in S:
         Let with= # of associates of P classified correctly with P as a neighbor.
         Let without= # of associates of P classified correctly without P.
         If without >= with
                 Remove P from S.
                 For each associate A of P
                         Remove P from A's list of nearest neighbors.
                         Find a new nearest neighbor for A.
                         Add A to its new neighbor's list of associates.
 Return S.
```

Fig. 3. Pseudo-code for DROP5 algorithm

3.3 DROP5

DROP5 [29] is decremental algorithm which removes the instance "S" if at least as many of its associates (instances have "S" on their nearest neighbor list) are classified correctly without it. It considers removing first the instances that are nearest to their nearest enemy (i.e instance from different class), and proceeding outward. By removing points near the decision boundary first, the decision boundary is smoothed. Figure 3 shows the pseudo code for DROP5 algorithm.

4 Empirical Results for CN2 Using the Reduced Set

Using the noise filtering methods to reduce the border instances before applying the induction method can avoid the overfitting problem. That may improve the predictive accuracy for the induction method. El Hindi and Alakhras (2009) showed that filtering out border instances before training artificial neural network will improve the predictive accuracy and speed up the training process by reducing the training epochs.

The CN2 [4] algorithm induces an ordered list of classification rules from examples, using entropy as its heuristic. Then Clark and Boswell improved CN2 by using a Laplacian error estimate as alternative evaluation function and producing unordered classification rules [3]. Our objective is to apply some Instance reduction methods before applying the modified CN2 algorithm and compare the results with and without applying the reduction

4.1 Method

We applied the three methods for instance reduction (AllKnn, ENN and DROP5) that are intended to remove the border and noisy instances before using the CN2. We also apply DROP5 [29] method on instances flagged by AllKnn to be removed and we call this method as AllKnnDROP5 method.

To test if these methods will affect the accuracy of the CN2 algorithm, we conducted experiment on a collection of Machine Learning data sets available from the repository at University of California at Irvine [18]. Predictive accuracy was estimated using 10-fold cross-validation [16]. Instance-removal was performed separately for each fold of the cross-validation.

4.2 Results

Table 1 shows the results obtained using the four prepruning methods with respect to the predictive accuracy. Our experiments show that applying the AllKnnDrop5 algorithm is generally better than applying the other pruning methods with respect to predictive accuracy. Also the results reveal that the predictive accuracy increased on 13 datasets when applying AllKnnDrop5, on 11 datasets when applying AllKnn and on 10 datasets after using the ENN. However the predictive accuracy increased on only 4 datasets after using the DROP5. On average we can see that CN2 after using

AllknnDROP5 and AllKnn has better predictive accuracy than CN2 without pruning. This shows that applying a prepruning technique on training set before applying the rule induction can reduce the overfitting problem that can adversely effect on predictive accuracy.

Table 2 shows that all of the instance reduction techniques reduce the number of rules generated by CN2. We can see that DROP5 achieved the largest reduction. Applying AllKnnDrop5 and AllKnn reduces the generated rules by 44% and 48% on average respectively.

From our results we can state that applying instance reduction techniques as a prepruning process for rule induction will reduce the number of generated rules and not adversely affecting the predictive accuracy and may improve it in some cases

Table 1. Empirical results comparing predictive accuracy for using AllKnn ENN, DROP5 and AllKnnDrop5 prepruning

Data Sets	*Without pruning*	*ENN*	*AllKnn*	*DROP5*	*AllKnnDrop5*
Iris	89.98	92.00	92.67	80.67	93.34
Voting	95.34	95.10	95.33	85.35	95.57
Vowels	67.11	65.97	66.75	85.07	67.31
heart (C)	80.66	76.66	77.33	71.66	79.34
Glass	64.76	58.05	61.98	51.92	66.22
Liver disorders	66.77	64.11	65.64	60.3	66.52
Wine	91.77	94.11	93.52	70	95.28
Pima Indians Diabetes	74.61	73.69	76.34	73.82	76.18
Promoters	85.00	81.00	80.00	63	80.00
Hepatitis	78.65	80.00	80.00	52.67	79.34
Vehicle	57.85	60.10	60.71	54.99	60.10
pole-and-cart	61.68	63.88	66.24	62.56	63.51
Blood Transfusion Service Center	75.68	76.61	76.35	73.11	75.96
Ecoli	79.10	83.31	80.91	73.34	80.90
Soybean	86.32	82.67	83.01	63	83.32
ZOO	92.00	87.00	90.00	81	89.00
Yeast	48.98	55.47	56.43	51.82	56.56
Led Creator	72.30	72.30	71.30	68.9	71.90
vertebral_column	80.96	83.21	81.28	81.28	82.24
Ionosphere	89.43	85.71	86.56	53.71	85.71
Wave	69.70	70.38	70.74	67.96	71.38
Balance Scale	75.30	74.70	74.34	67.1	74.34
Letter recognition	70.52	69.50	67.91	58.87	69.69
Average	***76.28***	***75.89***	***76.32***	***67.48***	***76.68***

Table 2. Empirical results comparing generated rules for using AllKnn ENN, DROP5 and AllKnnDrop5 prepruning**.**

		ENN		***AllKnn***		***DROP5***		***AllKnnDROP5***	
Data Sets	R_{CN2}	R_{ENN}	R_{ENN} / R_{CN2}	R_{AllKnn}	R_{AllKnn} / R_{CN2}	R_{DROP5}	R_{DROP5}/ R_{CN2}	$R_{AllKnnDROP5}$	$R_{AllKnnDROP5}$/ R_{CN2}
Iris	*6.30*	*3.9*	*0.62*	*3.6*	*0.57*	*3*	*0.48*	*3.6*	*0.57*
Voting	*17.3*	*6.2*	*0.36*	*5.7*	*0.33*	*3*	*0.17*	*6.1*	*0.35*
Vowels	*46.2*	*42.2*	*0.91*	*41.5*	*0.9*	*31.7*	*0.69*	*44.3*	*0.96*
heart (C)	*21.3*	*11.2*	*0.53*	*9.4*	*0.44*	*7*	*0.33*	*10.6*	*0.5*
Glass	*22.0*	*12.8*	*0.58*	*12.1*	*0.55*	*9.2*	*0.42*	*10.3*	*0.47*
Liver disorders	*31.3*	*17.6*	*0.56*	*15.2*	*0.49*	*12.6*	*0.4*	*18.1*	*0.58*
Wine	*8.60*	*7.4*	*0.86*	*6.9*	*0.8*	*3*	*0.35*	*6.9*	*0.8*
Pima Indians Diabetes	*44.4*	*20.8*	*0.47*	*18.1*	*0.41*	*15.6*	*0.35*	*21.3*	*0.48*
Promoters	*12.4*	*10.4*	*0.84*	*9.6*	*0.77*	*2.7*	*0.22*	*9.7*	*0.78*
Hepatitis	*17.8*	*1.80*	*0.1*	*4.2*	*0.24*	*1.7*	*0.1*	*4.7*	*0.26*
Vehicle	*48.4*	*29.3*	*0.61*	*25.9*	*0.54*	*27.2*	*0.56*	*29.3*	*0.61*
pole-and-cart	*109.8*	*56.9*	*0.52*	*46.7*	*0.43*	*51.7*	*0.47*	*50.8*	*0.46*
Blood Transfusion Service Center	*61.2*	*13.0*	*0.21*	*11.9*	*0.19*	*13.2*	*0.22*	*16.5*	*0.27*
Ecoli	*24.7*	*12.7*	*0.51*	*10.5*	*0.43*	*7.7*	*0.31*	*12.3*	*0.5*
Soybean	*32.7*	*15.9*	*0.49*	*24.8*	*0.76*	*21.3*	*0.65*	*27.2*	*0.83*
ZOO	*8.70*	*6.1*	*0.7*	*6.3*	*0.72*	*6.2*	*0.71*	*6.3*	*0.72*
Yeast	*121.2*	*40.7*	*0.34*	*37.0*	*0.31*	*40.5*	*0.33*	*47.3*	*0.39*
Led Creator	*79.9*	*21.8*	*0.27*	*19.9*	*0.25*	*23.4*	*0.29*	*24.3*	*0.3*
vertebral_column	*16.7*	*10.4*	*0.62*	*9.1*	*0.54*	*6.9*	*0.41*	*10.1*	*0.6*
Ionosphere	*17.6*	*6.5*	*0.37*	*7.2*	*0.41*	*4.9*	*0.28*	*9.7*	*0.55*
Wave	*204.8*	*118.0*	*0.58*	*102.3*	*0.5*	*60.3*	*0.29*	*111.6*	*0.54*
Balance Scale	*150.1*	*75.4*	*0.5*	*63.0*	*0.42*	*21.6*	*0.14*	*65.2*	*0.43*
Letter recognition	*263.8*	*232.5*	*0.88*	*228.0*	*0.86*	*173.8*	*0.66*	*233.0*	*0.88*
Average	***59.4***	***33.6***	***0.54***	***31.3***	***0.52***	***23.83***	***0.38***	***33.9***	***0.56***

5 Conclusion

In this paper, we have mentioned different instance reduction techniques, and applied them as preprocessing before CN2 algorithm. Our experiments showed that for most datasets, pruning the training set using AllKnn, ENN or AllKnnDrop5 significantly reduces the number of rules generated by CN2 without adversely affecting the predictive performance. Also applying AllKnnDrop5 gave the best result with respect to predictive accuracy on average. Other instance reduction algorithms, such as C-Pruner [30], conduct instance pruning more carefully, so as to avoid deleting important instances. For future work, we recommend comparing them with ENN, AllKnn and DROP5, to investigate whether the technique of preceding rule induction with instance reduction can be further improved.

Only one rule-induction algorithm was used in our experiments. To investigate how generic the technique is, instance reduction should be applied as a pre-processing step before using other rule induction algorithms and the effect on number of generated rules and prediction accuracy observed.

El Hindi and Alakhras (2009) investigated using instance reduction on Neural Network and they reported good results. We recommend testing instance reduction with other types of classifiers like decision trees.

Acknowledgments. We would like to express our sincere gratitude to Dr. Khalil el Hindi for the useful comments, remarks and the suggested ideas which led us to investigate instance pruning as pre pruning technique for rule induction.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance – based learning algorithm. Machine Learning 6, 37–66 (1991)
2. Brunk, C., Pazzini, M.: An investigation of noise-tolerant relational concept learning algorithms. In: Proceedings of the 8th International Workshop on Machine Learning, Evanston, Illinois, pp. 389–393 (1991)
3. Clark, P., Boswell, R.: Rule induction with CN2: some recent improvements. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 151–163. Springer, Heidelberg (1991)
4. Clark, P., Niblett, T.: The CN2 induction algorithm. Machine Learning 3, 261–283 (1989)
5. Cohen, W.: Efficient pruning methods for separate-and-conquer rule learning systems. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 988–994. Morgan Kaufmann, Chambery (1993)
6. Cohen, W.: Fast effective rule induction. In: Prieditis, A., RussellIn, S.J. (eds.) Machine Learning: Proceedings of the 12th International Conference, vol. 3, pp. 115–123. Morgan Kaufmann, Lake Tahoe (1995)
7. Cohen, W., Singer, Y.: A simple, fast and effective rule learner. In: Hendler, J., Subramanian, D. (eds.) Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 335–342. AAAI/MIT Press, Menlo Park (1999)

8. Dain, O., Cunningham, R., Boyer, S.: IREP++ a faster rule learning algorithm. In: Michael, W., Dayal, U., Kamath, C., Davis, B. (eds.) Proceeding Fourth SIAM Int. Conf. Data Mining, Lake Buena Vista, FL, USA, pp. 138–146 (2004)
9. Gamberger, D., Lavrac, N., Dzeroski, S.: Noise Elimination in inductive concept learning: A case study in medical diagnosis. In: Arikawa, S., Sharma, A.K. (eds.) ALT 1996. LNCS, vol. 1160, pp. 199–212. Springer, Heidelberg (1996)
10. El Hindi, K., Alakhras, M.: Eliminating border instance to avoid overfitting. In: dos Reis, A.P. (ed.) Proceeding of Intelligent Systems and Agents 2009, pp. 93–99. IADIS press, Algarve (2009)
11. Fürnkranz, J., Widmer, G.: Incremental reduced error pruning. In: Cohen, W., Hirsh, H. (eds.) Proceedings of the 11th International Conference on Machine learning (ML 1994), pp. 70–77. Morgan Kaufmann, New Brunswick (1994)
12. Gates, G.W.: The reduced nearest neighbor rule. Institute of Electrical and Electronics Engineers Transactions on Information Theory 18(3), 431–433 (1972)
13. Grudziński, K., Grochowski, M., Duch, W.: Pruning Classification Rules with Reference Vector Selection Methods. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part I. LNCS, vol. 6113, pp. 347–354. Springer, Heidelberg (2010)
14. Grudzinski, K.: EkP: A fast minimization – based prototype selection algorithm. In: Intelligent Information System XVI, pp. 45–53. Academic Publishing House EXIT, Warsaw (2008)
15. Hart, P.E.: The condensed nearest neighbor rules. Institute of Electrical and Electronics Engineers Transactions on Information Theory 14(3), 515–516 (1968)
16. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Mellish, C. (ed.) Proceedings of 14th International Joint Conference on Artificial Intelligence, pp. 1137–1143. Morgan Kaufmann, San Francisco (1995)
17. Lukasz, A., Krzysztof, J.: Highly scalable and robust rule learner: performance evaluation and comparison. IEEE Transactions on Systems, Man, and Cybernetics, Part B 36(1), 32–53 (2006)
18. Murphy, P.M., Aha, D.W.: UCI repository of Machine Learning Data bases. available by anonymous ftp to ics.uci.edu in the pub/machine-learning-databases directory (1994)
19. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
20. Othman, O., El Hindi, K.: Rule reduction technique for RISE algorithm. Advances in Modeling, Series B: Signal Processing and Pattern Recognition 47, 2 (2004)
21. Pham, D.T., Bigot, S., Dimov, S.: A rule merging technique for handling noise in inductive learning. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 218 (C), 1255–1268 (2004)
22. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An Algorithm for a Selective Nearest Neighbor Decision Rule. IEEE Transactions on Information Theory 21(6), 665–669 (1975)
23. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. In: Bartlett, P.L., Mansour, Y. (eds.) Proceeding COLT 1998 Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 80–91. ACM press, New York (1998)
24. Shehzad, K.: Simple Hybrid and Incremental Post-Pruning Techniques for Rule Induction. IEEE Transactions on Knowledge and Data Engineering (99), 1–6 (2011)
25. Tomek, I.: An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics 6(6), 448–452 (1976)

26. Weiss, S., Indurkhya, N.: Reduced complexity rule induction. In: Mylopouslos, J., Reiter, R. (eds.) Proceedings of 12th International Joint Conference on Artificial Intelligence, pp. 678–684. Morgan Kauffmann, Sydney (1991)
27. Wilson, D.L.: Asymptotic properties of nearest neighbor rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics 2(3), 408–421 (1972)
28. Wilsson, D.R., Martinez, T.R.: Instance Pruning Technique. In: Fisher, D.H. (ed.) Machine Learning: Proceedings of the Fourteenth International Conference (ICML 1997), pp. 403–411. Morgan Kauffmann, San Francisco (1997)
29. Wilsson, D.R., Martinez, T.R.: Reduction techniques for instance based learning algorithms. Machine Learning 38(3), 257–286 (2000)
30. Zhao, K.P., Zhou, S.G., Guan, J.H., Zhou, A.Y.: C-Pruner: An improved instance pruning algorithm. In: Proceedings of the 2th International Conference on Machine Learning and Cybernetics, Sheraton Hotel, Xi'an, China, vol. 1, pp. 94–99. IEEE, Piscataway (2003)

Analytic Feature Selection for Support Vector Machines

Carly Stambaugh[1], Hui Yang[2], and Felix Breuer[1,⋆]

[1] Department of Mathematics
[2] Department of Computer Science
San Francisco State University
1600 Holloway Avenue,
San Francisco, CA 94132
cstambau@mail.sfsu.edu, huiyang@sfsu.edu, felix@fbreuer.de

Abstract. Support vector machines (SVMs) rely on the inherent geometry of a data set to classify training data. Because of this, we believe SVMs are an excellent candidate to guide the development of an analytic feature selection algorithm, as opposed to the more commonly used heuristic methods. We propose a filter-based feature selection algorithm based on the inherent geometry of a feature set. Through observation, we identified six geometric properties that differ between optimal and suboptimal feature sets, and have statistically significant correlations to classifier performance. Our algorithm is based on logistic and linear regression models using these six geometric properties as predictor variables. The proposed algorithm achieves excellent results on high dimensional text data sets, with features that can be organized into a handful of feature types; for example, unigrams, bigrams or semantic structural features. We believe this algorithm is a novel and effective approach to solving the feature selection problem for linear SVMs.

1 Introduction

Support Vector Machines (SVMs) are kernel-based machine learning classifiers [1]. Using optimization methods such as quadratic programming, SVMs produce a hyperplane that separates data points into their respective categories. When a new, unlabeled, data point is introduced, its position relative to the hyperplane is used to predict the category the new point belongs to. One of the most important aspects of any machine learning classification problem is determining the particular combination of variables, or features, within a data set that will lead to the most accurate predictions, which is commonly known as the feature selection problem. Currently the methods used by most machine learning engineers are heuristic in nature, and do not depend heavily on intrinsic properties of the data set [2]. Due to the geometric nature of an SVM, it is natural to suggest that

⋆ Felix Breuer was supported by Deutsche Forschungsgemeinschaft (DFG) grant BR 4251/1-1.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 219–233, 2013.

the performance of a particular feature set may be tied to its underlying geometric structure. This structure-performance relationship has in turn motivated us to develop an analytically driven approach to the feature selection problem for linear SVMs.

The primary goal of this research is to identify underlying geometric properties of optimal feature sets, and use these properties to create a feature selection algorithm that relies solely on the inherent geometry of a particular feature set. To accomplish this, we first create n-dimensional point clouds to represent known optimal and suboptimal feature sets. These point clouds are then used to identify structural differences between the optimal and suboptimal feature sets. Once these differences are identified, we design an algorithm to identify optimal feature sets based on these observations.

This feature selection algorithm is based on mathematical properties of the feature sets, making it analytic in nature. This sets the algorithm apart from the current, most widely used, wrapper-based or filter-based feature selection methods, which are mostly heuristic in nature [2]. These methods sometimes require assumptions about the data set, for example, independence among the features, that might not be met by the data. Since our method is based on the geometric structure of the data set, it does not make such assumptions. Also, as machine learning techniques such as SVM become more widely adopted in various application domains, it is important to understand more about the interaction between a learner and a particular data set, as these insights may guide further development in the field. By discovering some mathematical properties that separate optimal feature sets from suboptimal feature sets, we can guide the feature selection process in a much more precise manner. Additionally, knowing these properties can help us to maximize the efficacy of SVMs for a particular classification problem. These properties could even be used to guide data collection efforts, in effect ensuring that the data collected is capable of providing a good feature space.

The algorithm is based on six properties that have been observed across several text data sets. The properties are based on dimensionality and intersection qualities of the affine hulls of the n-dimensional point clouds generated from a particular feature set. We evaluated the algorithm on several types of data sets, including low dimensional continuous data, low dimensional categorical data, high dimensional text data in a binary sparse vector format, and high dimensional text data in a word frequency-based sparse vector format. We identified the optimal feature sets of each data set using a wrapper based feature selection method which considers all possible subsets of the whole feature space. These optimal feature sets are then used to develop and evaluate the proposed feature selection algorithm, based on accuracy, precision and recall. We have observed that the algorithm delivers the best performance on the high dimensional text data, in both binary and word frequency-based formats. The algorithm is best suited to data whose features can be grouped together into feature types, for example, unigrams and bigrams.

Our algorithm achieves accuracies ranging from 76% to 86% within the data sets on which the model was trained, with an average precision of 86%, and an average recall of 72%. On test sets with dimensions ranging from 480 to 1440, accuracy ranges from 76% to 86%, with an average precision of 83% and an average recall of 81%. Precision remains high (.9-1) for data sets up to 3000 dimensions. However, the proposed algorithm does not perform well on test sets with dimension lower than approximately 500. More efforts are required to understand and address this phenomenon. While the CPU time used by the algorithm increases quadratically in both the number of features and the number of examples, the proposed algorithm requires no human interaction during its runtime.

We believe that this algorithm has a significant impact on the problem of feature selection. Its analytic nature sets it apart from current, more heuristic, methods used widely throughout industry. The process requires no supervision from the user, and thus provides a marked reduction in man hours needed to determine optimal feature sets.

2 Related Work

A great deal of studies have been carried out to identify the optimal features for a classification problem [3] [4]. However, such studies are mostly heuristic in nature. In this section we review the two studies that are most germane to our proposed feature selection algorithm.

Garg, *et al.* introduce the projection profile; a data driven method for computing generalization bounds for a learning problem [5]. This method is especially meaningful in high dimensional learning problems, such as natural language processing [5]. The method hinges on random projection of the data into a lower dimensional space. Garg, *et al.* assert that if the data can be projected into a lower dimensional space with relatively small distortions in the distances between points, then the increase in classification error due to these distortions will be small [5]. This is important, because in the lower dimension the generalization error bounds are smaller. Bradley, *et al.* [6] state that a lower data dimension also corresponds to a lower VC dimension, which in turn also causes lower generalization error [1]. Expanding on this idea, we apply these concepts to the feature selection problem by quantifying a particular feature sets capacity for dimensionality reduction, giving preference to those feature sets that have the potential to produce lower generalization error.

Bern, *et al.* emphasize the importance of the maximizing the margin between reduced convex hulls in the case of non linearly separable data. [7]. We investigate a relationship between classifier accuracy and properties of the intersection of the affine hulls of the class separated point clouds. In a sense, we are describing an upper bound on this margin, the idea being that the more intertwined the class separated point clouds are, the smaller the maximum margin between their reduced convex hulls becomes. We use the affine intersection of the class separated point clouds as a measure of a feature set's suitability for SVM classification with a linear kernel. The choice of affine hulls will be discussed further in the next section.

3 Identifying the Relevant Geometric Properties of a Data Set

The overall approach of this work is to examine feature sets arising from several natural language processing classification problems. We seek to identify key geometric properties that can be used to describe those feature sets for which an SVM performs well. This process of identifying relevant geometric properties is described in this section. In the next section, we construct an empirical algorithm for feature selection based on the geometric properties identified here.

Our training data consists of 717 feature sets, each manually labeled as optimal or suboptimal. These labels, based on classifier accuracy, were determined using an all subsets wrapper-based feature selection method on five data sets from four different classification problems. (These data sets are summarized in Table 3, Section 5.) For each classification problem, we train every possible binary SVM using every possible subset of features.

SVMs are inherently binary classifiers. There are several ways to address this when using SVMs for multi class problems. A commonly used approach, as described in [2], is the *one vs all* approach. We use the following variation on this method. Consider a multi class classification problem with ℓ classes; this problem consists of 2^ℓ classifiers that represent all possible ways to subdivide ℓ classes into two groups. We remove half of these possibilities due to symmetry. Finally, we do not consider the subset with an empty positive class to be a viable classifier, leaving $2^{\ell-1} - 1$ possible binary classifiers. Each example in our training data represents one possible feature set for one possible binary classifier for a particular classification problem.

Because, for most of the five data sets we used, the number of samples we have is smaller than the total number of available features, we chose to focus on a linear kernel SVM, since, given the small number of samples, more complex kernels will likely lead to overfitting. As a linear kernel SVM performs linear separation if possible, it would have been natural to study the convex hulls of the positive and negative classes of samples. However, due to considerations of performance and ease of implementation, we instead chose to focus on a much simpler geometric invariant: the affine hulls of the positive and negative classes of samples. This choice allows us to use standard and widely-available linear algebra libraries for our computations so that we can work with high dimensional data sets, like those associated with natural language applications, in a manner that is computationally feasible.

In this paragraph, we review some basic material on affine hulls. For more information, we refer the reader to standard geometry textbooks such as [8, 9]. Let $v_0, v_1, \ldots, v_k$ be vectors. For any set of vectors, we write $(v_0, v_1, \ldots, v_k)$ to denote the matrix with the v_i as columns. The *linear hull* $\operatorname{span}(v_0, v_1, \ldots, v_k)$ of the vectors v_i is the smallest linear space containing all v_i which, equivalently, can be defined as $\mathrm{span}(v_0, v_1, \ldots, v_k) = \{\sum \lambda_i v_i \mid \lambda_i \in \mathbb{R}\}$. The dimension of the linear hull is the rank of the matrix with the v_i as columns. An *affine space* is a translate of a linear space. In particular, it does not necessarily contain the origin. The *affine hull* $\mathrm{aff}(v_0, v_1, \ldots, v_k)$ of the v_i is the smallest affine space containing all v_i, which, equivalently, can be defined as

$$\mathrm{aff}(v_0, v_1, \ldots, v_k) = \left\{ \sum_{i=0}^{k} \lambda_i v_i \;\middle|\; \lambda_i \in \mathbb{R}, \sum_{i=0}^{k} \lambda_i = 1 \right\}.$$

The dimension of the affine hull is the dimension of the linear space that the affine hull is a translate of. In particular, the dimension of the affine hull of a point set is the same as the dimension of the polytope that is its convex hull. Thus, by definition, the dimension of the affine hull can be written as:

$$\dim(\mathrm{aff}(v_1, \ldots, v_d)) = \mathrm{rank}(v_1 - v_d, \ldots, v_{d-1} - v_d)$$

This simple observation makes calculations for higher dimensional data sets easy to implement and computationally efficient.

Now, suppose we have $n+1$ points in n-dimensional space. If the points are in general position, then the dimension of their affine hull is n. Moreover, assuming the points are in general position, then we can find a separating hyperplane for *any* partition of the points into two classes, i.e., the point set can be shattered. Somewhat surprisingly, it turns out however that the samples in our natural language processing data sets are not in general position. In fact, their affine hull has very low dimension, compared with the dimension of corresponding feature space. The ratio of the dimension of the affine hull and the dimension of the feature space in the data sets used to develop our training data are as low as .2, with an average of .52. Intuitively, if the ratio of the dimension of the affine hull over the dimension feature space is low, we expect the data set to contain a lot of structure, which the SVM can use to construct a classifier. (See also [5, 6], who show that if a data set can be effectively projected into a lower dimension with small distortions in the distances between points, the generalization error of that data set is lower than that of a data set lacking this property.) This observation has led us to consider several geometric measures, called f_1 through f_6, defined in terms of simple ratios. We use these measures to assess the differences in the geometric structure of optimal and suboptimal feature sets with respect to the given data.

Before we can define the properties f_i, we need to introduce some notation. The input data set to a binary classification problem is given in terms of a sparse matrix, with each point in the original data set represented as a row. The unique value of each feature is represented by a column in the matrix. That is, an entry a_{ij} in the matrix is 1 if the data point i contains feature j and it is 0 otherwise. The rows of this matrix are organized into blocks such that each block contains all the data points belonging to the same class. We refer to this matrix as the full matrix, or M_f. The submatrix consisting of only the rows in the positive class is referred to as the positive matrix M_p and the submatrix consisting of only the rows in the negative class is referred to as the negative matrix M_n. By considering every row as point in feature space, we can associate to each of these matrices a set of points in feature space. We refer to the resulting three point sets as the *full point cloud*, P_f, the *positive point cloud*, P_p and the *negative point cloud*, P_n. The *affine dimension* of a point cloud P is the dimension of its affine hull, $\mathrm{aff}(P)$. To assess the dimension of the ambient space, we could use the dimension of feature space, i.e., the total number of columns. However, it may happen that some of the columns in a given matrix are zero, and as such columns contain no additional information, we chose to exclude them from our count. Therefore the *ambient dimension* is defined as the number of non-zero columns in a given matrix. Geometrically, this is the dimension of the smallest coordinate subspace the corresponding point cloud is contained in. Given this terminology, we now define ratios f_1 through f_6 as given in Table 1. Ratios f_1 through f_5 each contain in affine dimension in their numerator and an ambient dimension in their denominator. Ratio f_6 divides the number of samples contained in both of the affine hulls of P_p and P_n by the total number of samples.[1]

Table 1. Definitions of Geometric Properties f_1-f_6

property	numerator	denominator
f_1	affine dimension of P_p	ambient dimension of P_p
f_2	affine dimension of P_n	ambient dimension of P_n
f_3	affine dimension of P_p	ambient dimension of P_f
f_4	affine dimension of P_n	ambient dimension of P_f
f_5	affine dimension of P_f	ambient dimension of P_f
f_6	# of samples in $\mathrm{aff}(P_p) \cap \mathrm{aff}(P_f)$	# of total samples

The purpose of the properties f_i is to allow us to assess the geometric structure of the data with respect to different feature sets. In this setting, a feature set is a set of columns. Selecting a certain subset of features amounts to removing the other columns from the matrices. Geometrically, this means projecting the point set onto the coordinate subspace corresponding to the selected feature set. We can then apply the measures f_i to these projected data sets and compare the values we obtain.

For each feature set in our training data, we trained a linear kernel SVM on the training data and assessed the performance of the linear classifier obtained on the test data. We also computed the values of the f_i for each feature set using the LinAlg library of NumPy [10].

The results of this experiment are shown in Figure 1. Each plot shows the standardized z-scores of the values of a particular geometric property for each of the 717 feature sets in our training data. The value of this ratio is plotted

[1] Note that if, in the definition of f_6, we used the term convex hull instead of affine hull, a value of $f_6 = 0$ would guarantee linear separability. However, with our definition of f_6 a value of $f_6 = 0$ is neither necessary nor sufficient for separability.

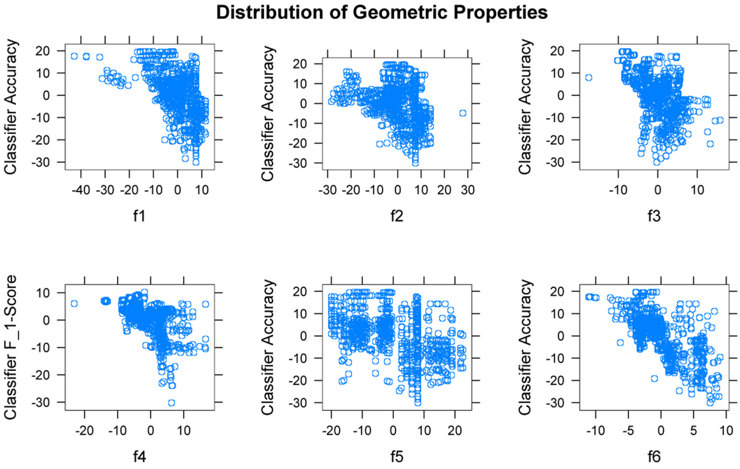


Fig. 1. Distributions of Geometric Properties

against a standardized measure of that particular feature set's performance. In most cases this measure is classifier accuracy, but in the case of f_4, we noticed a much stronger correlation between the f_4 value and the F_1-Score for a given feature set, which is defined as

$$F_1\text{-Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where precision and recall are given by

$$\text{precision} = \frac{tp}{tp + fp} \quad \text{and} \quad \text{recall} = \frac{tp}{tp + fn}$$

and tp, fp, fn represent the number of true positives, false positives and false negatives, respectively. (These values are calculated by comparing the predicted values against the labels that were manually assigned during the generation of our training set.) Notice the clear negative relationship between each of the properties and classifier performance. Each of the linear regression models pictured in Figure 1 are significant on an $\alpha = .01$ level.

Clearly, the geometric properties f_i contain information about the quality of a given feature set. In the next section we use the f_i as predictor variables to develop a logistic regression model, as well as a linear regression model that is the basis of our feature selection algorithm. We chose linear and logistic regression based on the observations in Figure 1, and the fact that we wish to determine whether a feature set is optimal or suboptimal, ultimately a binary decision.

4 Geometric Properties-Based Feature Selection Algorithm

The goal of this algorithm is to use the observations discussed in the previous section to identify optimal feature sets for a text classification problem using an SVM with a linear kernel. This section describes the specifics of the algorithm.

The input includes a training data set, a list of categories used to label the data, a set of boundary values for the feature types and a directory to store the output files. The columns representing a given feature type must be stored in consecutive columns, as previously described in Section 3. It is necessary for each training vector to contain at least one nonzero column for each feature type. If the data does not lend itself to this naturally, the user must handle missing values in the manner best suited to the particular learning problem. The vectors of the training data set should be represented in sparse vector notation.

```
1  for each unique binary classifier:

2    for each possible subset of features:
3           generate training vectors for subset
4           build positive, negative and full matrices

5           for each matrix:
6                   calculate ambient and affine dimension
7           calculate dimension based features:
                 (see Section 3 for details)
8           calculate affine intersection rate (f_6)
                 (see Section 3 for details)

9        standardize values for f1-f6 for all possible subsets

10   for each possible subset of features:
11         lin_pred = predict using linear regression model
12         log_pred = predict using logistic regression model

13         if lin_pred > 0 and log_pred >= .5:
14              prediction = optimal
15              write subset to file
16         else:
17              prediction = suboptimal
```

Fig. 2. Structural Feature Selection Pseudo Code

Figure 2 shows the structure of the algorithm. The program starts by identifying all the unique binary classifiers, and all the possible combinations of feature types(lines 1-2). It does this by generating all possible combinations of labels and eliminates those which are symmetric to an existing subset. It is necessary to remove the empty feature set, and the binary classifier with an empty positive or negative class. The program creates a directory chosen by the user and creates files within it to store the results for each of the unique binary classifiers. Then, the program executes a nested loop as shown in figure 2(lines 3-9). For each subset, we first need to process the training vectors so that they only include vectors for that particular feature set. Once this is done, the data points in the training set are split into positive and negative examples. Then, three matrices are used to represent the point clouds P_n,P_p and P_f(line 5). The ratios, described in Section 3, are calculated using the affine and ambient dimensions of these point clouds.

Table 2. Logistic and Linear Regression Models

Predictor	Logistic Coefficient	Linear Coefficient
β_0	-0.64063267	-1.039011e-12
f_1	0.15706603	.0
f_2	0.1327297	0
f_3	-0.03350878	09114375
f_4	-0.15182902	-.01223389
f_5	0.19548473	-.0200644
f_6	-0.68787718	0

Finally, the algorithm makes a predication for a particular feature set based on the linear and logistic regression models detailed in Table 2. These models were selected using forward stepwise inclusion with the AIC as the evaluation criterion. In order for a feature set to receive a prediction of *optimal*, the logistic regression model must predict a value greater than .5, and the linear model must predict a positive standardized accuracy. (Recall that a z-score of zero indicates the norm.) If both of these conditions are met, then the subset is written to the appropriate output file.

The output of the algorithm is a list of suggested feature sets that have the structural characteristics associated with optimal feature sets. Remember, an optimal subset need not be unique. The algorithm gives the user a list of subsets to chose from, based on the user's own criteria.

5 Algorithm Evaluation

In this section, we evaluate the power of the feature selection algorithm. We discuss some limitations of the algorithm, particularly, the relationship between the algorithm's performance and the dimensionality of the input data. We also present a theoretical and empirical time complexity analysis for the algorithm.

5.1 Algorithm Performance

The algorithm was run on each of the text data sets used to build the training set, and the results are presented in table 4. The polarity1, polarity2 and strength sentences are data sets originally used to classify the polarity and strength of relationships between a food/chemical/gene and a disease [12]. The movies documents [13] and webtext sentences [14] are built from corpora included in Python's Natural Language Tool Kit [14]. The movie review corpus is intended for classifying the polarity of reviews as positive or negative, and the webtext corpus consists of sentences from six different websites, each labeled according to their site of origin.

Table 3. Summary of Data Suite Used to Train Model

Data Set	R	C	BC	FT	resulting feature sets
Polarity1 Sentences	463	645	7	5	156
Polarity2 Sentences	324	600	7	5	183
Strength Sentences	787	645	7	5	179
Movies Documents	300	1000	1	3	7
Webtext Sentences	1200	500	15	4	192

Table 3 is a brief summary table of each set we used to train the model used in our feature selection algorithm. It includes the number of rows(R) and columns(C) of each raw data set. Each data set contains different types of features, and the number of these, (FT), is also listed for each data set. The number of unique binary classifiers (BC) resulting from the classification labels is also listed. Finally, the number of feature sets added to our training set as a result of the creation process is listed.

To evaluate our feature selection algorithm, we calculate its accuracy, precision and recall by comparing the predictions made by the algorithm to the labels that were generated during creation of the training set. (See Section 3 for the label generation process.) Using these labels, we define accuracy, precision and recall as follows:

$$\begin{aligned} \text{accuracy} &= \frac{tp + tn}{tp + fp + tn + fn} \\ \text{precision} &= \frac{tp}{tp + fp} \\ \text{recall} &= \frac{tp}{tp + fn}, \end{aligned}$$

where tp, tn, fp, fn represent the number of true positives, true negatives, false positives and false negatives, respectively. With respect to our algorithm, precision evaluates whether the feature sets selected by the algorithm actually perform optimally. Recall, on the other hand, measures how well the algorithm identifies all optimal feature sets. Recalling that an optimal feature set need not be

Table 4. Algorithm Performance for Training Data

Data Set	Accuracy	Precision	Recall
Polarity1 Sentences	0.7564	0.8621	0.625
Polarity2 Sentences	0.7542	0.6304	0.8529
Strength Sentences	0.7814	0.8986	0.6526
Movie Documents	0.8571	1	0.8
Webtext Sentences	0.8091	0.9067	0.6602

unique, we see that precision is extremely important to this task. It is of more value to the user that the percentage of recommended feature sets that actually produce optimal results is high, since these results are the pool from which the user will ultimately choose a feature set. Optimal feature sets that are excluded from the results, or false negatives, do not have nearly as much consequence.

Note, in table 4, the high precision within each data set. These numbers indicate that the algorithm we designed is quite effective for selecting optimal feature sets within the training data. Especially within the Movie Documents, where the algorithm achieves a precision of 1. This means that every feature set the algorithm returned was in fact an optimal feature set for classifying the Movies Documents with a linear SVM. While the algorithm's precision is somewhat lower on the Polarity2 Sentences, it is still impressive, given that only 38% of the feature sets within the Polarity2 Sentences are actually labeled as optimal.

In the aforementioned data sets the full feature set is close to optimal, which means that running a linear SVM directly on the data with all features included gives almost the same accuracy as first running our feature selection algorithm and then applying the linear SVM. To assess if our algorithm can effectively reduce the dimension when the full feature set is not optimal, we ran the following experiment. The Polarity1 data set was modified by adding 25% additional columns, increasing the total number of columns to 806. Each additional column was a random binary vector and received a random label. We applied our algorithm to each of the resulting binary classification problems. In all cases our algorithm recognized that the random columns did not contain relevant information and excluded them from the feature set. Applying the linear SVM to the reduced feature set, as selected by our wrapper algorithm, leads to a substantial improvement over applying the linear SVM directly to the full feature set: Accuracy increased by between 10% and 26% with a median increase of 15%.

To test our algorithm on larger data sets, we created several data sets from the Amazon Customer Review Data, available from the UCI Machine Learning Repository [15]. The raw data consists of 10,000 features and 1500 examples, with labels corresponding to 50 different authors. We developed each test set using a different set of five authors. Using different authors ensures that the reviews will be entirely different from one data set to the next. Because the reviews are different, the particular set of features generated will also be different, even though they are created in the same manner. The dimension of the resulting data sets can increased or decreased by controlling the frequency requirements for inclusion of a feature. For example, to reduce the numbers of features, we would require that a particular unigram feature be present within the reviews at least 10 times. Then, to increase the dimension, we simply include less and less frequent features. Each test set also went through the same labeling process as the training data, in order to determine the algorithm's accuracy, precision and recall on previously unseen data. Recall this process was based on a wrapper based, all subsets algorithm that is commonly used to address the problem of feature selection. The results indicate that the algorithm also performs very well on previously unseen data. The Amazon data set was used to test the algorithm over a range of dimensions, and table 5 summarizes the performance for these tests for column dimensions ranging from 480 to 1440. These results indicate that the algorithm performs very well within this range of column dimensions. We have observed that precision remains high (.9-1) for dimensions up to 3000.

Table 5. Peak Algorithm Performance for Amazon Data

Dimension	Accuracy	Precision	Recall
480	0.768888889	0.823529412	0.711864407
640	0.76	0.838095238	0.704
800	0.831111111	0.844444444	0.870229008
960	0.817777778	0.837037037	0.856060606
1120	0.813333333	0.822222222	0.860465116
1280	0.795555556	0.8	0.850393701
1440	0.804444444	0.82962963	0.842105263

5.2 Limitations

As explained in Section 3, the proposed algorithm is designed to work well for linear kernel SVMs. In situations where the ratio of the number of samples to the total number of features is very large and the use of a higher degree kernel is warranted, we do not expect the affine geometry of the data set to reveal much useful information about which feature sets allow the SVM to generalize well.

Moreover, the proposed algorithm is tailored towards binary data and we do not expect it to perform well on continuous data: Suppose the data consists of n points in n-dimensional space that are drawn from a model that generates points on a 1-dimensional affine subspace with a small additive error that is normally distributed. In this scenario the n data points will span an affine space of dimension n, even though the true model is 1-dimensional. These theoretical considerations are confirmed by experiments which show that the algorithm does not perform well for continuous and categorical data. Table 6 provides a summary of the algorithm's performance on several test data sets according to column dimension and data type. A precision or recall score of 0 indicates that the algorithm did not accurately identify any optimal feature sets.

Table 6. Predictive Results for Low Dimensional Data

Column Dimension	Data Type	Accuracy	Precision	Recall
13	continuous	0.3218	0.1111	0.2083
38	categorical	0.4444	0	0
100	categorical	0.4286	0	0

Moreover, the data presented in Table 6 suggest that low dimensional data sets may limit the performance of the proposed algorithm. To better understand the relationship between our algorithm's performance and dimensionality, we designed an experiment using an Amazon data set as described above. The columns within each of the four feature types are organized in terms of frequency, so that the most common features occur in the earlier columns of each feature type block. The algorithm is used on these data sets repeatedly, while incrementing the number of dimensions included each time. For instance, the first run of the algorithm may include a total of 80 dimensions, the first 20 columns from each feature type. The algorithm's accuracy, precision and recall are recorded for the particular dimension, as well as the CPU time. The total number of features included is then increased to 160, by including the first 40 columns of each feature type. This process is repeated until all available dimensions are being used in the feature selection process. This is different than the previous Amazon data sets, because we are using the same set of five authors throughout the entire experiment, to control for variance between raw data sets. Figure 3 shows the results of this experiment. This experiment was repeated several times each using a different set of five authors with similar results.

These experiments indicate that the performance of the algorithm is very dependent on the dimensionality of the input data. Note the low values in accuracy, precision and recall for those data sets with less than 400-500 columns. Figure 3 shows the rapid growth in accuracy, precision and recall for the lower dimensions that becomes much slower for dimensions larger than 500. Further study may be warranted to discover the cause of the dimensionality dependence observed in these experiments.

In figure 3, we see that the CPU time increases quadratically with column dimension. Note though, that the number of rows, feature types and labels are all held constant through out the experiment. The theoretical time complexity of the algorithm is in fact a function of all of these variables;

$$\mathcal{O}\left((2^{(\ell-1)}-1)(2^k-1)(m^2n^2)\right),$$

where ℓ is the number of classification labels in the problem, k is the number of feature groups present, and m, n are the number of rows and columns, respectively, in the training data. The $\mathcal{O}(m^2n^2)$ terms come from the complexity of the singular value decomposition algorithm which is $\mathcal{O}(mn^2)$ [16]. In our algorithm, we perform this calculation $(m+2)$ times during the calculation of the affine intersection ratio. Recall, that the affine intersection ratio is calculated for

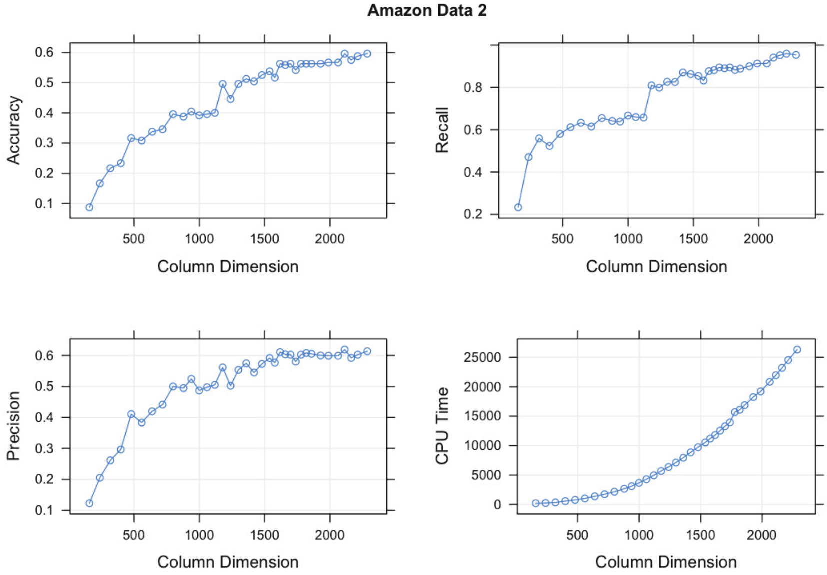


Fig. 3. Dimension Testing Results

$(2^k - 1)$ feature sets, for each of $(2^{\ell-1} - 1)$ unique binary classifiers. While the Amazon data sets had the capacity to test up to 10,000 columns, the run time became unreasonably long after around 2400 dimensions on a lap top computer.

6 Conclusion

Support Vector Machines are machine learning classifiers that use geometry to separate training examples into their respective classes. Because of this, SVMs are an excellent candidate for a structural based feature selection algorithm. Many of the commonly used feature selection algorithms are heuristic in nature and do not use inherent characteristics of the data. A more data driven, analytic approach to feature selection will help machine learning engineers to better understand the relationship between a particular classification problem and a given optimal feature set. This understanding can influence data collection efforts and improve classification accuracy.

Through investigating the geometric structure of optimal and suboptimal feature sets, we found six qualities that differ significantly between them. We have discovered a linear relationship between the values of our dimensionality and intersection based features with classifier performance. We built linear and logistic regression models that use these six properties as predictor variables to identify optimal feature sets. We used these models to design a filter based feature selection algorithm that is analytic in nature, as opposed to the more commonly used wrapper based heuristic methods.

Our feature selection algorithm performs best on text data sets that have more than approximately 500 features that can be organized into a handful feature types. While the precision remains high for data sets with more that 2500 features, the computation time needed for these sets is too long to be practical on a single computer. Because of this, further study into parallelization of the algorithm may be warranted.

The algorithm did not perform well on low dimensional data sets. More study is needed to determine the cause of the relationship between the dimensionality of the original input data set. Currently, the algorithm does not support feature selection for SVMs using non linear kernels. However, we hypothesize that the algorithm could be successful when applied to other kernel types, if the data is first transformed using the chosen kernel, and the f_i's are then calculated in the transform space. Further study is needed to accept or reject this hypothesis. Despite these limitations, our algorithm presents a useful and innovative approach to the problem of feature selection.

References

1. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
2. Han, J., Kamber, M.: Data mining: concepts and techniques, 2nd edn. Morgan Kaufmann (2006)
3. Molina, L.C., Belanche, L., Nebot, A.: Feature selection algorithms: a survey and experimental evaluation. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 306–313. IEEE Comput. Soc. (2002)
4. Joachims, T.: Making large-scale support vector machine learning practical (1998)
5. Garg, A., Har-peled, S., Roth, D.: On generalization bounds, projection profile, and margin distribution (2002)
6. Bradley, P., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Machine Learning Proceedings of the Fifteenth International Conference, ICML 1998, pp. 82–90. Morgan Kaufmann (1998)
7. Bern, M., Eppstein, D.: Optimization over zonotopes and training support vector machines (2001)
8. Webster, R.: Convexity. Oxford University Press, Oxford (1994)
9. Ziegler, G.M.: Lectures on Polytopes. Springer (1995)
10. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001)
11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes: The Art of Scientific Computing, 3rd edn. Cambridge University Press, New York (2007)
12. Swaminathan, R., Sharma, A., Yang, H.: Opinion mining for biomedical text data: Feature space design and feature selection. In: The Nineth International Workshop on Data Mining in Bioinformatics, BIOKDD 2010 (July 2010)
13. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the ACL (2004)
14. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media (2009)
15. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
16. Tesic, J.: Evaluating a class of dimensionality reduction algorithms abstract

Evaluation of Hyperspectral Image Classification Using Random Forest and Fukunaga-Koontz Transform

Semih Dinç[1] and Ramazan S. Aygün[2]

Computer Science Department
University of Alabama in Huntsville
Huntsville, AL 35899 USA
{sd0016,aygunr}@uah.edu
www.cs.uah.edu

Abstract. Since hyperspectral imagery (HSI) (or remotely sensed data) provides more information (or additional bands) than traditional gray level and color images, it can be used to improve the performance of image classification applications. A hyperspectral image presents spectral features (also called spectral signature) of regions in the image as well as spatial features. Feature reduction, selection, and transformation has been a challenging problem for hyperspectral image classification due to the high number of dimensions. In this paper, we firstly use Random Forest (RF) algorithm to select significant features and then apply Kernel Fukunaga Koontz Transform (K-FKT), a non-linear statistical technique, for the classification. We provide our experimental results on AVIRIS hyperspectral image dataset that contains various types of field crops. In our experimental results, we have obtained overall classification accuracy around 84 percent for the classification of 16 types of field crops.

1 Introduction

An ordinary color image has three bands (e.g., red, green, and blue) of the visible light just as human eye can see. So most imaging systems are restricted only a few spectral bands. These few bands or dimensions are usually not enough for the classification of a single pixel. However, hyperspectral imaging sensors divide the visible light spectrum into hundreds of bands for a single pixel as presented in Figure 1.

Spectral features (or bands) are sensitive to the type of material in addition to the color or shapes of objects. Because of this ability, hyperspectral imagery has different applications in many areas such as agriculture, mineralogy, physics, and homeland security. Although it was inconvenient technology in the past, recent advances have simplified the capture and process of hyperspectral images.

On the other hand, hyperspectral imagery (HSI) has some challenges to be overcome. Firstly, some of the spectral bands may have large amount of noise due to water absorption bands and some other environmental effects, since spectral

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 234–245, 2013.

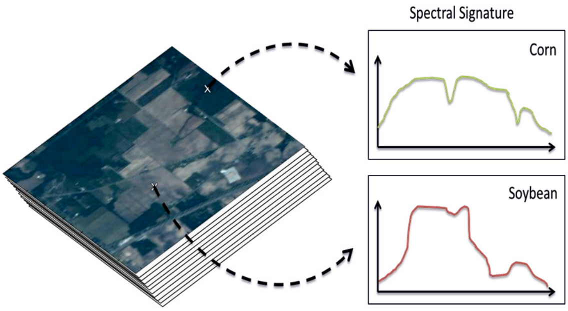


Fig. 1. Example of Hyperspectral Imagery

bands are more sensitive than typical vision sensors. Secondly, image processing and classification takes longer than the analysis of traditional gray scale and color images, since there are many number of features to be analyzed and processed. Therefore, some spectral bands might be noisy and redundant for image content analysis. It is necessary to detect the redundant bands and eliminate them to improve the processing speed and classification accuracy [1].

The research on HSI classification can be categorized based on feature selection and the classification method. The feature selection is sometimes handled manually by using prior information about the spectral bands or using previous experimental results. Various classification techniques including neural networks, support vector machines, bayesian classifier and decision trees have been used. In 2005, Benediktsson et al. [2] proposed a classification method using extended morphological models and neural networks. Banarjee et al. [3] studied on anomaly detection in HSI and used support vector machines for the classification in 2006. In 2007, Borges et al. [4] proposed discriminative class learning using a new Bayesian based HSI segmentation method. In 2008, Alam et al. [5] proposed a Gaussian filter and post processing method for HSI target detection. Du et al. [6] studied on HSI classification based on decision level fusion in 2010. In 2011, Tuia et al. [7] worked on the same topic using multi-scale cluster kernels. Samiappan et al. [8] proposed an SVM based HSI classification study which uses the same dataset used in this paper. Automated feature selection and acquiring high accuracy are still major problems for HSI.

In this study, we evaluate the combination of Random Forest (RF) algorithm and Kernel Fukunaga-Koontz transform (K-FKT). Firstly we performed automated feature elimination using RF algorithm, and then K-FKT is used to classify HSI data. We present our experimental results on the AVIRIS dataset [9] that contains images of 145x145 pixels with 220 spectral bands. Each spectral interval is 10 nm from the range 400 to 2450 nm wavelength. AVIRIS Image covers 2 x 2 mile portion of Northwest Tippecanoe County, Indiana.

This paper is organized as follows. The following section explains feature selection or ranking using Random Forest algorithm. Section 3 presents Kernel-Fukunaga-Koontz Transform in detail including its training and testing stages.

Classification results of different feature sets are presented in Section 4. The last section concludes our paper.

2 Feature Selection Using Random Forest

Random Forest (RF) algorithm is applied to select informative features in spectral signatures. RF is a type of an ensemble classifier that employs many different (independent) decision trees. Basically in this method every single decision tree makes a prediction for a data item. The predictions of each decision tree are evaluated to determine the class of the data item. If the majority voting is used, the most voted class is chosen as the class of the data item. This approach as a random forest was first proposed by Leo Breiman and Adele Cutler in 2001 [10]. RF algorithm provides remarkably high accuracy in various studies [11] especially on large data sets. Cumbaa et al. [12] present the performance of RF algorithm on protein crystallization analysis using very a large database.

Random forests (RF) are comprised of decision trees and starts with selecting many bootstrap samples. A bootstrap data set has almost 63 percent of the original observations which are chosen randomly from the original dataset. The other samples which are not in the bootstrap dataset are called as out-of-bag observations. The best split for each tree is selected by using chosen attributes. This process repeats for each branch until our bootstrap grows into a proper well-formed tree. When the leaf nodes have small number of samples to split or no splitting criterion can be found, the decision tree induction ends. A decision tree is constructed for each bootstrap sample. Then each decision tree is employed to classify the out-of-bag observations. The predicted class of an observation is calculated by majority of votes of all decision trees for that observation (see Figure 2).

Generally a statistical classification or a regression method measures feature importance by choosing variables using statistical importance. However, RF approach runs in a completely different way. For each individual decision tree in the forest, there is a misclassification rate for the out-of-bag samples. In order to decide the importance of a specific predictor variable or a feature, the values of the features are randomly ordered for the out-of-bag observations. The algorithm performs prediction and checks for the change of the mean squared error (MSE) of out-of-bag data in which the corresponding variable is reordered and all others are fixed. In this way, a variable can be scored based on the prediction results.

3 Classification using Kernel Fukunaga-Koontz Transform

Kernel Fukunaga-Koontz Transform is applied in order to classify field crops in hyperspectral image for each selected feature set. Classical FKT is a well-known approach [13] [14] [15] [16] for separating two classes, and it operates by

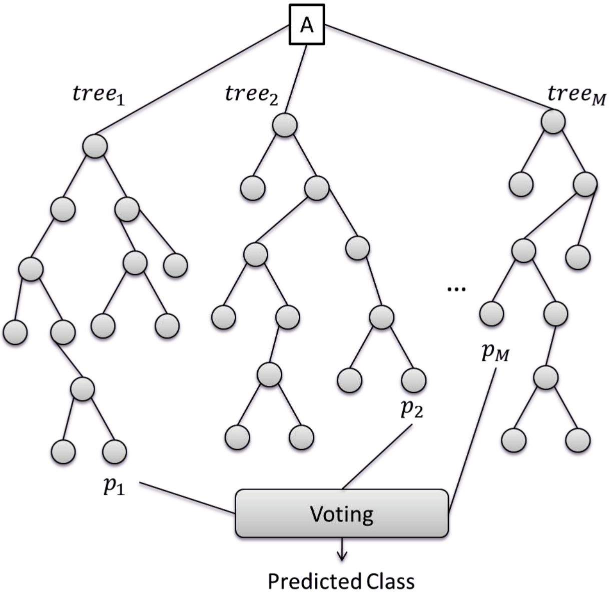


Fig. 2. Voting in Random Forest

transforming data into a new space where both classes share the same eigenvalues and eigenvectors. However, when the data is non-linearly distributed, the classical FKT method is not able to give satisfactory results for the classification. Therefore, Kernel transformation is combined with classical the FKT to classify non-linearly distributed data as if it was linearly distributed. K-FKT algorithm as a supervised classification approach consists of two stages: training and testing.

3.1 Training Stage

K-FKT is a binary classification approach. So we employ one-versus-all method to deal with the multi-class classification problem. The target class is the one which we want to classify, and the background class is the combination of all the other classes. Equations (1) and (2) represent these separated datasets for a sample target class as follows:

$$X = [x_1, x_2, ...x_N] \tag{1}$$

$$Y = [y_1, y_2, ...y_N] \tag{2}$$

where X and Y contain the target training data and the background training data, respectively; and x_i and y_i represent the samples (observations) for the target and background classes (or training signatures), respectively.

When data have a non-linear distribution, we are not able to separate classes with a linear classifier. But we can achieve this limitation with transforming the data into a higher dimensional space. Assume that a virtual mapping function maps the data into higher dimensional space as shown in (3) and (4):

$$\tilde{X} = [\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_N] \tag{3}$$

$$\tilde{Y} = [\tilde{y}_1, \tilde{y}_2, ...\tilde{y}_N] \tag{4}$$

where the symbol '$\sim$' indicates that corresponding sample has been transformed to the higher dimensional kernel space. In that case transformed data may be linearly separable only if the proper mapping function is chosen. However, in most cases, the virtual mapping function does not exist (or we do not need to use) in applications. Instead, a kernel function 5 is employed by following a 'kernel trick' approach [15]. By using kernel trick, we are able to measure the distance between samples directly in higher dimensional kernel space without mapping data into that space.

$$K(x_i, x_j) = exp(\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{5}$$

In this study Gaussian type kernel function is selected. This function measures the distances between two samples by using a calibration parameter σ $(0 < \sigma < 1)$. In this way the classification process can be achieved in a linear fashion.

According to the FKT, covariance matrices of $\tilde{X}$ and $\tilde{Y}$ must be computed. As shown in (6) and (7), covariance matrices are named as T_0 and C_0 for the target and the background datasets, respectively:

$$T_0 = \tilde{X}\tilde{X}^T \tag{6}$$

$$C_0 = \tilde{Y}\tilde{Y}^T \tag{7}$$

The next step of the FKT algorithm is to sum T_0 and C_0 and to decompose this sum matrix into eigenvalues and eigenvectors. In (8) V represents the eigenvector matrix and D represents the eigenvalue matrix. Diagonal elements of D correspond to eigenvalues of the summation matrix.

$$T_0 + C_0 = VDV^T \tag{8}$$

In this way we are able to construct transformation operator P by using V and D matrices. This new operator is employed to transform data into the eigenspace. Note that this transformation is different from the kernel transformation in (5). Equation (9) shows how to derive the transformation operator:

$$P = VD^{-\frac{1}{2}} \tag{9}$$

Then T_0 and C_0 matrices are transformed into a lower dimensional eigenspace using the transformation operator P. The new T and C matrices are obtained as shown in (10) and (11),

$$T = PT_0P^T \tag{10}$$

$$C = PC_0P^T \tag{11}$$

where T and C represent the transformed target and background matrices, respectively. The sum of the transformed matrices should be equal to the identity matrix as shown in (12) since they are transformed by the same operator and share the same eigenvectors and eigenvalues:

$$I = T + C \tag{12}$$

Equation (12) states that while T contains more valuable information for the target class, C contains more important information for the background class. After obtaining these two matrices, the training stage is completed. These matrices are in the testing stage.

3.2 Testing Stage

In the testing stage, the test vector z must be transformed into the kernel space as performed in the training stage. Due to similar requirements as in the training stage, the 'kernel trick' method must be applied again for the kernel transformation:

$$Z = [K(x_1, z), K(x_2, z), ..., K(x_N, z)] \tag{13}$$

In (13), z is a test sample (spectral signature that we want to classify) on the hyperspectral image, x_i is the i^{th} target training sample in (1), $K(x, y)$ is a kernel function, and Z is the kernel matrix of the corresponding test sample. This matrix is necessary to have the transformed feature vector for the test sample as shown in (14):

$$F_j = \frac{1}{\sqrt{\lambda_j}}\phi_j^T Z \qquad j = 1, 2, ...N \tag{14}$$

where λ and ϕ represent the eigenvalues and eigenvectors of the normalized target matrix $\hat{T}$ in (15):

$$\hat{T} = T_0 - I_{1/N}T_0 - T_0I_{1/N} + I_{1/N}T_0I_{1/N} \tag{15}$$

where $I_{1/N}$ is equal to the division of an identity matrix I_{NxN} by N (I_{NxN}/N).

Then we decompose the eigenvalues and eigenvectors of the normalized target matrix $\hat{T}$ and multiply the feature vector F with the transpose of the eigenvector matrix of $\hat{T}$ as shown in (16):

$$R = \Phi^T F \tag{16}$$

where Φ represents the eigenvectors of $\hat{T}$. The decision result R is obtained by (16). This operation helps us decide the class that a test sample belongs. If the value of R is greater than a threshold value, it belongs to the target class; otherwise, it belongs to the background class.

4 Experimental Results

Dataset and Preprocessing. In this section, we explain the set of experiments which we performed using the AVIRIS Hyperspectral Image datasets called 'Indian Pines' [17]. The Indian Pines scene contains several type of areas. Among these areas, there are agricultural fields, forests, highways, a rail line, and

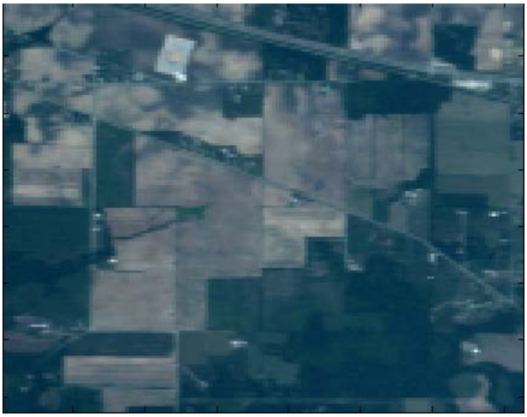


Fig. 3. RGB view of AVIRIS Image

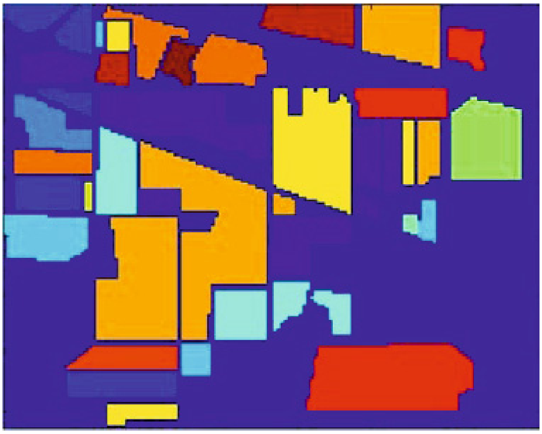


Fig. 4. Ground Truth data of AVIRIS Image

Table 1. Class Names and Number of Samples

Class Number	Class	Samples
1	Alfalfa	54
2	Corn-notill	1434
3	Corn-mintill	834
4	Corn	234
5	Grass-pasture	497
6	Grass-trees	747
7	Grass-pasture-mowed	26
8	Hay-windrowed	489
9	Oats	20
10	Soybean-notill	968
11	Soybean-mintill	2468
12	Soybean-clean	614
13	Wheat	212
14	Woods	1294
15	Buildings-Grass-Trees-Drives	380
16	Stone-Steel-Towers	95
Total		10366

some low density housing. The colored view of the scene is shown in Figure 3. In this dataset, there are 16 different classes which are represented as a ground truth data in Figure 4. The names of the classes and the number of samples are shown in Table 1.

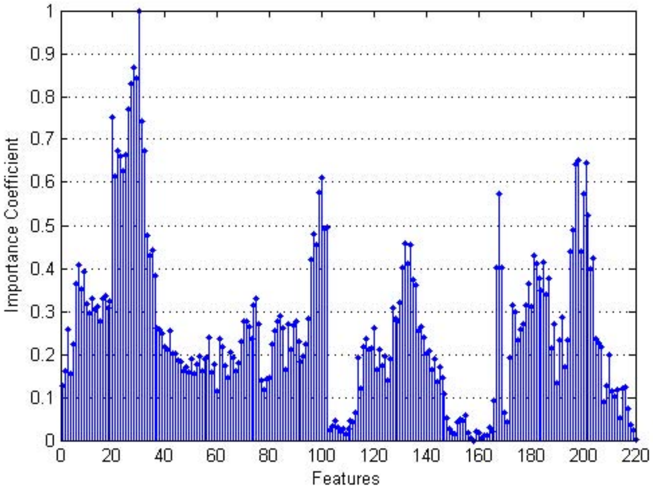


Fig. 5. Importance Coefficients of Features

The AVIRIS data (or image) comprises of a 145 × 145 × 220 matrix that corresponds to 220 different bands of images having size of 145 × 145. We transform the matrix data to a vector form as a 21025 × 220 matrix. This representation indicates that there are 21025 samples with 220 different features.

Our goal is to classify the field crops. So firstly, we removed regions that do not correspond to field crops (dark blue areas in Figure 4) from the dataset. Among 21025 different samples, nearly half of them do not have meaning since they do not correspond to field crops. These samples are labeled with class 0 (zero). After this operation, the number of remaining samples is 10336 as presented in Table 1.

Table 2. Precision and Recall Classification Results of 16 Classes with respect to number of features

#of features		219	190	182	163	129	98	70	53	43	29	20	12	7	#of samp.
Class 1	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.93	0.90	0.90	0.90	0.87	26
	recall	0.77	0.77	0.77	0.77	0.92	0.96	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 2	prec	0.76	0.76	0.76	0.76	0.71	**0.65**	0.61	0.60	0.60	0.60	0.60	0.47	0.46	716
	recall	0.89	0.89	0.89	0.90	0.92	0.95	0.97	0.98	0.98	0.99	**1.00**	0.85	0.81	
Class 3	prec	0.88	0.88	0.88	0.88	0.86	**0.82**	0.78	0.77	0.76	0.70	0.64	0.60	0.45	416
	recall	0.73	0.73	0.73	0.73	0.77	0.81	0.84	0.85	0.85	0.85	**0.85**	0.80	0.76	
Class 4	prec	0.95	0.95	0.95	0.95	0.89	**0.70**	0.65	0.64	0.64	0.64	0.64	0.64	0.49	116
	recall	0.93	0.93	0.93	0.93	0.95	0.97	1.00	1.00	1.00	1.00	1.00	**1.00**	0.86	
Class 5	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.08	249
	recall	0.95	0.95	0.95	0.95	0.96	0.98	1.00	1.00	1.00	1.00	1.00	**1.00**	0.08	
Class 6	prec	1.00	1.00	1.00	1.00	1.00	**1.00**	0.97	0.92	0.89	0.43	0.41	0.41	0.36	373
	recall	0.87	0.87	0.87	0.87	0.88	0.90	0.93	0.95	**0.95**	0.76	0.69	0.69	0.57	
Class 7	prec	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.80	0.80	0.80	0.80	0.80	0.71	12
	recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 8	prec	1.00	1.00	1.00	1.00	1.00	**1.00**	0.98	0.97	0.95	0.92	0.89	0.89	0.85	245
	recall	0.95	0.95	0.95	0.95	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**1.00**	
Class 9	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.92	0.92	0.92	10
	recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 10	prec	0.76	0.76	0.76	0.76	0.71	**0.68**	0.61	0.61	0.46	0.46	0.46	0.46	0.46	484
	recall	0.87	0.87	0.87	0.87	0.89	0.93	0.99	**0.99**	0.85	0.84	0.83	0.83	0.80	
Class 11	prec	0.79	0.79	0.79	0.79	0.75	**0.69**	0.66	0.65	0.64	0.45	0.45	0.45	0.45	1234
	recall	0.76	0.76	0.76	0.77	0.81	0.86	0.92	0.93	**0.94**	0.80	0.79	0.79	0.75	
Class 12	prec	0.85	0.85	0.85	0.85	0.84	**0.80**	0.70	0.66	0.63	0.60	0.46	0.46	0.46	306
	recall	0.91	0.91	0.91	0.91	0.92	0.92	0.97	0.98	0.99	**1.00**	0.86	0.86	0.83	
Class 13	prec	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.96	0.86	0.78	0.73	0.68	0.21	106
	recall	0.94	0.94	0.94	0.94	0.95	0.97	0.98	0.99	0.99	1.00	1.00	**1.00**	0.26	
Class 14	prec	0.81	0.81	0.81	0.81	0.79	**0.79**	0.80	0.80	0.80	0.80	0.79	0.79	0.79	646
	recall	0.91	0.91	0.91	0.91	0.93	0.96	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 15	prec	0.91	0.91	0.91	0.91	0.88	**0.84**	0.73	0.73	0.73	0.73	0.73	0.73	0.44	190
	recall	0.86	0.86	0.86	0.87	0.87	0.87	0.89	0.98	0.93	0.94	0.95	**0.95**	0.79	
Class 16	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.87	0.74	0.47	0.41	47
	recall	0.72	0.72	0.72	0.72	0.72	0.72	0.79	0.83	0.85	0.89	**0.94**	0.89	0.70	
Avg.	prec	0.91	0.91	0.91	0.91	0.90	**0.87**	0.84	0.82	0.79	0.72	0.69	0.66	0.52	5176
	recall	0.87	0.88	0.88	0.88	0.90	0.93	0.95	**0.96**	0.95	0.94	0.93	0.92	0.77	

We performed a randomization to the permutation of the data in order to have a reliable classification result. After that we split data into two parts and choose 5190 samples for the training set and 5176 samples for the testing set.

Feature Selection Experiments. For the feature selection part, the number of of decision trees that are trained is set to 500. Only 14 features are available as the candidates at each split to increase the independence among decision trees. The square root of the total number of features is usually recommended as the number of candidate features at a node of a decision tree. According to these parameters, Figure 5 represents the importance coefficients, produced by RF, for each feature in the AVIRIS data.

Classification Performance.We rank the features and select the best N features using importance coefficients in Figure 5 for each experiment. Table 2 lists the precision and recall for each feature set. The results show that reducing the number of features increases the accuracy for most classes. This consequence verifies the presence of redundant features in HSI to be eliminated.

For overall evaluation of experimental results, Table 3 is generated with a single accuracy value for each case. This accuracy is computed as the ratio of the number of 'True Positive' and 'True Negative' samples to the number of all samples.

At the bottom of the table, the weighted accuracy is presented in which the ratio of the number of test samples in a class to the total number of test samples is considered as the weight of a class. Table 3 points out that to select best 98 features offers the best performance for classes.

Table 3. Overall Accuracy Results

#of features	219	190	182	163	129	98	70	53	43	29	20	12	7	#of samp.
Class 1	0.93	0.93	0.93	0.93	0.94	0.94	0.98	0.98	0.98	**1.00**	0.98	0.98	0.98	26
Class 2	0.72	0.72	0.72	0.72	0.74	0.75	0.75	0.75	**0.75**	0.74	0.74	0.74	0.74	716
Class 3	0.80	0.80	0.80	0.80	0.80	**0.80**	0.79	0.77	0.76	0.76	0.74	0.73	0.72	416
Class 4	0.89	0.89	0.89	0.89	0.92	**0.90**	0.87	0.80	0.79	0.71	0.68	0.68	0.51	116
Class 5	0.84	0.84	0.84	0.85	0.86	0.88	0.90	0.92	0.92	**0.92**	0.90	0.87	0.86	249
Class 6	0.91	0.91	0.91	0.91	0.91	0.91	0.94	**0.94**	0.93	0.90	0.84	0.70	0.63	373
Class 7	0.96	0.96	0.96	0.96	0.96	1.00	1.00	1.00	1.00	1.00	**1.00**	0.96	0.96	12
Class 8	0.93	0.93	0.93	0.93	0.94	0.96	0.97	0.98	**0.98**	0.97	0.96	0.95	0.95	245
Class 9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.97	0.94	0.94	10
Class 10	0.83	0.84	0.84	0.84	0.85	**0.87**	0.81	0.77	0.73	0.69	0.68	0.67	0.66	484
Class 11	0.83	0.83	0.83	0.84	0.83	**0.82**	0.81	0.80	0.80	0.79	0.77	0.77	0.77	1234
Class 12	0.78	0.78	0.78	0.78	0.78	**0.77**	0.74	0.73	0.73	0.72	0.68	0.65	0.64	306
Class 13	0.94	0.94	0.94	0.95	0.95	0.97	0.98	**0.98**	0.97	0.96	0.97	0.95	0.94	106
Class 14	0.84	0.84	0.84	0.84	0.84	0.84	0.87	0.88	0.88	0.88	**0.88**	0.87	0.87	646
Class 15	0.84	0.84	0.84	0.84	0.86	**0.88**	0.84	0.83	0.82	0.82	0.81	0.79	0.79	190
Class 16	0.81	0.81	0.81	0.81	0.82	0.84	0.84	0.87	0.89	**0.91**	0.89	0.87	0.86	47
Weig. Acc.	0.83	0.83	0.83	0.83	0.84	**0.84**	0.83	0.82	0.82	0.81	0.79	0.77	0.76	5176

However, the performance of some classes such as "Class 2" and "Class 12" does not exceed 80% accuracy. In order to clarify this lower performance spectral signatures are examined in more detail and we realized that these two classes does not have a specific distinguishing behaviour like other classes. In other words, there are samples which are very similar to the other class signatures. That is why our classification approach is not able to predict these problematic samples. We have focused on using only spectral features in this study but employing also spatial features (e.g, neighbourhood information) of HSI may improve the results.

Comparison. Samiappan et al. [8] classifies the same dataset using support vector machines with non-uniform feature selection. They divide the spectral bands into regions in order to obtain the best feature set combination. Finally they employ radial-based SVM (Support Vector Machine) classifier to classify regions. Their results presented 75% of overall accuracy by using the 100 of 220 features on the AVIRIS dataset. In this study we select the best 98 features of 220. Our results point out a remarkable contribution and exceeds 75% accuracy by reaching 84% overall accuracy.

In a different study [18], same 'Indian Pine' dataset is used for the same classification problem. According to that study, their overall classification accuracy is 87% and slightly higher than our results. Although there is a 3% difference, the way of selecting training and testing samples may be different because the authors do not mention how they divided their data into training and testing. They also used higher number of training samples and lower number of test samples than our study.

Another important issue is that optimal number of features may be different for classes. So we do not have to use all 98 features for all classes. For example, choosing the best 29 features for classes 1, 5, 9 and 16 produce the highest accuracy. Similarly classes 7 and 14 give the best results using best 20 features. Therefore, if our goal is to classify a specific class we can consider the optimal feature set for that particular class. Otherwise, we can use 98 features (bold values in Table 3) for all classes with 84% overall accuracy.

5 Conclusion

In this study we evaluated a combination of two powerful methods for the HSI classification problem. We have used Random Forest algorithm to select the important features. Then we used Kernel Fukunaga-Koontz Transform to apply binary classification. Experimental results show that reducing the number of features using RF algorithm increases the performance up to some limit for majority of classes. Moreover we have obtained promising weighted classification accuracy around 84%.

References

1. Binol, H., Bal, A., Dinc, S.: Classification on hyperspectral images using enhanced covariance descriptor. In: 20th Signal Processing and Communications Applications Conference (SIU), pp. 1–4 (2012)
2. Benediktsson, J.A., Palmason, J.A., Member, S., Sveinsson, J.R., Member, S.: Sveinsson, classification of hyperspectral data from urban areas based on extended morphological profiles. IEEE Transactions on Geoscience and Remote Sensing 43, 480–491 (2005)
3. Banerjee, A., Burlina, P., Diehl, C.: A support vector method for anomaly detection in hyperspectral imagery. IEEE T. Geoscience and Remote Sensing, 2282–2291 (2006)
4. Borges, J.S., Bioucas-Dias, J.M., Marçal, A.R.S.: Bayesian hyperspectral image segmentation with discriminative class learning. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4477, pp. 22–29. Springer, Heidelberg (2007)
5. Alam, M.S., Islam, M.N., Bal, A., Karim, M.A.: Hyperspectral target detection using gaussian filter and post-processing. Optics and Lasers in Engineering 46(11), 817–822 (2008)
6. Du, P., Zhang, W., Xia, J.: Hyperspectral remote sensing image classification based on decision level fusion. Chin. Opt. Lett. 9(3), 031002 (2011)
7. Tuia, D., Camps-Valls, G.: Urban Image Classification With Semisupervised Multiscale Cluster Kernels. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 1 (2011)
8. Samiappan, S., Prasad, S., Bruce, L.M.: Automated hyperspectral imagery analysis via support vector machines based multi-classifier system with non-uniform random feature selection. In: IGARSS, pp. 3915–3918. IEEE (2011)
9. Aviris hyperspectral data (2012), `http://aviris.jpl.nasa.gov/html/data.html`
10. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)
11. Verikas, A., Gelzinis, A., Bacauskiene, M.: Mining data with random forests: A survey and results of new tests. Pattern Recognition 44(2), 330–349 (2011)
12. Cumbaa, C., Jurisica, I.: Protein crystallization analysis on the world community grid. Journal of Structural and Functional Genomics 11, 61–69 (2010), doi:10.1007/s10969-009-9076-9
13. Fukunaga, K., Koontz, W.L.G.: Application of the karhunen-loéve expansion to feature selection and ordering. IEEE Trans. Comput. 19(4), 311–318 (1970)
14. Ochilov, S., Alam, M.S., Bal, A.: Fukunaga-koontz transform based dimensionality reduction for hyperspectral imagery. Proceedings-SPIE The International Society For Optical Engineering 6233, 62332A–62332A–8 (2006)
15. Li, Y.H., Savvides, M.: Kernel fukunaga-koontz transform subspaces for enhanced face recognition. 2012 IEEE Conference on Computer Vision and Pattern Recognition 0, 1–8 (2007)
16. Dinc, S., Bal, A.: A statistical approach for multiclass target detection. Procedia Computer Science, Complex A1daptive Sysytems 6(0), 225–230 (2011)
17. Hyperspectral remote sensing scenes, aviris sensor (indian pines) (2013), `http://www.ehu.es/ccwintco/index.php/Sensores-hiperespectrales`
18. Bagan, H., Yasuoka, Y., Endo, T., Wang, X., Feng, Z.: Classification of airborne hyperspectral data based on the average learning subspace method. IEEE Geoscience and Remote Sensing Letters 5(3), 368–372 (2008)

SOM++: Integration of Self-Organizing Map and K-Means++ Algorithms

Yunus Dogan, Derya Birant, and Alp Kut

Dokuz Eylul University, Department of Computer Engineering,
Tinaztepe Campus, Buca, 35397 Izmir, Turkey
{yunus,derya,alp}@cs.deu.edu.tr

Abstract. Data clustering is an important and widely used task of data mining that groups similar items together into subsets. This paper introduces a new clustering algorithm SOM++, which first uses K-Means++ method to determine the initial weight values and the starting points, and then uses Self-Organizing Map (SOM) to find the final clustering solution. The purpose of this algorithm is to provide a useful technique to improve the solution of the data clustering and data mining in terms of runtime, the rate of unstable data points and internal error. This paper also presents the comparison of our algorithm with simple SOM and K-Means + SOM by using a real world data. The results show that SOM++ has a good performance in stability and significantly outperforms three other methods training time.

Keywords: Data Mining, Clustering, Self-Organizing Map, K-Means++, Mining Methods and Algorithms.

1 Introduction

Cluster analysis is the process of grouping data into subsets such that each item in a cluster is more similar to the items in the same cluster than to the other items at the outside of the cluster. Generally, distance measures like Euclidean distance, Manhattan distance are utilized to evaluate the dissimilarity between data points. Cluster analysis is one of the most useful tasks in machine learning and data mining, and has been used in a variety of fields such as marketing, banking, medicine and telecommunication. It has been widely used in dimensionality reduction, information extraction, density approximation and data compression [15] [6] [7] [16].

The K-means [12] algorithm is the most commonly used partitioning cluster algorithm with its easy implementation and its efficient execution time. Self-organizing map (SOM) [11] is an unsupervised, well-established and widely used clustering technique.

In SOM, initial weight values are assigned randomly, method performance is sensitive to these values and it is prohibitively slow in large data applications. In order to decrease the time complexity of SOM, we investigated different initialization procedures for optimal SOM and now propose K-Means++ as the most convenient method, given the proper training parameters.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 246–259, 2013.

K-Means++ algorithm gives more successful results than standard K-Means at accuracy and consistency [3]. Because, the K-Means algorithm works only to find a local optimum and this local optimum often becomes poor by using random initial center points; however, K-Means++ starts with rational initial points, thus K-Means++ approximates the best clustering space. Also, K-Means++ outperforms in speed, too. K-Means++ guarantees O(logk) as the complexity time; however, K-Means has a complexity time as O(n^{kd+1} logn), where k is the number of clusters, n is the number of data and d is the Euclidean distance between two clusters [3].

This paper proposes a new clustering algorithm SOM++, which is composed by K-Means++ method followed by SOM clustering. The algorithm consists of two stages: First, using K-Means++ method to determine the initial weight values instead of assigning in randomly, then clustering task is done in the second stage by SOM as an unsupervised clustering method. The experimental results show that the proposed algorithm, SOM++, is considerably better than the conventional SOM based algorithms in terms of runtime, the rate of unstable data points and internal error. It generates similar clustering results with other SOM based clustering algorithms, however the use of it requires the smaller training time.

The rest of this paper is organized as follows. The second section reviews the literature and describes SOM, K-Means++ algorithm and how SOM and K-Means were combined in the previous studies. After the new clustering algorithm SOM++ is explained in detail and also the principle and the architecture of the algorithm are presented in the third section, we demonstrate how the proposed algorithm can be applied on a real world data, including dataset description, experimental setup, and performance analysis and clustering results in the fourth section. The experimental studies indicate with figures of map combinations and tables of error rates. Section 5 points out the comparison of the results of SOM++, simple SOM, SOM+K-Means and SOM+K-Means++ (include all phases of K-Means++). Finally, the summary, conclusions and future work are given in the Section 6.

2 K-Means++ & SOM

The complexity of SOM algorithm is O(NC), where N is the input vector size and C is the number of dataset presentation cycles. N contains n^2w as the multiply of the map size n^2 and the number of weights w. C contains n^2a as the multiply of the map size n^2 and the number of attributes a. The number of attributes is equal to the number of weights; therefore, the complexity of SOM algorithm obtains O(N^2)[14].

While simple SOM has been previously used in many applications, extended versions of SOM has also been proposed such as FSOM (Fast SOM) [15], ABSOM (Ant Based SOM) [6], and ESOM (Emergent SOM) [7].

When the performance of the K-Means++ algorithm were evaluated on four datasets and K-Means++ consistently outperformed K-Means, both by completing faster and by achieving a lower potential value. For example, on one dataset, K-Means++ terminates almost twice as fast while achieving potential function values about 20% better, on the larger dataset, it is obtained up to 70% faster and the potential value is better by factors of 10 to 1000. For this reason, we propose K-Means++ algorithm in this paper, instead of K-Means [3].

In recent years, several studies have compared different SOM-based two-stage methods. For instance, while Souza et al. [16] compared SOMK (SOM+K-Means) with SOMAK (SOM+Ant K-Means), Chi and Yang [5] compared both ABSOM (Ant-based Self-Organizing Map) with Kohonen's SOM individually, and SOM+K-Means with ABSOM+K-Means. Besides, Chiu et al. [7], [8] compares four approaches simple K-Means, SOM+K-Means, PSO+K-Means (Particle swarm optimization (PSO)) and PSHBMO (Particle Swarm Optimization with Honey Bee Mating Optimization). As another example, the study of Corrêa and Ludermir [9] about the comparison of several SOM-based two-stage approaches: SOM, SOM+KM (K-Means), SOM+W.KM (Weighted K-Means), SOM+AY (proposed by Azcarraga and Yap) and SOM+W.AY (Weighted AY Method), in terms of classification accuracy and runtime.

Recently, performing K-Means method after usage of SOM was also studied for different purposes as examples of the emergency planning to deal with extreme events such as earthquake, flood and fire [2], clustering meteorological data [10], the biological wastewater treatment process [1], and the identification of day types of electricity load [4]. Our proposed algorithm, SOM++ is a general clustering algorithm; as a result, it can be used in many different applications for different purposes.

To the best of our knowledge, however, this paper is the first in performing K-Means method before training neurons by usage of SOM to determine the initial weight values of SOM.

3 Methodology

The study of Su, Liu and Chang shows that the initializing the weight values increases the performance of SOM [17]. SOM++ shows that initializing the weight values by K-Means++ (without K-Means clustering) increases the performance of SOM. Also, the study of Attik, Bougrain and Alexandre mentions that initializing the weight values by K-Means clustering increases the performance of SOM without any example or method [3]; SOM++ is a supportive and integral study of these studies with K-Means++ (without K-Means clustering) and a new sequential assignment algorithms.

In this section, it is explained that our new algorithm SOM++, a two-stage clustering algorithm uses the combination of two data mining techniques, namely SOM and K-means++ clustering. SOM algorithm uses neurons for all points on its map and these neurons have weight values for all attribute values. Before showing the details of SOM++ algorithm, these weight values are indicated in the following part. In SOM, input neurons are fully connected to output neurons, and each connection has a weighting value. In the initialization process of SOM, each neuron is associated with a random weight vector (wi = wi1, wi2, …, win), which has the same dimension (n) as the input vector (xi = xi1, xi2, …, xin). Using the Euclidean measure, distance between the input vector and the incoming weight vector of each output map neuron is calculated. The output neuron with the smallest distance is declared the winner.

After that, neuron weights are subsequently updated according to (1) using a neighborhood function (2), which minimizes the overall distance between the neuron itself and its neighbors.

$$wij(t + 1) = wij + h(t)(xi - wij) \tag{1}$$

where wij(t) is the connection weight from input i to output neuron j at time t; xi is element i of input vector x, and h is the neighborhood function.

$$h(t) = \alpha GF \tag{2}$$

where α is the learning rate; GF is the Gaussian Function (3).

$$GF = exp\ (-\sum \| wui\ \text{-}\ cui \| 2/2\phi 2\ (t)) \tag{3}$$

where ϕ is the neighborhood width parameter and GF uses the Euclidean distance between the winner unit (wu) and the current unit (cu).

SOM method performance is sensitive to the randomly assigned initial weight values and it is prohibitively slowed in the large-scale applications. In order to decrease the time complexity of SOM, this paper proposes K-Means++ to determine the initial weight values, instead of random process. In this approach, K-Means++ centers are assigned as SOM weight values; thus, SOM will require fewer iterations. Since the K-means algorithm is more computationally efficient than SOM, the general solution will be faster.

3.1 Description of SOM++ Algorithm

The proposed SOM++ is a two-stage algorithm, which is a combination of SOM and the initialization method of centers in K-means++. Fig. 1 and Fig. 2 show pseudo codes for SOM++ algorithm. The algorithm starts by finding the center points for the all clusters by using the method of K-Means++ which initializes the centre points of the clusters. There must be two sets named as Φ and D. Set D collects the centers of all clusters during the first part of SOM++ (step 1 and step 6 in Fig. 1). Set Φ collects the distances between each data and each center (step 2 in Fig. 1). The distances are calculated by using the Euclidean Distance. According to sum of squares of distances in set Φ, the centers are obtained (step 3-4-5 in Fig. 1). After obtaining k centers in set D (step 7 in Fig. 1), the second part of SOM++ is started (Fig. 2).

After these steps, all centre points for all clusters are collected in a set Φ. These centers have the attribute values and these values must initialize the weight values of neurons on the map of SOM++. However, the most suitable method must be decided for locating these initial weight values. If the locating method is not suitable according to the distances of neurons, the result of SOM++ is not different from the result of the standard SOM with the random initialization (Comparing the error rates is given in Section 4.4). Therefore, a new sequence assignment algorithm is implemented by considering the distances between K centers in set Φ.

Fig. 2 and Fig. 3 show the pseudo code of this sequence assignment algorithm.

Firstly, the most different point in set D is calculated (step 1 in Fig. 2) and according to the Euclidean Distance, a sorting operation is done by comparing to this outlier point. At the end of sorting operation, a new set which has sorted points according to the least similar point is obtained (step 2). The values in these points are assigned to the weight values of neurons as the initial values for SOM++ algorithm like the sequence in Fig. 3 (step 3 in Fig. 2).

Finding initial weight values of neurons (K-Means++ part)	
1	**Select** data from the data set β randomly **Add** this data into a set *D*
	$D \leftarrow \beta_{Random(0,i)}$

2	**For each** data *i* in the dataset β
	    For each data *j* in the dataset *D*
	        Find the Euclidean Distances between βi and *Dj*
	    Add the minimum distance into set Φ

	$\forall i \left(\Phi \leftarrow min \left(\forall j \; (\sum_{k=0}^{n} \lVert \beta ik - Djk \rVert^{2}) \right) \right)$, *where n is equal to the number of attributes, i is equal to the number of data β and j is equal to the number of data in D*
3	**Find** the sum of squares *S* for the values in Φ
	$S = \sum_{k=0}^{i} \Phi k^2$, *where i is equal to the number of data in Φ*
4	**Select** a real number *R* between 0 and *S* randomly
	$R = Random(0, S)$
5	**Find** the unique integer *q* so that $1^2 + 2^2 + ... + q^2 >= R > 1^2 + 2^2 + ... + (q\text{-}1)^2$
	$\sum_{k=1}^{q} k^2 \geq R > \sum_{k=1}^{q-1} k^2$, *where* $q \in \mathbb{N}^{+}$, $q > 2$
6	**Add** q^{th} data in β into *D*
	$D \leftarrow \beta_q$
7	**Repeat** steps 2-3-4-5-6 **until** the number of data in *D* is equal to the number of clusters *K*

Fig. 1. The K-Means++ part of SOM++ algorithm

Before training operations in SOM++, the number of iteration is initialized as the number of total data (step 4 in Fig. 2), because actually, the neurons on the map of SOM++ become trained at the beginning; therefore, the number of iteration must not start zero. The advantages of initializing both the number of iteration and weight values of neurons with the values coming from K-Means++ are shown in the Section 4 and 5 in detail. Finally, training operations of neurons starts by using the standard SOM algorithm (step 5 in Fig. 2).

The weight values become close to the final and decisive values by means of K-Means++; on the other hand, it is not enough singly, because the single aim of SOM algorithm is not to do clustering of data correctly. It is also a mapping algorithm; therefore, the places of the clusters win the importance in SOM algorithm. If locating of the weight values which come from the initializing method of K-Means++ is done randomly, the correct neuron cannot have the correct weight values, and the success of SOM++ algorithm is not realized certainly.

In Section 4, the importance of the sequential assignment is tested in detail.

Our sequential assignment algorithm needs the sorted values according to the least similar value to each other in the set. Then, locating operation starts from the neuron in [0, 0] which is the one of the furthest four neurons ([0, 0], [0, (n-1)], [(n-1), 0] and [(n-1), (n-1)] as n^2 is the number of neurons) from the center on the map, because the top element in sorted values is the least similar value. The next values after the least similar value are located like the sequence in Fig. 3.

Initializing weight values of neurons and the number of iteration	
1	**Find** the least similar center *L* to the other centers in *D*
	$L = max\left(\forall i \left(\forall j \left(\sum_{k=0}^{n} \|Dik - Djk\|^2 \right) \right) \right)$, *where n is equal to the number of attributes, i is equal to the number of data in D and j is equal to i+1 for each i.*
2	**Sort** the centers in *D* from the most similar center to the least similar center according to *L* by using the Euclidean Distance **Collect** these centers in *θ*
	$\theta \leftarrow L,$ $\forall i \begin{pmatrix} L \notin D \\ L = Dmin\left(\forall i \left(\sum_{k=0}^{n} \|Lk - Dik\|^2 \right) \right), \\ \theta \leftarrow L \end{pmatrix}$, *where n is the number of attribute and i is equal to the number of data in D*
3	**Initialize** attribute values of these sorted centers into the weight values of neurons on the map sequentially like the sequence in Fig 3.
4	**Initialize** the number of iteration as the number of the data
	$I = N$, *where N the number of* data in the dataset *β and I is the iteration number of the standard SOM algorithm*
5	**Start** the standard SOM algorithm

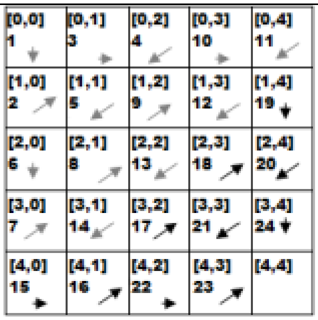
Fig. 2. Initializing parameters part of SOM++ algorithm

The steps on the left and right sides are done to down and inner-cross directions. The steps on the top and bottom sides are done to left and inner-cross directions; however, the steps on the center side are done mix-cross directions. Finally, the furthest values are located in the furthest neurons on the map.

4 Experimental Studies

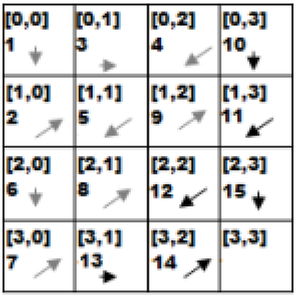
The error rates and stability of the maps are tested by a dataset with 699 vectorial tuples. These datasets have an attribute which puts the correct classes of all data. However, the datasets are used without this attribute while SOM clustering. Because SOM is a clustering algorithm and does not need a target attribute while training neurons. This attribute is used while testing the stability of the maps.

1	$x = 0, y = 0, n = 0, \forall i\ (M[x,y].W[i] = \theta[n].W[i])$, *where i is the number of the weights in a neuron, M is the map matrix, W is the weight array in a neuron, x and y are the indexes of M, and n is the index of the* θ
2 ↓	$y < BY \Rightarrow \big(y = y + 1,\quad n = n + 1,\ \forall i\ (M[x,y].W[i] = \theta[n].W[i])\big)$, *where BY is the boundary of y in M*
3 ↗	$\exists x, y[\ (y < BY\ \Lambda\ x < BX) \Rightarrow$ $(x = x + 1,\quad y = y - 1,\quad n = n + 1,\ \forall i\ (M[x,y].W[i] = \theta[n].W[i])],$ *where BX is the boundary of x in M*
4 →	$x < BX \Rightarrow \big(x = x + 1,\quad n = n + 1,\ \forall i\ (M[x,y].W[i] = \theta[n].W[i])\big)$, *where BX is the boundary of x in M*
5 ↙	$\exists x, y[\ (x < BX\ \Lambda\ y < BY) \Rightarrow$ $(x = x - 1,\quad y = y + 1,\quad n = n + 1,\ \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$
6	**Repeat** 2-3-4-5 **until** $x = \begin{cases} 0\ , & BX \bmod 2 = 0 \\ BX\ , & BX \bmod 2 = 1 \end{cases}\ \Lambda\ \ y = \begin{cases} BY\ , & BY \bmod 2 = 0 \\ 0\ , & BY \bmod 2 = 1 \end{cases}$
7 ↙	$\exists x, y[\ (x < BX\ \Lambda\ y < BY) \Rightarrow$ $(x = x - 1,\quad y = y + 1,\quad n = n + 1,\ \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$
8 →	$x < BX \Rightarrow \big(x = x + 1,\quad n = n + 1,\quad \forall i\ (M[x,y].W[i] = \theta[n].W[i])\big)$
9 ↗	$\exists x, y[\ (y < BY\ \Lambda\ x < BX) \Rightarrow$ $(x = x + 1,\quad y = y - 1,\quad n = n + 1,\ \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$
10 ↓	$y < BY \Rightarrow \big(y = y + 1,\quad n = n + 1,\quad \forall i\ (M[x,y].W[i] = \theta[n].W[i])\big)$
11	**Repeat** 7-8-9-10 **until** $x = BX - 1\ \ \Lambda\ \ y = BY$
12	$x = x + 1, n = n + 1, \forall i\ (M[x,y].W[i] = \theta[n].W[i])$,



[0,0] 1 ↓	[0,1] 3 →	[0,2] 4 ↙	[0,3] 10 →	[0,4] 11 ↙
[1,0] 2 ↗	[1,1] 5 ↙	[1,2] 9 ↗	[1,3] 12 ↙	[1,4] 19 ↓
[2,0] 6 ↓	[2,1] 8 ↗	[2,2] 13 ↙	[2,3] 18 ↗	[2,4] 20 ↙
[3,0] 7 ↗	[3,1] 14 ↙	[3,2] 17 ↗	[3,3] 21 ↙	[3,4] 24 ↓
[4,0] 15 →	[4,1] 16 ↗	[4,2] 22 →	[4,3] 23 ↗	[4,4]

where BX = BY = 4. The first loop terminates after the 14 steps by using 2nd, 3rd, 4th and 5th conditions. The second loop terminates after the 24 steps by using 7th, 8th, 9th and 10th conditions.



[0,0] 1 ↓	[0,1] 3 →	[0,2] 4 ↙	[0,3] 10 ↓
[1,0] 2 ↗	[1,1] 5 ↙	[1,2] 9 ↗	[1,3] 11 ↙
[2,0] 6 ↓	[2,1] 8 ↗	[2,2] 12 ↙	[2,3] 15 ↓
[3,0] 7 ↗	[3,1] 13 →	[3,2] 14 ↗	[3,3]

where BX = BY = 3. The first loop terminates after the 9 steps by using 2nd, 3rd, 4th and 5th conditions. The second loop terminates after the 15 steps by using 7th, 8th, 9th and 10th conditions.

Fig. 3. Sequential assignment of the initial weight values which come from K-Means++

After training neurons without the target attribute, all correct classes for each data are obtained. Because a neuron can contain more data than one, containing the elements of the same class or not could be showed with colored neurons. For the visual compares, the dataset with 699 vectorial tuples is used. This dataset contains 10 attributes and it is about Breast Cancer Wisconsin (BCW) [19]. There are two certain classes and each neuron which contains the elements of the same class is shown by a different color on maps.

On all maps in the following figures, there are three different colored neurons as silver, grey and black. The silver and grey neurons show the correct clustered neurons. In the other words, these neurons contain the elements of the same class; however, the black neurons contain the elements of the different classes and the number of black neurons shows the instability of the map.

4.1 Visual Compares at the Beginning Maps

Before starting to train neurons in SOM algorithm, initializing the weight values of neurons by a pre-treatment supplies a stability to the map at the beginning immediately. In the following versions, the distinction, between the sequential assignments according to the similarities of the centre points which are returned from K-Means clustering algorithms and the random assignments of them, is observed, too.

In Fig. 4, the beginning map for the standard SOM algorithm with 20x20 neurons is shown. The initial weight values are assigned randomly in the version of the standard SOM; therefore, the map is observed indecisively.

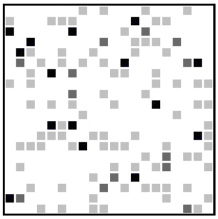


Fig. 4. Standard SOM at the beginning phase by using 20x20 neurons

On the first map in Fig. 5, there are neurons with initialized weight values by the centre points which are returned from K-Means clustering. On the second map, there are neurons with initialized weight values by the initial centre points of K-Means++ without K-Means clustering. On the third map, there are neurons with initialized weight values by the centre points which are returned from K-Means++ clustering.

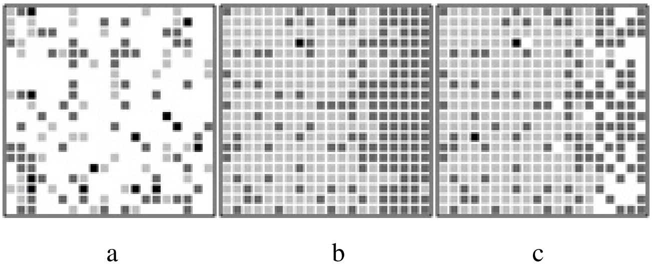


Fig. 5. SOM is at the beginning phase by using 20x20 neurons without any training a) K-Means, b) K-Means++ (without K-Means phase), c) K-Means++

These weight values are not located by using the sequential assignment and these neurons on the maps are at the beginning phase of SOM algorithm. However, it seems that the instability cannot be prevented in these examples.

In Fig. 6, the sequential assignment is used and neurons are located according to the similarities. The importance of the sequential arraignment is observed from the maps in Fig. 6.

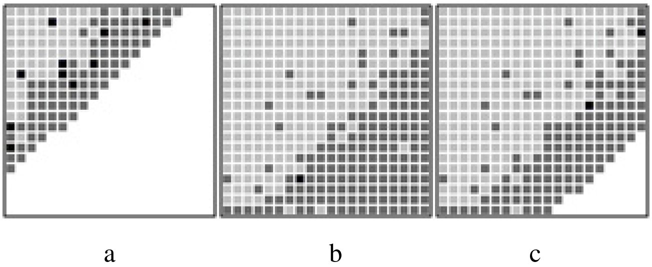


Fig. 6. SOM is at the beginning phase by using 20x20 neurons with initialized weights by the sequential assignment. a) K-Means, b) K-Means++ (without K-Means phase), c) K-Means++ (with K-Means Clustering)

It is observed at the visual tests that the successes of the new versions of SOM algorithm have more stability and less indecisive neurons than the standard SOM. Also, visually, it can be obtained that using K-Means++ without K-Means clustering has a near success together with using K-Means++ with K-Means clustering for the sequential initialized weight values and initialized number of iteration of SOM as the number of data (699 in the previous examples).

The numerical comparisons of the new versions are done by calculating the error rates at the phase of training neurons in SOM algorithm.

4.2 Visual Compares at the Beginning Maps

The indecisive neurons are shown by black neurons on the previous maps. They mean that irrelevant data tuples are in same neurons on the map and if the number of these neurons is high, the map is not consistent.

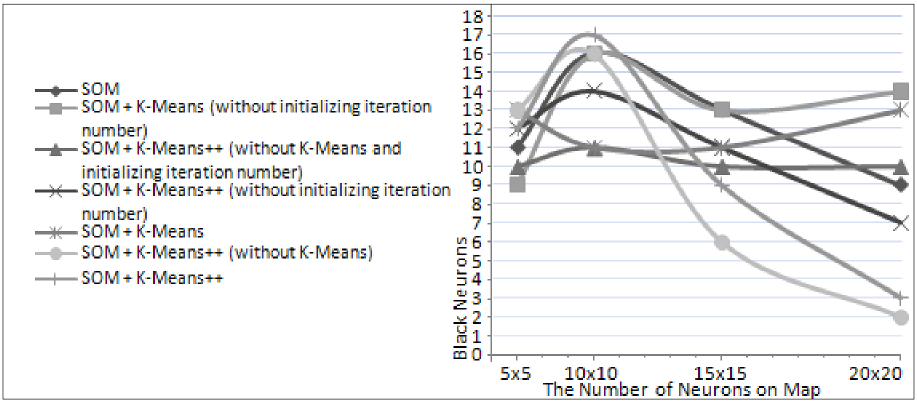


Fig. 7. The comparison of indecisive neurons

The numbers of indecisive neurons in the previous visual compares are collected like the graph in Fig. 7. This graph shows that because the numbers of total neurons for the first 5x5 neurons are low, the numbers of indecisive neurons are low, too. Therefore, the numbers of indecisive neurons are higher for 10x10 neurons and the versions of SOM could be compared according to the numbers of the indecisive neurons for 10x10, 15x15 and 20x20 neurons.

The steadiest algorithms are SOM++ (or SOM + K-Means++ (without K-Means)) and SOM + K-Means++ according to the graph. Both of these algorithms use both initializing iteration number as the number of data and the sequential assignment algorithm.

As a result, the most successful algorithm is SOM++ with the least number of indecisive neurons as 6 for 15x15 neurons and only 2 for 20x20 neurons.

4.3 The Error Rates

Error rates are calculated according to is the Gaussian Function in Eq. 3 and they are obtained for all versions of algorithms like the other comparison tests; however, Table 1 shows the least error results only for K-Means++ without K-Means clustering and with K-Means clustering.

In Table 1, a1) Error rates for K-Means++ (without K-Means) + SOM; error rates for K-Means++ (without K-Means clustering) + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations b1) Error rates for K-Means++ (without K-Means)+ SOM with the initialized number of iteration as the number of total data (699) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c1) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized number of iteration as the number of total data (699) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations.

Error Rates for K-Means++ (with K-Means Clustering) + SOM; a2) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations b2) Error rates for K-Means++ (with K-Means Clustering)+ SOM with the initialized number of iteration as the number of total data (699) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c2) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized number of iteration as the number of total data (699) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations.

Table 1. The error rates list for 25 and 100 neurons

	Iteration Number	5x5 neurons	10x10 neurons
a1	1^{th}	0.42720	0.49013
	2^{nd}	0.01000	0.01873
	3^{rd}	0.00518	0.00177
b1	1^{th}	1.61001	0.68055
	2^{nd}	0.08828	0.01924
	3^{rd}	0.03508	0.00733
c1	1^{th}	0.28725	0.05095
	2^{nd}	0.00252	0.00058
	3^{rd}	0.00125	0.00007
a2	1^{th}	0.43723	0.47991
	2^{nd}	0.05145	0.01509
	3^{rd}	0.01086	0.00112
b2	1^{th}	1.81375	0.67474
	2^{nd}	0.09418	0.01972
	3^{rd}	0.03524	0.00777
c2	1^{th}	0.28313	0.08267
	2^{nd}	0.00795	0.00029
	3^{rd}	0.00298	0.00024

4.4 Training Times

The tests of training times are implemented by using the dataset with 18781 vectorial tuples and 390 attributes. This large dataset is produced the distinctions on performance of trainings between simple SOM, SOM++ (with the initialization method of K-Means++), SOM + K-Means++ and SOM + K-Means absolutely.

The computer, which tests the performances of the algorithms for this large dataset, has 3.24 GB of RAM and Intel(R) Core(TM) 2 Duo CPU E6550 @ 2.33 GHz. This computer trains 600x600 neurons with this large dataset which has a large attribute set for three weeks.

The Fig. 8 shows that the simple SOM trains 600x600 neurons since the first moment; however, SOM++ (with the initialization method of K-Means++), SOM +

K-Means and SOM + K-Means++ need a preparation time for SOM. Therefore, the error rates of SOM are observed stably until a few times for these versions of SOM. These times are less than an hour for SOM++, about 2 hours for SOM + K-Means++ and about 3 hours for SOM + K-Means, because there are lots of attributes and tuples in the dataset. After these times, the simple SOM starts with initialized weight values and iteration values for these versions.

At the end of 7 days, the simple SOM gives about 2.4 of the error rate; however, SOM++ gives a less error rate than $10x10^{-7}$ before 8 days. Also, the error rate for SOM + K-Means++ is taken a less error rate than $10x10^{-7}$ before 9 days and the error rate for SOM+ K-Mean is taken a less error rate than $10x10^{-7}$ before 10 days. These results show that the versions of SOM have better performances than the simple SOM. However, SOM++ has the best performance. Because of the complexity of SOM algorithm as $O(N^2)$ where N is the input vector size (N is 390 x 18781 and N^2= 53649618668100), SOM algorithm gets the result map with minimum error rates after some days.

Also, it is observed that the initialization method of center points at K-Means++ accelerates K-Means, because K-Means++ with all steps needs about 2 hours to cluster a large dataset. However, K-Means needs about 3 hours.

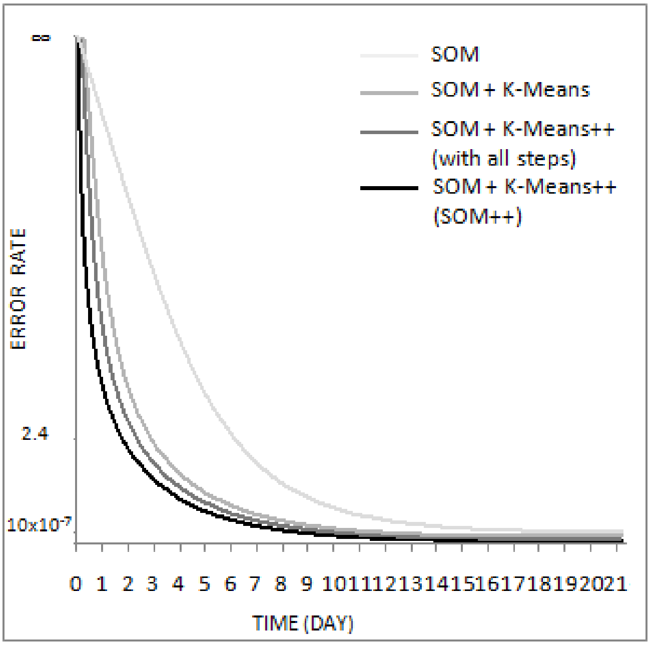


Fig. 8. The error rates - time (day) graphic for the versions of SOM

5 Comparison Results

The accuracy, about which of the SOM versions must be used, is taken by the visual tests. It is observed that the SOM versions of K-Means++ without K-Means clustering and with K-Means clustering have the least number of indecisive neurons in the visual tests. If the versions of K-Means++ with K-Means clustering and without K-Means clustering have a close success, a comparison of time can do the distinction between them and it is observed that K-Means++ (without K-Means clustering) + SOM has the best results.

Consequently, our experimental results empirically prove that K-Means++ (without K-Means clustering) + SOM (SOM++ with its short name) is best suited to data clustering due to its high speed and lower error rates as compared with other SOM based techniques.

6 Summary, Conclusion and Future Work

This paper introduces a new clustering algorithm SOM++. The significant difference between SOM++ algorithm and the standard SOM is that SOM++ does not start to initialize the weight values of neurons with random numbers. SOM++ uses the initializing center points of clusters method in K-Means++. Eventually, each neurons represent a cluster and thus, SOM++ takes advantage of K-Means++. Separately, these significant initial values are not located in the neurons on the map and SOM++ has a special locating algorithm namely the sequential assignment.

Another difference between SOM++ algorithm and the standard SOM is that SOM++ initializes the starting number of iteration. Because the standard SOM starts with the random values in neurons, the number of iteration is declared as 0. However, because SOM++ starts with significant values in neurons, the number of iteration is declared as the number of total data. This initializing increases stability and decreases error rates.

In other words, SOM++ algorithm has many advantages over conventional SOM based methods. The most remarkable advantage of SOM++ is in saving training time for clustering large and complicated data sets by using K-Means++ algorithm in the weight initialization procedure of SOM. Furthermore, the rate of unstable data points decreases and internal error decreases.

For future work, the proposed algorithm, SOM++**,** can be used for the computer security, the healthcare, ecological modeling, the financial sector and another area which needs clustering its large data successfully.

References

1. Aguado, D., Montoya, T., Borras, L., Seco, A., Ferrer, J.: Using SOM and PCA for Analysing and Interpreting Data from a P-removal SBR. Engineering Applications of Artificial Intelligence 21(6), 919–930 (2008)
2. Arthur, D., Vassilvitskii, S.: K-Means++ the Advantages of Careful Seeding. In: Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1027–1035 (2007)
3. Attik, M., Bougrain, L., Alexandre, F.: Self-organizing map initialization. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 357–362. Springer, Heidelberg (2005)
4. Benabbas, F., Khadir, M.T., Fay, D., Boughrira, A.: Kohonen Map Combined to the K-Means Algorithm for the Identification of Day Types of Algerian Electricity Load. IEEE Proc. 7th Computer Information Systems and Industrial Management Applications, 78–83 (2008), doi:10.1109/CISIM.2008.27

5. Chi, S.-C., Yang, C.-C.: A Two-stage Clustering Method Combining Ant Colony SOM and K-means. Journal of Information Science and Engineering 24, 1445–1460 (2008)
6. Chi, S.-C., Yang, C.C.: Integration of Ant Colony SOM and K-Means for Clustering Analysis. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4251, pp. 1–8. Springer, Heidelberg (2006)
7. Chiu, C.-Y., Chen, Y.-F., Kuo, I.-T., Ku, H.-C.: An Intelligent Market Segmentation System Using K-Means and Particle Swarm Optimization. Expert Systems with Applications 36(3), 4558–4565 (2008)
8. Chiu, C.-Y., Kuo, I.-T., Chen, P.-C.: A Market Segmentation System for Consumer Electronics Industry Using Particle Swarm Optimization and Honey Bee Mating Optimization. Global Perspective for Competitive Enterprise, Economy and Ecology pt. 12, ch. 1 (2009)
9. Corrêa, R.F., Ludermir, T.B.: A Hybrid SOM-Based Document Organization System. In: IEEE Proc. 9th Brazilian Symposium on Neural Networks (SBRN 2006), pp. 90–95 (2006), doi:10.1109/SBRN.2006.3
10. Khedairia, S., Khadir, M.T.: Self-Organizing Map and K-Means for Meteorological Day Type Identification for the Region of Annaba -Algeria-. In: IEEE Proc. 7th Computer Information Systems and Industrial Management Applications, pp. 91–96 (2008), doi:10.1109/CISIM.2008.29
11. Kohonen, T.: The Self-Organizing Map. Proc. of the IEEE 78(9), 1464–1479 (1990), doi:10.1109/5.58325
12. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. of 5th Berkeley Symposium on Mathematical Statistic and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
13. Poelmans, J., Elzinga, P., Viaene, S., Van Hulle, M.M., Dedene, G.: How Emergent Self Organizing Maps can Help Counter Domestic Violence. In: IEEE Proc. 2009 WRI World Congress on Computer Science and Information Engineering (CSIE), Los Angeles, USA, vol. 4, pp. 126–136 (2009), doi:10.1109/CSIE.2009.299
14. Roussinov, D.G., Chen, H.: A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation. Communication and Cognition in Artificial Intelligence Journal 15(1-2), 81–111 (1998)
15. Sagheer, A., El., T.N., Maeda, S., Taniguchi, R., Arita, D.: Fast Competition Approach using Self Organizing Map for Lip-Reading Applications. In: IEEE Proc. International Joint Conference on Neural Network (IJCNN), pp. 3775–3782 (2006), doi:10.1109/IJCNN.2006.1716618
16. Souza, J.R., Ludermir, T.B., Almeida, L.M.: A Two Stage Clustering Method Combining Self-Organizing Maps and Ant K-Means. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 485–494. Springer, Heidelberg (2009)
17. Su, M.-C., Liu, T.-K., Chang, H.T.: Improving the self-organizing feature map algorithm using an efficient initialization scheme. Tamkang Journal of Science and Engineering 5(1), 35–48 (2002)
18. Wolberg, W.H.: Breast Cancer Wisconsin (Original) Dataset (1992), `http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29`
19. Yang, Y., Rong, L.: Establishment of the Evaluation Index System of Emergency Plans Based on Hybrid of SOM Network and K-means Algorithm. In: IEEE Proc. 4th International Conference on Natural Computation, pp. 347–351 (2008), doi:10.1109/ICNC.2008.454

Satellite Image Mining for Census Collection: A Comparative Study with Respect to the Ethiopian Hinterland

Kwankamon Dittakan[1], Frans Coenen[1], and Rob Christley[2]

[1] Department of Computer Science,
University of Liverpool, Liverpool, L69 3BX, United Kingdom
[2] Department of Epidemiology and Population Health,
University of Liverpool, Leahurst Campus,
Chester High Road, CH64 7TE Neston, Cheshire, United Kingdom
{dittakan,coenen,robc}@liverpool.ac.uk

Abstract. Census data provides an important source of information with respect to decision makers operating in many different fields. However, census collection is a time consuming and resource intensive task. This is especially the case in rural areas where the communication and transportation infrastructure is not as robust as in urban areas. In this paper the authors propose the use of satellite imagery for census collection. The proposed method is not as accurate as "on ground" census collection, but requires very little resource. The proposed method is founded on the idea of collecting census data using classification techniques applied to relevant satellite imagery. The objective is to build a classifier that can label households according to "family" size. More specifically the idea is to segment satellite images so as to obtain pixel collections describing individual households and represent these collections using some appropriate representation to which a classifier generator can be applied. Two representations are considered, histograms and Local Binary Patterns (LBPs). The paper describes the overall method and compares the operation of the two representation techniques using labelled data obtained from two villages lying some 300km to the northwest of Addis Ababa in Ethiopia.

Keywords: Data Mining, Image Classification, Satellite Image Analysis, Satellite Image Mining, Census Analysis, Census Mining.

1 Introduction

National census data provides an important source of statistical information with respect to planners and decision makers working in a wide diversity of domains. Census data is seen as an important source of information with which to measure the "well being" of a population and to provide input to national and regional development projects (both economic and social). For example development of public services such as education, health and transport services. Census data is typically obtained using a questionnaire format either for self-completion by individuals (in which case they are typically distributed by post or electronically), or for completion by field staff. There are a number

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 260–274, 2013.

of obstacles to the collection of national census data: it is both time consuming and resource intensive (with respect to both collection and analysis) while at the same time people are often reluctant to participate (typically they are suspicious of the motivation for the census). In financial terms the cost of census collection is often substantial. For example it has been suggested that the 2006 Australian national census cost an estimated $300 million Australian dollars, while the 2010 US census was expected to cost more than $11 billion US dollars and involve a million part time employees[1]. The difficulties associated with census collection are compounded in rural areas where the population tends to be sparser and the communication and transport infrastructure tends to be less robust than in rural areas [8].

The solution proposed in this paper is founded on the idea of using satellite imagery to generate census data by segmenting images, identifying households and using a classifier to predict household size (number of people living in the household). Given a training set of hand-labeled households we can build a classifier to predict household type and size and use this to generate census information. The proposed approach is not applicable with respect to all areas (such as inner city areas where population estimates are difficult to obtain from satellite imagery) but is applicable in more rural areas. The focus for the study is the Ethiopia hinterland. The advantages offered by the proposed approach are: (i) low cost, (ii) speed of collection and (iii) automate processing. The disadvantage is that it will not be as accurate as more traditional "on ground" census collection, however it is suggested that the advantages out weigh the disadvantages.

The main challenge of the proposed census collection method, with respect to the work presented in this paper, is how best to represent the image data so that classifier generation techniques can be applied and census data effectively collected. Two image representations are considered: (i) Colour Histograms and (ii) Local Binary Patterns (LBPs). These are two techniques that have been "tried and tested" with respect to other image analysis applications. Histogram representations have been widely used for *whole image* representation (see for example [11]) because they offer the advantage that they obviate the need for image object identification. LBPs, alternatively, have been extensively used with respect to texture analysis [16].

The proposed approach is fully described in the remainder of this paper and evaluated using test data collected from two villages lying some 300km to the northwest of Addis Ababa in Ethiopia. The rest of this paper is organised as follows. In Section 2 some previous works is presented. Section 3 provides detail of the geographical study area in Ethiopia used for evaluation purposes. Section 4 then provides a description of the proposed census mining framework. Section 5 describes the proposed colour histogram based representation and Section 6 the proposed LBP representation. Section 7 reports on the evaluation of the framework. Finally, a summary and some conclusions are presented in Section 8.

2 Previous Work

Image understanding is an important and fundamental problem in domains such as computer vision and pattern recognition where the main objective is to understand

[1] `http://usgovinfo.about.com/od/censusandstatistics/a/aboutcensus.htm`

the characteristics of an image and interpret its semantic meaning. Image classification is an emerging image interpretation technique which can be used to categorise image sets according to a predefined set of labels. The performance of classifiers depends on the quality of the features used, features such as colour and texture. One method of encapsulating image colour is to use a histogram image representation technique whereby colour histograms represents the number of pixels associated with a particular subset of colours. The advantages offered by histogram-based representation are: (i) low storage requirements, (ii) automated generation and (iii) fast querying. Histogram representations have been used with respect to many applications including image retrieval [19,4,13] and remote sensing applications such as land usage analysis [1,2] and land change detection [14,3]. The histogram representation is one of the representations considered in this paper.

Texture is an important feature with respect to both human and computer vision. one example where texture analysis has been usefully employed is with respect to pattern recognition [21]. There are three principle mechanisms that may be adopted to describe the texture in digital images: (i) statistical, (ii) structural and (iii) spectral. The statistical approach is concerned with capturing texture using quantitative measures such as “smooth”, “coarse” and “grainy”. Structural approaches describe image texture in terms of a set of texture primitives or elements (texels) that occur as regularly spaced or repeating patterns. In the spectral approach the image texture features are extracted by using the properties of (say) the Fourier spectrum domain so that “high-energy narrow peaks” in the spectrum can be identified [9]. Local Binary Patterns (LBPs) are a texture representation method which is both statistical and structural in nature [17]. Using the LBP approach a binary number is produced, for each pixel, by thresholding its value with its neighbouring pixels. LBP offers the advantages of tolerance with respect to illumination changes and its computational simplicity. The LBP method has been used in many application such as face recognition [10,20]. The LBP representation is the second representation considered in this paper.

Remote Sensing is concerned with techniques for observing the Earths surface, or its atmosphere, using sensors located on spacecraft or aircraft platforms and producing images of regions on the earths surface as a result. Satellite image interpretation offers advantages with respect to many applications for example: geoscience studies, astronomy, military intelligence, and geographic information systems [6,12]. There are a small number of reports available on the use of satellite imagery for census data collection purposes. For example “nightlight” satellite images have been used to produce population census data and to analysis issues concerned with population density at the “sub-district level” [5]. In [15] classification techniques were applied to satellite image data to estimate the population density distribution with respect to one kilometre “blocks”. The difference between the work described in [15] and that proposed in this paper is that the considered approach operates at a much finer level of granularity. The authors have themselves conducted some previous work concerned with the application of classification techniques to satellite imagery to generate census data. This is described in [7]. The work attempted to define satellite image data using an earlier version of the histogram based approach presented in this paper, the evaluation was also directed at a much smaller data set and therefore not conclusive.

3 Case Study Application Domain

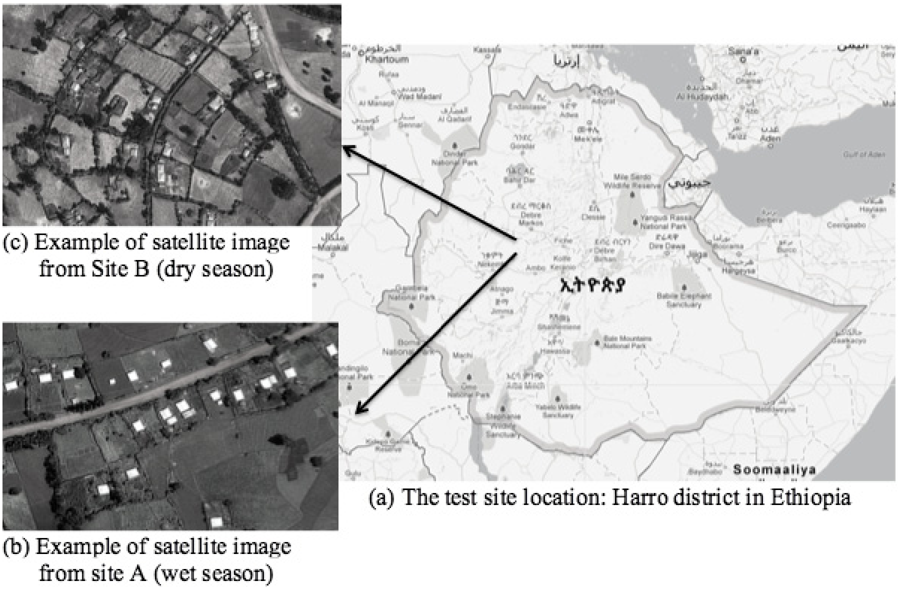
To act as a focus for the research a case study was considered directed at a rural area within the Ethiopian hinterland, more specifically two data sets were collected with respect to two villages (Site A and Site B) located within the Harro district in the Oramia Region of Ethiopia (approximately 300 km north-west of Addis Abba) as shown in Figure 1. Site A was bounded by the parallels of latitude 9.312650N and 9.36313N, and the meridians of longitude 37.123850E and 37.63914E and. Site B was bounded by the parallels of latitude 9.405530N and 9.450000N, and the meridians of longitude 36.590480E and 37.113550E. Using the know bounding latitudes and longitudes of our two test sites appropriate satellite imagery was extracted from Google Earth[2]. The images were originally obtained using the GeoEye satellite with a 50 centimetre ground resolution. The satellite images for Site A were released by Google Earth on 22 August 2009 (Figure 1(b)) and those for Site B (Figure 1(c)) on 11 February 2012. The Site B satellite images were obtained during the "dry season" (September to February), while the site A images were obtained during the rainy season (June to August). From Figure 1(b) the households can be clearly identified, many of the households have tin roofs which are easy to differentiate from the (green) backgrounds, the households are less easy to identify in Figure 1(a) where they tend to merge into the (light-brown) background. On-the-ground household data (including family size and household latitude and longitude) was collected by University of Liverpool field staff in May 2011 and July 2012. The minimum and maximum family size were 2 and 12 respectively, the mean was 6.31, the medium were 6 and standard deviation was 2.56. These two data sets then provided the training and test data required for our proposed census collection system.

4 Census Mining Framework

The proposed census mining framework is presented in this section. A schematic of the proposed framework is given in Figure 2. The framework supports a three phase census collection form satellite imagery process: (i) Data preprocessing, (ii) Classifier generation and (iii) Classifier evaluation.

During the data preprocessing phase (left hand block in Figure 2) the satellite image input data is prepared ready for the application of the classifier generation phase. The preprocessing stage comprises five individual stages: (i) coarse segmentation. (ii) image enhancement, (iii) detailed segmentation, (iv) image representation and (v) feature selection. During coarse segmentation the input imagery is roughly segmented to give a set of large scale sub-images each covering a number of households (typically between two and four). In the next stage various image enhancement processes are applied to the identified sub-images. During the detailed segmentation stage the enhanced coarse sub-images are segmented to obtain individual households. Figures 3(a) and (b) show two example of segmented household images taken from Site A and Site B respectively. The result is one image per household. For classifier generation purposes each labeled segmented households must be represented in a manner that ensures that the salient

[2] http://www.google.co.uk/intl/en_uk/earth/index.html



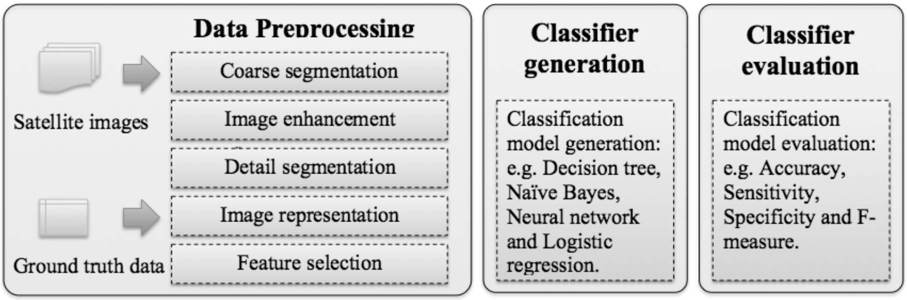
(c) Example of satellite image from Site B (dry season)

(a) The test site location: Harro district in Ethiopia

(b) Example of satellite image from site A (wet season)

Fig. 1. The test site location: Harro district in Ethiopia

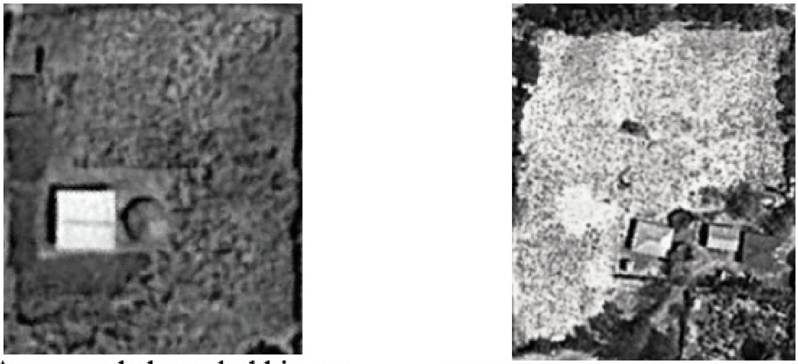
features are maintained (so as to ensure an effective classifier); as noted in the introduction to this paper two representation techniques are considered: a histogram based technique and an LBP based technique. The final step in the preprocessing phase comprised feature selection, the aim here was to reduce the overall size of the feature space (histogram based or LBP based) so that those features that best served to discriminate between classes were retained. For details concerning steps 1 to 3 the reader is referred to the authors earlier work presented in [7]. The histogram and LBP representations to support effective classifier generation are amongst the main contributions of this paper and are considered in more detail in Sections 5 and 6 respectively.



Data Preprocessing
Satellite images
Ground truth data
Coarse segmentation
Image enhancement
Detail segmentation
Image representation
Feature selection
Classifier generation
Classification model generation: e.g. Decision tree, Naïve Bayes, Neural network and Logistic regression.
Classifier evaluation
Classification model evaluation: e.g. Accuracy, Sensitivity, Specificity and F-measure.

Fig. 2. Proposed census data from satellite imagery framework

Classifier generation is the second phase (centre block in Figure 2) in the proposed framework during which the desired classifier was generated from labeled training data produced during the data preprocessing phase (Phase1) described above. The final phase (right hand block in Figure 2) was classifier evaluation where the classifier was applied to a labelled test set and the generated results compared with known results, the aim was to produce statistical measures indicating the confidence that can be associated with the generated classifier.


(a) An example household image from Site A (wet season)

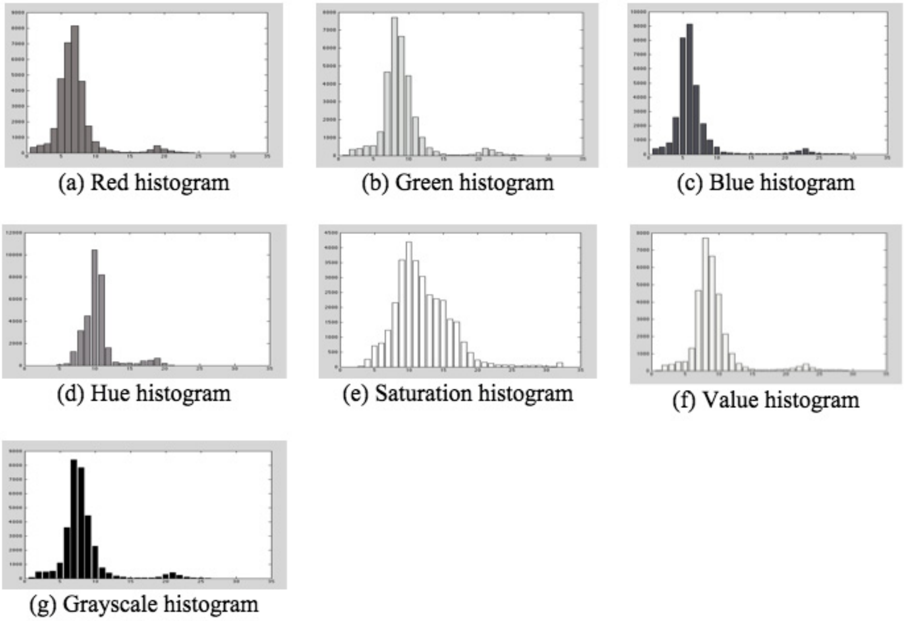
(b) An example household image from Site B (dry season)

Fig. 3. Example segmented household images

5 Color Histogram

In the histogram based satellite image representation image colour is the central feature used. Image colour distribution is captured using a set of histograms (one set per satellite image). The advantage offered is that histograms are simple to generate, invariant to translation and rotation of image content, low on storage requirement and allow for fast query execution. The X-axis of each histogram comprises a number of “bins” each representing a “colour range”. The Y axis of each histogram then represents the number of pixels falling into each bin. For each preprocessed household satellite image seven different histograms was extracted: (i) three histogram from the RGB colour channels (red, green, blue), (ii) three histograms from the HSV colour channels (hue, saturation, value) and (iii) a grayscale histogram. Each of the seven histograms comprised 32 bins, giving 224 (7 × 32) features in total. Figure 4 shows the seven example histograms produced from one of the identified household image used in the evaluation presented below (Section 7).

A simple alternative representation was to extract some simple statistical colour information from the image data. The idea here was that this statistical information could be used to augment the colour histogram information (or used as a representation


(a) Red histogram (b) Green histogram (c) Blue histogram

(d) Hue histogram (e) Saturation histogram (f) Value histogram

(g) Grayscale histogram

Fig. 4. Histogram representation for an example image

on its own). A total of 13 statistical features were identified: (i) 5 features describing the RGB colour channels, (ii) 5 features describing the HSV colour channels and (iii) 3 feature describing the grayscale channel. The individual features are listed in Table 1. Thus on completion of the histogram based representation stage each household is represented using a feature vector of length 237 ($224+13=237$).

Table 1. Additional colour based statistical features

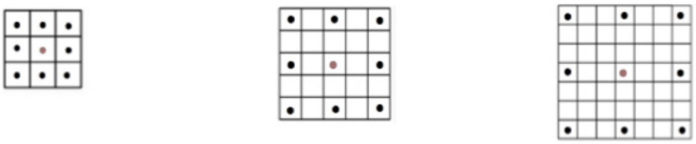
#	RGB color channel descripton	#	HSV color channel description	#	Grayscale channel description
1	Average red	6	Average hue	11	Mean of grayscale
2	Average green	7	Average saturation	12	Standard deviation of grayscale
3	Average blue	8	Average value		
4	Mean of RGB	9	Mean of HSV	13	Average of grayscale histogram
5	Standard deviation of RGB	10	Standard deviation of HSV		

6 Local Binary Pattern

As already noted, Local Binary Patterns (LBPs) are generally used for representing image texture. However, there is no reason why LBPs cannot be used to represent images irrespective of whether we are interested in texture or not. The LBP representation offers the advantages that they are easy to generate and tolerant against illumination changes. The use of LBPs was therefore considered as an alternative to the proposed histogram based representation.

To generate a set of LBPs from individual household images the images were first transform into grayscale. A 3 × 3 pixel window, with the pixel of interest at the centre, was then used as the basic "neighbourhood" definition with respect to the LBP representation. For each neighbourhood the grayscale value for the centre pixel was defined as the threshold value with which the surrounding eight neighbourhoods were compared. For each neighbourhood pixel a 1 was recorded if the grayscale value of the neighbourhood pixel was greater than the threshold, and a 0 otherwise. The result is an eight digit binary number. In other words 256 (2^8) different patterns can be described (note that LBPs calculated in this manner are not rotation invariant).

Variations for the basic LBP concept can be produced by using different sizes (radii) of neighbourhoods. These variations can be described using the (P,R) notation where P is the number of sampling points and R is the radius surrounding the centre pixel [18]. For evaluation purposes three different variations of the LBP representation were used (Figure 5): LBP(8,1), 8 sampling points within a radius of 1; LBP(8,2), 8 sampling points within a radius of 2; and LBP(8,3), 8 sampling points within a radius of 3.



(a) LBP(8,1) (B) LPB(8.2) (C) LBP(8,3)

Fig. 5. Local Binary Pattern (LBP) variations

The resulting LBP representation were conceptualised in terms of a 2^R dimensional feature vector where each element represented a potential LBP value and the value held within each element corresponded to the number of pixels associated with each LBP value.

As in the case of the histogram representation, an alternative to the LBP representation is to use statistical measures of texture. Again the idea was that such statistical features could be used to augment the LBP representation (or used as a representation on its own). Three categories of texture statistic were identified: (i) entropy features (E), (ii) grey-level occurrence matrix features (M) and (iii) wavelet transform features (W). Table 2 lists the statistical features generated (the letter in parenthesis in each case indicates the category of the feature). Thus on completion of the LBP based representation stage each household is represented using a feature vector of length 266 ($2^R + 10 = 2^8 + 10 = 256 + 10 = 266$).

Table 2. Additional texture based statistical features

#	Description	#	Description	#	Description
1	Entropy (E)	7	Average approximation coefficient matrix, *cA* (W)	10	Average diagonal coefficient matrix, *cD* (W)
2	Average Local Entropy (E)				
3	Contrast (M)	8	Average horizontal coefficient matrix, *cH* (W)		
4	Correlation (M)				
5	Energy (M)	9	Average vertical coefficient matrix, *cV* (W)		
6	Homogeneity (M))				

7 Evaluation

To evaluate and compare the proposed histogram and LBP representations in the context of classifier generation for census data collection the test data introduced in Section 3 was used. A total of 120 records were selected from the two test sites: 70 records from Site A and 50 records from Site B. The labeled household data was separated into three classes: (i) “small family” (less than or equal to 5 people), (ii) “medium family” (between 6 and 8 people inclusive) and (iii) “large family” (more than 8 people). Some statistics concerning the class distributions for the Site A and B data sets are presented in 3.

Table 3. Class label distribution for data set Site A and data set Site B

	small family	medium family	large family	total
Site A	28	32	10	70
Site B	19	21	10	50
total	47	53	20	120

Before classification could commence the input data was first discretised (ranged), a sub-process that served to decrease the size of the feature space (fewer values for each feature/dimension). Feature selection was then applied so as to reduce the number of dimensions (Step 5 in Phase 1 of the proposed framework). In the context of the evaluation presented in this section the Chi-square feature selection strategy was applied. For classifier generation purposes a number of classifier learners were used as implemented in the Waikato Environment for Knowledge Analysis (WEKA) machine learning workbench[3]. For the evaluation purposes Ten fold Cross-Validation (TCV) was applied and

[3] `http://www.cs.waikato.ac.nz/ml/weka/`

the effectiveness of the generated classifiers reported in terms of: (i) accuracy (AC), (ii) area under the ROC curve (AUC), (iii) sensitivity (SN), (iv) specificity (SP) and precision (PR).

Three sets of experiments were conducted directed at:

1. A comparison of the proposed histogram and LPB based representations in the context of classification for census data collection.
2. Identification of the most appropriate number (K) of attributes (features) to retain during feature selection.
3. Determination of the most appropriate classifier generator to use (for the purpose of classification for census data collection).

each set of experiments is discussed in further detail in the following three subsections (Subsections 7.1, 7.2 and 7.3).

7.1 Colour Histograms v. LBPs

To compare the operation of the proposed representations three different variations of the histogram representation were considered together with seven variations of the LBP representation. The histogram representations considered were: (i) colour histogram only (CH), (ii) colour statistics only (CS) and (iii) colour histogram and statistics combined (CH+CS). The LBP representations considered were: (i) LBP(8,1) only, (ii) LBP(8,2) only, (iii) LBP(8.3) only, (iv) texture statistics only (TS), (v) LBP(8,1) and statistics combined (LBP(8,1)+TS), (vi) LBP(8,2) and statistics combined (LBP(8,2)+TS) and (vii) LBP(83) and statistics combined (LBP(8,3)+TS). For the experiments Chi-Square feature selection was used with $K = 35$ (K = 25 was used for the experiments reported in Sub-section 7.2, had revealed that this was the most appropriate value for K) and a Neural Network learning method as these had been found to work well (see Sub-section 7.3). The results are presented in Table 4 (highest values indicated in bold font). Although the results are not conclusive from the Table it can be observed that:

- With respect to the colour histogram based representation best results were obtained using CH for Site A (wet season) and CH+CS for Site B (dry season), the distinction is assumed to occur because of the predominantly green colour of the wet season images in comparison with the predominantly brown colour of the dry season images.
- With respect to the LBP representation best results were produced using LBP(8,1) with respect to both Site A and Site B.
- Comparing the results obtained using both the colour histogram and the LBP representations, LBP(8,1) produced the best overall results.

Thus we conclude that in the context of the test scenario the LBP representation produced the most eftective results.

Table 4. Comparison of different variations of the proposed histogram and LPB based representations in terms of classification performance (neural network classifier and $K = 35$)

Types	Data set	Site A					Site B				
		AC	AUC	PR	SN	SP	AC	AUC	PR	SN	SP
Histogram	CH	**0.614**	**0.754**	**0.615**	**0.614**	**0.761**	0.560	0.753	0.568	0.560	0.723
	CS	0.400	0.550	0.404	0.400	0.629	0.480	0.645	0.480	0.307	0.754
	CH+CS	0.557	0.741	0.551	0.557	0.713	**0.580**	**0.755**	**0.602**	**0.580**	**0.728**
LBP	LBP(8,1)	**0.771**	**0.880**	**0.758**	**0.771**	**0.856**	**0.600**	**0.792**	**0.599**	**0.600**	**0.764**
	LBP(8,2)	0.614	0.765	0.618	0.614	0.725	0.600	0.705	0.600	0.600	0.762
	LBP(8,3)	0.586	0.718	0.583	0.586	0.710	0.600	0.792	0.599	0.600	0.764
	TS	0.414	0.502	0.379	0.414	0.595	0.500	0.650	0.602	0.500	0.754
	LBP(8,1)+TS	0.757	0.848	0.754	0.757	0.847	0.600	0.768	0.601	0.600	0.757
	LBP(8,2)+TS	0.643	0.770	0.645	0.643	0.746	0.580	0.736	0.580	0.580	0.742
	LBP(8,3)+TS	0.557	0.706	0.553	0.557	0.686	0.600	0.768	0.601	0.600	0.757

7.2 Number of Attributes

In order to investigated the effect on classification performance of using different values of K with respect to Chi-Squared feature selection a sequence of experiments were conducted using a range of values for K from 10 to 35 incrementing in steps of 5. For the experiments the colour histogram (CH) and LBP(8,1) representations were used because previous experiments had indicated that these two representations produced the best performance (see above). Neural Network machine learning was again adopted. The results produced are presented in Table 5 (best values obtained are again highlighted in bold). From the Table the following can be observed:

- With respect to the colour histogram based representation best results tended to be obtained using $K = 25$ for Site A (wet season) and $K = 10$ for Site B (dry season), with better results being produced using the Site B data.
- With respect to the LBP representation best results tended to be produced using $K = 35$ with respect to both the Site A and the Site B data.
- The LBP(8,1) representation outperformed the Colour histogram (CH) representation.

Thus it can be concluded that higher K values, such as $K = 35$, produce better results using the LBP representation, while with respect to the histogram representation lower values for K ($K = 10$) produced the best result. Here it should be noted that as K increases the time complexity increases. For example the processing time for LBP(8,1) with respect to the Site A data set using a neural network learning method with $K = 15$, $K = 25$ and $K = 35$ was 6.07, 16.47 and 32.73 seconds respectively.

Table 5. Comparison of different values of K with respect to Chi-Squared feature selection in terms of classification performance (CH and LBP(8,1), and neural network classifier)

Data set	No of attribute	Site A					Site B				
		AC	AUC	PR	SN	SP	AC	AUC	PR	SN	SP
CH	10	0.574	0.678	0.576	0.571	0.691	**0.680**	**0.777**	**0.679**	**0.680**	**0.801**
	15	0.614	0.719	0.608	0.614	0.754	0.580	0.757	0.559	0.580	0.749
	20	0.586	0.727	0.588	0.586	0.734	0.640	0.780	0.645	0.640	0.776
	25	**0.629**	**0.766**	**0.631**	**0.629**	**0.768**	0.620	0.796	0.637	0.620	0.754
	30	0.614	0.776	0.607	0.614	0.751	0.600	0.773	0.608	0.600	0.752
	35	0.614	0.754	0.615	0.614	0.761	0.560	0.753	0.568	0.560	0.723
LBP(8,1)	10	0.657	0.786	0.646	0.657	0.772	0.580	0.677	0.587	0.580	0.745
	15	0.757	0.875	0.752	0.757	0.835	0.580	0.699	0.590	0.580	0.730
	20	0.714	0.874	0.719	0.714	0.835	0.580	0.718	0.593	0.580	0.725
	25	0.757	0.887	0.749	0.757	0.849	0.600	0.713	0.595	0.600	0.754
	30	0.757	0.868	0.743	0.757	0.837	0.580	0.761	0.581	0.580	0.742
	35	**0.771**	**0.880**	**0.758**	**0.771**	**0.856**	**0.600**	**0.792**	**0.599**	**0.600**	**0.764**

7.3 Learning Methods

Eight learning method were considered with respect to the experiments directed at identifying the effect of different learning methods on classification performance including: (i) Decision Tree generators (C4.5), (ii) Naive Bayes, (iii) Averaged One Dependence Estimators (AODE), (iv) Bayesian Network, (v) Radial Basis Function Networks (RBF Networks), (vi) Logistic Regression, (vii) Sequential Minimal Optimisation (SMO) and (viii) Neural Networks Back-propagation (in WEKA this is referred to as a MultilayerPerceptron). The Colour Histogram (CH) and LBP(8,1) representations were again used. $K = 10$ Chi-Squared feature selection was used with the Colour Histogram (CH) representation and $K = 35$ for the LBP(8,1) representation. The obtained results are presented in Table 6. From the Table it can be observed that:

- With respect to the colour histogram based representation best results were obtained using the Bayes Network learner with respect to the Site A data, and AODE and Bayes Network with respect to the Site B data.
- With respect to the LBP based representation best results were obtained using using Neural Networks with respect to the Site A data, and Logistic Regression with respect to the Site B data.
- The C4.5, Naive Bayes, RBF Network and SMO learners did not perform well.
- Overall the Neural Network learner, combined with the LBP(8,1) representation and $K = 35$ Chi-Squared feature selection, produced the best overall result.

Thus in conclusion a number of different machine learners produced good results, different machine learners tended to be more compatible with different representations, but overall Neural Network learning produced the best result.

Table 6. Comparison of different classifier generators in terms of classification performance (CH with $K = 10$ and LBP(8,1) with $K = 35$)

Data set	Learning method	Site A					Site B				
		AC	AUC	PR	SN	SP	AC	AUC	PR	SN	SP
CH + 35 atts	C4.5	0.557	0.640	0.573	0.557	0.677	0.480	0.618	0.450	0.480	0.681
	Naive Bayes	0.629	0.731	0.635	0.629	0.734	0.640	0.780	0.646	0.640	0.797
	AODE	0.586	0.709	0.605	0.586	0.686	**0.700**	0.706	**0.702**	**0.700**	0.813
	Bayes Network	**0.629**	**0.751**	**0.630**	**0.629**	**0.736**	0.680	**0.794**	0.684	0.680	**0.815**
	RBF Network	0.571	0.692	0.576	0.571	0.720	0.440	0.649	0.449	0.440	0.700
	Logistic Regression	0.486	0.637	0.486	0.486	0.670	0.580	0.700	0.560	0.580	0.744
	SMO	0.457	0.542	0.481	0.457	0.598	0.600	0.710	0.589	0.600	0.768
	Neural Network	0.574	0.678	0.576	0.5710	.691	0.680	0.777	0.679	0.680	0.801
LBP(8,1) + 35 atts	C4.5	0.614	0.718	0.619	0.614	0.773	0.500	0.657	0.495	0.500	0.708
	Naive Bayes	0.543	0.709	0.583	0.543	0.763	0.540	0.762	0.594	0.540	0.796
	AODE	0.557	0.755	0.529	0.557	0.698	0.600	0.782	0.596	0.600	0.766
	Bayes Network	0.571	0.729	0.599	0.571	0.768	0.520	0.767	0.554	0.520	0.772
	RBF Network	0.657	0.740	0.665	0.657	0.818	0.600	0.743	0.601	0.600	0.783
	Logistic Regression	0.757	0.857	0.757	0.757	0.866	**0.757**	**0.857**	**0.758**	**0.757**	**0.866**
	SMO	0.729	0.789	0.718	0.729	0.823	0.729	0.789	0.718	0.729	0.823
	Neural Network	**0.771**	**0.880**	**0.758**	**0.771**	**0.856**	0.600	0.792	0.599	0.600	0.764

8 Conclusion

A data mining mechanism for generating census data (household size) from satellite imagery has been proposed. More specifically a three phase framework has been suggested that takes large scale satellite imagery as input and produces an evaluated classifier for household size census collection. The key element with respect to the process is the way in which individual households are represented so that an effective classifier can be generated. Two basic representations were proposed: colour histograms and LBPs. Ten variations of these representations were considered. Experiments were conducted using a training set obtained from a rural part of the Ethiopian hinterland. This was selected because of the availability of on-the-ground data and because the proposed process is intended for use in rural areas, particularly rural areas with poor communication and transport infrastructures that tend to exacerbated the issues associated with traditional forms of census collection. Experiments were also conducted to identify a best K value with respect to Chi-Squared feature selection and a number of classifier generators. The main findings were: (i) that it is possible to collect household size census data using the proposed approach to a reasonable level of accuracy (a best ROC value of 0.880 was obtained), (ii) that there was a performance distinction between the “green” wet season data (Site A) and the “brown” dry season data (Site B), (iii) that the LBP(8,1) representation tended to produce the best results, (iv) the most desirable value for K depended on the nature of the representation adopted (high with respect to the LBP representation, lower with respect to the histogram representation), and (v) that a number of machine learners performed well but the use of neural networks provided the best results. For future work the research team intend to investigate representations based on quad tree decompositions and better segmentation techniques to generate individual house hold images.

References

1. Abdelfattah, R., Nicolas, J.M.: Interferometric synthetic aperture radar coherence histogram analysis for land cover classification. In: Proc. Information and Communication Technologies (ICTTA 2006), vol. 1, pp. 343–348. IEEE (2006)
2. Atkinson, P.M.: Super-resolution land cover classification using the two-point histogram. In: Sanchez-Vila, X., Carrera, J., Gmez-Hernndez, J. (eds.) Proc. Geostatistics for Environmental Applications (geoENV VI), vol. 13, pp. 15–28. Springer, Netherlands (2004)
3. Beumier, C., Idrissa, M.: Building change detection by histogram classification. In: Proc. Signal-Image Technology and Internet-Based Systems (SITIS 2011), pp. 409–415. IEEE (2011)
4. Chakravarti, R., Meng, X.: A study of color histogram based image retrieval. In: Proc. Information Technology: New Generations (ITNG 2009), pp. 1323–1328. IEEE (2009)
5. Cheng, L., Zhou, Y., Wang, L., Wang, S., Du., C.: An estimate of the city population in china using dmsp night-time satellite imagery. In: Proc. IEEE International on Geoscience and Remote Sensing Symposium (IGARSS 2007), pp. 691–694. IEEE (2007)
6. Demirel, H., Anbarjafari, G.: Satellite image resolution enhancement using complex wavelet transform. IEEE Geoscience and Remote Sensing Letters 7(1), 123–126 (2010)
7. Dittakan, K., Coenen, F., Christley, R.: Towards the collection of census data from satellite imagery using data mining: A study with respect to the ethiopian hinterland. In: Bramer, M., Petridis, M. (eds.) Proc. Research and Development in Intelligent Systems XXIX, pp. 405–418. Springer, London (2012)
8. Fanoe, J.A.X.: Lessons from census taking in south africa: Budgeting and accounting experiences. The African Statistical 13(3), 82–109 (2011)
9. Gonzalez, R.C., Woods, R.E.: Digital Image Processing (3rd Edition), 3rd edn. Pearson Prentice Hall (2007)
10. Hadid, A.: The local binary pattern approach and its applications to face analysis. In: Proc. First Workshop on Image Processing Theory, Tools and Applications (IPTA 2008), pp. 1–9. IEEE (2008)
11. Hijazi, M.H.A., Coenen, F., Zheng, Y.: Retinal image classification using a histogram based approach. In: Proc. International Joint Conference on Neural Network (Special Session on Soft Computing in Medical Imaging), part of IEEE World Congress on Computational Intelligence (WCCI 2010), pp. 3501–3507 (2010)
12. Jang, J.H., Kim, S.D., Ra, J.B.: Enhancement of optical remote sensing images by subband-decomposed multiscale retinex with hybrid intensity transfer function. IEEE Geoscience and Remote Sensing Letters 8(5), 983–987 (2011)
13. Jeong, S., Won, C.S., Gray, R.M.: Image retrieval using color histograms generated by gauss mixture vector quantization. Computer Vision and Image Understanding 94(1-3), 44–66 (2004)
14. Kita, Y.: A study of change detection from satellite images using joint intensity histogram. In: Proc. International Conference on Pattern Recognition (ICPR 2008), pp. 1–4. IEEE (2008)
15. Li, G., Weng, Q.: Using landsat etm+ imagery to measure population density in indianapolis, indiana, usa. Photogrammetric Engineering and Remote Sensing 71(8), 947 (2005)
16. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)

17. Pietikäinen, M.: Image analysis with local binary patterns. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 115–118. Springer, Heidelberg (2005)
18. Song, C., Li, P., Yang, F.: Multivariate texture measured by local binary pattern for multispectral image classification. In: Proc. IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS 2006), pp. 2145–2148 (2006)
19. Vasconcelos, N., Lippman, A.: Feature representations for image retrieval: beyond the color histogram. In: Proc. IEEE International Conference on Multimedia and Expo (ICME 2000), vol. 2, pp. 899–902. IEEE (2000)
20. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI 2007) 29(6), 915–928 (2007)
21. Zhao, G., Wu, G., Liu, Y., Chen, J.: Texture classification based on completed modeling of local binary pattern. In: Proc. International Conference on Computational and Information Sciences (ICCIS 2011), pp. 268–271. IEEE (2011)

Density Ratio Estimation in Support Vector Machine for Better Generalization: Study on Direct Marketing Prediction

Muhammad Syafiq Mohd Pozi, Aida Mustapha, and Anas Daud

Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia, Serdang, 43400 Selangor, Malaysia
{msyafiqpozi,anas.daud}@gmail.com, aida@fsktm.upm.edu.my
http://fsktm.upm.edu.my

Abstract. In this paper we show how to improve the generalization performance of Support Vector Machine (SVM) by incorporating density ratio based on Unconstrained Least Square Importance Fitting (uLSIF) into the SVM classifier. ULSIF function is known to have optimal non-parametric convergence rate with optimal numerical stability and higher robustness. The ULSIF-SVM classifier is validated using marketing dataset and achieved better generalization performance as compared against classic implementation of SVM.

Keywords: Density ratio estimation, support vector machine, unconstrained least square importance fitting.

1 Introduction

Classic machine learning algorithms assume that probability distribution of testing data is the same as the probability distribution of training data [4]. As the result, life span of new machine learning model, especially in classification is short. This is because most of real world data are non-stationary, whereby they keep changing over time [8]. Therefore, instead of performing common model selection such as cross validation [1, 2], estimating ratio between two sample distribution and use it inside the learning model is preferable [3]. Density ratio or importance weight estimation is a technique to estimate weight from two different samples (training and testing) such that:

$$\beta(x) = \frac{p_{test(x)}}{p_{train}(x)}$$

There are several methods to estimate importance weight. The naive way is to estimate the probability density function for both samples. Histogram estimation or kernel density estimation can be used. However, these methods suffer from the curse of dimensionality when input dimension increases. The best method, especially in machine learning, is to directly estimate the importance weight without estimating the probability density function of both samples. We selected

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 275–280, 2013.

Unconstrained Least square Importance Fitting (ULSIF) [4] for this research. Several researches have shown good classification result when using importance weights estimated from ULSIF for importance weight estimation [5, 9, 10].

In this paper, we incorporate the density ratio into SVM formulation and attempt to improve the generalization of SVM classifier model on a highly imbalanced data [7] without sacrificing classification accuracy. In section 2, we briefly explain about SVM and how to incorporate importance weight into the SVM. Section 3 will discuss what is ULSIF. In section 4, we will detail out how we perform the experiment and analyze the results. Finally, section 5 will have our conclusion and future research suggestion.

2 Support Vector Machine

Support vector machine [6] is used to build a classification model from a training data given a set of instance-label pairs $(x_i, y_i), i = 1, ..., l$ where $x_i \in R^n$ and $y \in \{1, -1\}^l$, the support vector machines (SVM) requires the solution of following optimization problem:

$$\begin{aligned} \underset{w,b,\xi}{min} \quad & \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(w^{\mathbf{T}}\phi(x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{1}$$

where C is a regularization parameter and $C \geq 0$. Instead of solving the primal problem, we convert it into Lagrangian dual problem:

$$\begin{aligned} \underset{\alpha}{max} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i}^{l}\sum_{j}^{l}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C \end{aligned} \tag{2}$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel to map feature space to high dimensional space. Given a weight vector $v = \{\beta(x_i)|i_1^n\}$ from density ratio estimation of two samples, equation (1) become

$$\begin{aligned} \underset{w,b,\xi}{min} \quad & \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{l}v_i\xi_i \\ \text{subject to} \quad & y_i(w^{\mathbf{T}}\phi(x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{3}$$

The Lagrangian dual problem is:

$$\begin{aligned}
\underset{\alpha}{max} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i}^{l}\sum_{j}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\
\text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& 0 \leq \alpha_i \leq C v_i
\end{aligned} \tag{4}$$

3 Unconstrained Least Square Importance Fitting

Unconstrained Least Square Importance Fitting (ULSIF) is an approximation method based on Least Square Importance Fitting (LSIF) [4]. LSIF will estimate density ratio $w(x)$ between two samples by the following linear model:

$$\widehat{w}(x) = \sum_{\ell=1}^{b} \alpha_\ell \varphi_\ell(x), \tag{5}$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_b)^\top$ are parameters to be learned from data samples, $\top$ denotes the transpose of a matrix or a vector and $\{\varphi_\ell(x)\}_{\ell=1}^{b}$ are basis function such that

$$\varphi_\ell(x) \;\; \geq 0 \text{ for all } x \in \mathcal{D} \text{ and for } \ell = 1, 2, ..., b \tag{6}$$

The parameter $\{\alpha_\ell\}_{\ell=1}^{b}$ is determined by this optimization formula as follows

$$\begin{aligned}
\underset{\alpha \in \mathbb{R}^b}{min} \quad & \left[\frac{1}{2}\alpha^\top \widehat{H}\alpha - \widehat{h}^\top \alpha + \lambda 1_b^\top \alpha\right] \\
\text{subject to} \quad & \alpha \geq 0_b,
\end{aligned} \tag{7}$$

where 0_b and 1_b are the b-dimensional vectors with all zeros and ones, respectively. However, the author shows that LSIF tends to suffer from numerical problems which make it not reliable. The author then propose a method to approximate the density ratio by replacing $1_b^\top \alpha$ in (7) with $\alpha^\top \alpha / 2$ which results in new optimization problem

$$\min_{\alpha \in \mathbb{R}^b} \quad \left[\frac{1}{2}\alpha^\top \widehat{H}\alpha - \widehat{h}^\top \alpha + \frac{\lambda}{2}\alpha^\top \alpha\right]. \tag{8}$$

4 Experiment Results

The sample is collected from [7] has 16 features and is divided to 3 categories, which are bank client data, last contact of the current campaign and other additional features. There are total of 45211 instances in the dataset. Our objective is to predict if the client will sign a long term deposit or not based on these features.

4.1 Data Analysis and Preprocessing

The dataset is highly imbalanced with 5289 (11.70%) signed a long term deposit and 39922 (88.30%) did otherwise. Because the data consist both numerical and non-numerical, we transformed all non-numerical data into numerical and normalized all the features so that it ranges between 0 and 1. Next, we generated 3 experimental sets consisting of training and testing data from the original dataset. Both training and testing data are unique, and the probability distribution for training and testing data are similar to the original sample.

Set 1 : 60% (27127) training data and 40% (18084) testing data.
Set 2 : 50% (22606) training data and 50% (22605) testing data.
Set 3 : 40% (18084) training data and 60% (27127) testing data.

For every set, we split the training data into two set (Set A and Set B). Again, each set has similar probability distribution with the original sample. Table 1 shows the details of each dataset.

Table 1. Total Set A and Set B data for each experiment set from Set 1, Set 2 and Set 3

	Set A	Set B
Set 1	13563	13564
Set 2	11303	11303
Set 3	9042	9042

4.2 Result

We used LIBSVM to build classification model and C++ implementation of ULSIF to estimate the importance weight. Our experiment consist of two parts. In the first part, we built the SVM classification model using training data from each experimental set. Table 2 show the performance of the classification model.

In the second part, we extended the experiment by building two types of SVM classification model. For the first classification model, we built a classifier from set A for each experimental set. For second classification model, we estimated the importance weight of set A based on set B for every experimental set. Next, we built the classifier from set A by incorporating the importance weight. Table 3 shows the result of our experiments. From table 3, we can see that although the classification accuracy does not change, the total number of support vectors are lower. This means the weighted approach is able to improve generalization performance of the SVM classifier.

Table 2. Performance of SVM on training data

	Accuracy	**Total Support Vector**
Set 1	88.6474%	6395
Set 2	88.9759%	5327
Set 3	88.3400%	4259

Table 3. Performance of normal and weighted SVM

	Accuracy		**Support Vector**	
	Normal	Weighted	Normal	Weighted
Set 1	88.3046%	88.3046%	3211	2769
Set 2	88.3035%	88.3035%	2680	2513
Set 3	88.3142%	88.3032%	2151	1761

5 Conclusion and Future Research

In this paper, we have demonstrated that by incorporating importance weight to the classifier, the number of support vectors are reduced which increase the classifier generalization compared to normal SVM. Future research will be on (i) using and comparing other density ratio estimation algorithm to get importance weight, (ii) finding on how to improve the accuracy of the classifier and (iii) finding the correlation between the classification accuracy and the degree of difference in probability density function of both training and testing samples.

Acknowledgment. This project is funded by the Malaysian Ministry of Higher Education (MOHE) under the Exploratory Research Grant Scheme (ERGS).

References

[1] Stone, M.: Cross-validatory choice and assessment of statistical predictions, pp. 111–147 (1974)

[2] Wahba, G.: Society for industrial and applied mathematics, vol. 59 (1990)

[3] Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference 90(2), 227–244 (2000)

[4] Kanamori, T., Hido, S., Sugiyama, M.: A Least-squares Approach to Direct Importance Estimation. J. Mach. Learn. Res. 10, 1391–1445 (2009), `http://portal.acm.org/citation.cfm?id=1755831`

[5] Li, Y., Kambara, H., Koike, Y., Sugiyama, M.: Application of covariate shift adaptation techniques in brain–computer interfaces. IEEE Transactions on Biomedical Engineering 57(6), 1318–1324 (2010)

[6] Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), http://www.csie.ntu.edu.tw/~cjlin/libsvm

[7] Moro, S. and Laureano, R. and Cortez, P.: Proceedings of the European Simulation and Modelling Conference, ESM 2011. pp. 117–121. EUROSIS, Guimaraes, Portugal (October 2011)

[8] Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition 45(1), 521–530 (2012), http://dx.doi.org/10.1016/j.patcog.2011.06.019

[9] Sugiyama, M.: Learning under non-stationarity: Covariate shift adaptation by importance weighting, pp. 927–952. Springer (2012)

[10] Karasuyama, M., Harada, N., Sugiyama, M., Takeuchi, I.: Multi-parametric solution-path algorithm for instance-weighted support vector machines. Machine Learning 88(3), 297–330 (2012)

Pacc - A Discriminative and Accuracy Correlated Measure for Assessment of Classification Results

Madhav Sigdel and Ramazan S. Aygün

Department of Computer Science,
University of Alabama in Huntsville, Huntsville

Abstract. Measuring the performance of a classifier properly is important to determine which classifier to use for an application domain. The comparison is not straightforward since different experiments may use different datasets, different class categories, and different data distribution, thus biasing the results. Many performance (correctness) measures have been described to facilitate the comparison of classification results. In this paper, we provide an overview of the performance measures for multiclass classification, and list the qualities expected in a good performance measure. We introduce a novel measure, *probabilistic accuracy (Pacc)*, to compare multiclass classification results and make a comparative study of several measures and our proposed method based on different confusion matrices. Experimental results show that our proposed method is discriminative and highly correlated with accuracy compared to other measures. The web version of the software is available at http://sprite.cs.uah.edu/perf/.

1 Introduction

There are a number of factors that affect the performance of a classification problem: the classification algorithm, the features, the datasets, and the data distribution. The performance of a classification methodology should be compared and analyzed to select a particular method based on the usefulness of the classifier. Result of a classification problem is often represented in the form of a matrix called the confusion matrix. Thus, the performance of two classifiers can be evaluated by comparing the corresponding confusion matrices.

The most common method for the evaluation of classification results is to compute a performance metric based on the confusion matrix. Several such measures have been described in the literature. Most of these measures have been developed for binary classification. Accuracy is one of the widely used measures which is the percentage of correct decisions made by the classifier. However, the overall accuracy is not a very reliable measure for problems such as protein crystallization classification [1] where the cost of misclassifying crystals as non-crystals is very high or the proportion of data in the different categories is significantly different. There are also other measures like sensitivity, specificity,

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 281–295, 2013.

precision and F-measure are formulated for binary classification. Some research [2], [3] describes methods to extend F-measure for multiclass classification. Research studies [4], [5], and [6] propose extensions to area under the ROC curve (AUC) for multiclass result evaluation. There are also other measures like confusion entropy [7] and K-category correlation coefficient [8] that are naturally applicable for the performance evaluation of multiclass classification results.

Analysis based on multiple performance measures is also another popular method for evaluation of classifiers. For example, precision and recall are often analyzed together. For using multiple measures, a problem is that when comparing classifier A and classifier B, classifier A may outperform classifier B with respect to one measure, while classifier B may outperform classifier A with the other measure.

The advantages and disadvantages of the widely used performance measures like accuracy, precision, recall, correlation coefficient, relative entropy, etc. are analyzed in [9] and [10]. Sokolova et al. mention that different performance measures possess invariance properties with respect to the change in a confusion matrix and these properties can be beneficial or adverse depending on the problem domain and objectives [9]. Statistical techniques for comparison of classifiers over multiple datasets are described in [11] and [12]. Perner [13] describe a methodology for interpreting results from decision trees. Though there are research studies on measures for classification results, a comparative study of these measures with classification results for binary and multiclass classification have not been explored much.

This paper focuses on the analysis of performance measures for multiclass classification. We try to analyze the consistency between different measures and also the degree of discrimination for confusion matrix comparison. We propose a new measure, *probabilistic accuracy (Pacc)*, which is based on the difference in probability of correct classification and probability of misclassification given a confusion matrix. Accuracy measure is still one of the widely used measures despite its limitations. A major reason for this is that the accuracy measure has simple semantic correspondence to our understanding. For some other measures, the applicable range is different and it is hard to derive a semantic meaning from those measures. Therefore, we develop our measure in a way that it is consistent with accuracy but more discriminative than accuracy. Besides it is defined for every type of confusion matrix (binary or multiclass) with valid values and is less susceptible to scaling of the number of items in a class. The web-version of the software is available online at http://sprite.cs.uah.edu/perf/ which allows computation of *Pacc* measure along with other popular performance measures for evaluation of classification results.

The rest of this paper is organized as follows. Section 2 provides an overview of several multiclass classification performance measures. Section 3 discusses about the qualities expected in a good performance measure for classification results comparison. Section 4 provides the formal definition and semantics of *Pacc* measure. Section 5 provides a comparative study of several performance measures and our proposed measure considering several cases of confusion matrices. Section 6 concludes the paper.

2 Multiclass Performance Measures

The result of an N-class classification experiment with classes 0..N-1 can be visualized in matrix of size N x N. This matrix is called confusion matrix, contingency matrix, or contingency table. Matrix C represents a generalized N x N confusion matrix.

$$C = \begin{pmatrix} C_{00} & C_{01} & \ddots & C_{0(N-1)} \\ C_{10} & C_{11} & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots \\ C_{(N-1)0} & C_{(N-1)1} & \ddots & C_{(N-1)(N-1)} \end{pmatrix}$$

The value C_{ij} refers to the number of items of i^{th} class classified as j^{th} class where i represents the actual class and j represents the predicted class. The elements in the diagonals represent the number of items of each class correctly classified while the rest constitute misclassification. Brief discussion and definition of the performance measures for multiclass classification based on this generalized confusion matrix is presented below.

2.1 Confusion Entropy

Wei et al. introduce confusion entropy method as a performance measure for multiclass classification [7]. The authors apply the concept of probability and information theory for the calculation of confusion entropy. The misclassication probability of classifying samples of class i to class j subject to class j is denoted by P^j_{ij} and is given by (1). Similarly, P^i_{ij} is the misclassication probability of classifying samples of class i to class j subject to class i.

$$P^j_{ij} = \frac{C_{ij}}{\sum\limits_{k=0}^{N-1} C_{jk} + C_{kj}} \qquad i \neq j,\, i, j = 0..N-1 \tag{1}$$

Note that P^i_{ii}=0. Confusion entropy of class j is defined as in (2).

$$CEN_j = -\sum_{k=0,k\neq j}^{N-1} P^j_{jk} log_{2(N-1)} P^j_{jk} + P^j_{kj} log_{2(N-1)} P^j_{kj} \tag{2}$$

Overall entropy (CEN) is defined by (3).

$$CEN = \sum_{j=0}^{N-1} P_j CEN_j \tag{3}$$

where P_j is defined as in (4):

$$P_j = \frac{\sum_{k=0}^{N-1} C_{jk} + C_{kj}}{2\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} C_{kl}} \tag{4}$$

The value of CEN ranges from 0 to 1 with 0 signifying the best classification and 1 indicating the worst classification.

2.2 K-Category Correlation Coefficient

Gorodkin proposes K-category correlation coefficient to compare two confusion matrices [8]. The method utilizes the concept of covariance and tries to compute the covariance between actual K-category assignment and the observed assignment. Consider two matrices X, Y of size N x K where N is the number of items and K is the number of categories. Let matrix X and matrix Y represent the actual assignment and predicted assignment, respectively. The correlation coefficient, R_k is defined as (5).

$$R_k = \frac{cov(X,Y)}{\sqrt{cov(X,X)}\sqrt{cov(Y,Y)}} \tag{5}$$

In terms of confusion matrix as denoted in the beginning of this section, the covariances can be written as follows:

$$cov(X,Y) = \sum_{k,l,m=0}^{N-1} C_{kk}C_{ml} - C_{lk}C_{km} \tag{6}$$

$$cov(X,X) = \sqrt{\sum_{k=0}^{N-1}\left(\sum_{l=0}^{N-1} C_{lk}\right)\left(\sum_{f,g=0,f\neq k}^{N-1} C_{gf}\right)} \tag{7}$$

$$cov(Y,Y) = \sqrt{\sum_{k=0}^{N-1}\left(\sum_{l=0}^{N-1} C_{kl}\right)\left(\sum_{f,g=0,f\neq k}^{N-1} C_{fg}\right)} \tag{8}$$

The value of R_k ranges from -1 to +1 with +1 indicating the best classification and -1 indicating the worst classification.

2.3 F-Measure for Multiclass Problems

F-measure combines the two metrics - recall and precision and is defined as the harmonic mean of the two. As described in [3], the recall (R_i), precision (P_i), and F-measure (F_i) for class i in a multiclass problem can be defined by following equations:

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i}, \tag{9}$$

$$(F_i) = \frac{2P_iR_i}{P_i + R_i} \tag{10}$$

where TP_i is the number of objects from class i assigned correctly to class i, FP_i is the number of objects that do not belong to class i but are assigned to class i, and FN_i is the number of objects from class i predicted to another class. To compute the overall F-measure, macro-averaging and micro-averaging are used. Macro-averaged F-measure, F(macro), is calculated as the average of F-measure for each category. Micro-averaged F-measure, F(micro), aggregates the recall and precision of classes.

$$F(macro) = \frac{1}{N}\sum_{i=0}^{N-1} F_i, \quad F(micro) = \frac{2PR}{P + R} \tag{11}$$

where P and R are defined by the following equations:

$$P = \frac{\sum\limits_{i=0}^{N-1} TP_i}{\sum\limits_{i=0}^{N-1} TP_i + FP_i}, \quad R = \frac{\sum\limits_{i=0}^{N-1} TP_i}{\sum\limits_{i=0}^{N-1} TP_i + FN_i} \tag{12}$$

2.4 Kappa Statistic

Kappa statistic is defined as the proportion of agreement between two rankings corrected for chance [14]. In the context of classification result, the agreement between the actual categories and predicted categories forms the basis for calculation of Kappa. Let $S = \sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{N-1} C_{ij}$ represent the total number of items in the confusion matrix, $C_{i.} = \sum\limits_{j=0}^{N-1} C_{ij}$ represent the i^{th} row marginal and $C_{.i} = \sum\limits_{j=0}^{N-1} C_{ji}$ represent the i^{th} column marginal. Then, Cohen's Kappa (K) is given by (13).

$$K = \frac{P_o - P_e}{1 - P_e} \tag{13}$$

where $P_o = \frac{1}{S}\sum\limits_{i=0}^{N-1} C_{ii}$ is the proportion of agreement between observed and actual categories, and $P_e = \frac{1}{S^2}\sum\limits_{i=0}^{N-1} C_{i.}C_{.i}$ is the proportion of observations for which agreement is expected by chance. $P_o - P_e$ is the proportion of agreement beyond what is expected by chance, and $1 - P_e$ is the maximum possible proportion of agreement beyond what is expected by chance. Values of Kappa can range from -1 to +1, with -1 indicating perfect disagreement below chance, and +1 indicating perfect agreement above chance.

3 Qualities of Good Performance Measure

Consider the following confusion matrices.

$$M = \begin{pmatrix} 70 & 10 \\ 10 & 10 \end{pmatrix} \quad N = \begin{pmatrix} 80 & 0 \\ 20 & 0 \end{pmatrix} \quad K = \begin{pmatrix} 20 & 0 \\ 20 & 10 \end{pmatrix} \quad L = \begin{pmatrix} 100 & 0 \\ 20 & 10 \end{pmatrix}$$

Matrix M has 10 items of each class misclassified and Matrix N has all items of class 1 misclassified while all items in class 0 are correctly classified. The accuracy for both matrices is 80%. However, the classifier that results N might not be useful since all items have been classified to a single class. This is because the distribution of misclassification is ignored by the measure. Consider other hypothetical classification results given by matrix K and L. The first category has all items correctly classified while 20 of 30 objects are misclassified for the second category. Suppose the data items of the first category are increased by 5 folds and the new confusion matrix is given by matrix L. This could be the case where it is easy to classify the objects of the first category. The accuracy of the experiment is increased from 60% to 84% just by increasing the items in the first category. This result can be misleading if the judgement is based on accuracy measure.

In this section, we list the qualities expected in a good performance measure for the evaluation of classification results. We first list the desired qualities in a good performance metric irrespective of the problem domain. These are listed as follows:

- The measure should have the highest value for the best case i.e., when all items correctly classified. There can be many varieties of the worst case depending on the distribution of misclassification. We may want to distinguish those cases.
- The measure should not be affected by a scale factor. This means that the measure for matrix C should be same as the measure for a x C where a is a scale factor.
- The measure should be based on all the values in the confusion matrix for the calculation.
- The measure should be able to distinguish different confusion matrices. The value should decrease with increase in misclassified cases and increase with decrease in misclassified cases or vice versa.
- It should be useful for classification with any number of classes.

Likewise, there are some other qualities which may or may not be desired depending on the application. These are listed as follows:

- If the important class occurs very rarely, performance measures are affected by the scaling of data. Thus we desire that a performance measure should be less affected by the scaling of one or more of the classes as long as the distribution of misclassification is proportional. However, in some cases, we may want to pay extra attention to a class which is more likely than others.

- Misclassification into a single class may be considered better than misclassification into several classes. If the misclassification occurs into a single class, the classifier may be tuned by focusing on the problematic classes.

4 Proposed Pacc Measure

We introduce *probabilistic accuracy (Pacc)* measure for comparison of two N-class confusion matrices. This measure is based on the difference in the probability of correct classification and the probability of misclassification. C_{ij} refers to the number of items in class i that are classified to class j. The occurrence (probability) of C_{ij} is related to both the number of items in class i and the number of items of other classes that are classified to class j. P_{ij} is the probability of occurrence of C_{ij} subject to actual class i and observed class j and is defined as in (14). Apparently, C_{ij} is contained in both sub-parts of the denominator. If $i \neq j$, P_{ij} is the probability of misclassifying item of class i subject to class j. P_{ij} should increase if the majority of incorrect classifications into class j are coming from items in class i. Note that the numerator does not contain the correctly classified cases. Likewise, the probability of correctly classifying items, denoted by P_{ii}, can be defined as (15). Here the numerator consists only the correctly classified cases i.e., the diagonal elements of the confusion matrix.

$$P_{ij} = \frac{2C_{ij}}{\sum\limits_{k=0}^{N-1} C_{ik} + C_{kj}} \quad i \neq j, k = 0, ..N-1 \tag{14}$$

$$P_{ii} = \frac{2C_{ii}}{\sum\limits_{k=0}^{N-1} C_{ik} + C_{ki}} \tag{15}$$

Maximum value for any P_{ij} is 1. This occurs when all items of class i are classified solely to class j and none of the items from other classes are classified to class j. In other words, this is like pure misclassification probability between class i and j which indicates that all items of class i are classified as class j and all observed class j classifications result from class i items. The minimum value is 0 which occurs when all items in a class are correctly classified and no items of other classes is predicted to this class.

Now we define two terms: error (ϵ) and correctness (c) in terms of P_{ii} and P_{ij} as given in (16) and (17), respectively. Error probability (ϵ) is the average of the probabilities of misclassification and correctness probability (c) is the average of the probabilities for correct classification. c and ϵ lie between 0 and 1. High correctness probability and low error probability are desired for good classification. The difference between c and ϵ yields a value between -1 to +1. It is normalized to the range 0 to 1 as in (18) so that it can be correlated with the accuracy measure.

Table 1. Classification results for 2-class problem

	O_0	O_1	O_0	O_1	O_0	O_1	O_0	O_1	O_0	O_1
	A		B		C		D		E	
C_0	50	0	25	25	50	0	10	40	0	50
C_1	0	50	25	25	50	0	40	10	50	0
	F		G		H		I		J	
C_0	80	0	70	10	80	0	40	40	0	80
C_1	0	20	10	10	20	0	10	10	20	0

$$\epsilon = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0,i\neq j}^{N-1} P_{ij} \tag{16}$$

$$c = \frac{1}{N}\sum_{i=0}^{N-1} P_{ii} \tag{17}$$

$$Pacc = \frac{1}{2} + \frac{c-\epsilon}{2} \tag{18}$$

The value of *Pacc* is maximum (i.e., 1) when all the items are correctly classified. In this case, ϵ is 0 because every C_{ij} where i $\neq$ j is 0. Likewise, c is equal to 1. Hence, the difference between c and ϵ is 1 and the normalization to [0-1] gives the value 1.

$$\epsilon = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0,i\neq j}^{N-1} P_{ij} = 0$$

$$c = \frac{1}{N}\sum_{i=0}^{N-1} P_{ii} = \frac{1}{N}\sum_{i=0}^{N-1} \frac{2C_{ii}}{C_{ii}+C_{ii}} = \frac{1}{N}\sum_{i=0}^{N-1} 1 = 1$$

$$Pacc = \frac{1}{2} + \frac{c-\epsilon}{2} = \frac{1}{2} + \frac{1-0}{2} = 1$$

The value of *Pacc* is minimum (i.e., 0) when every items from a class are misclassified to a unique single class.

5 A Comparative Study of Performance Measures

In this section, we perform a comparative study of the following performance measures: accuracy (ACC), Kappa statistic (KAPPA), K-category correlation coefficient (R_k), confusion entropy (CEN), macro-averaged F-measure (FMEAS) and our *Pacc measure* for several confusion matrices. The measure for CEN is subtracted from 1 to simplify the comparison (for this measure the lowest value (i.e., 0) is the best and the highest value (i.e., 1) is the worst).

Table 2. Performance measures for matrices A to J in Table 1

MATRIX	ACC	KAPPA	R_k	1-CEN	FMEAS	PACC
A	1.00	1.00	1.00	1.00	1.00	1.00
B	0.50	0.00	0.00	0.00	0.50	0.50
C	0.50	0.00	NAN	0.60	NAN	0.50
D	0.20	-0.60	-0.60	-0.06	0.20	0.20
E	0.00	-1.00	-1.00	0.00	NAN	0.00
F	1.00	1.00	1.00	1.00	1.00	1.00
G	0.80	0.37	0.38	0.40	0.69	0.74
H	0.80	0.00	NAN	0.68	NAN	0.64
I	0.50	0.00	0.00	0.17	0.45	0.50
J	0.00	-0.47	-1.00	0.28	NAN	0.00

5.1 Analysis of Measures for 2-class Classification

Consider the classification results for the 2-class problems as given by confusion matrices A to J in Table 1. The matrices follow the generalized confusion matrix structure outlined in Section 2. The columns O_i indicate the objects classified to class i and the rows C_i indicate the actual categories. Table 2 shows the performance measures for the matrices in Table 1.

Accuracy does not account for the distribution of misclassified items. As long as the numbers of correct predictions remain the same, accuracy remains the same. In matrix G, the accuracy is 80% where half of the items in class 1 are misclassified to class 0. Similarly, in matrix H, the accuracy is still 80% where all items of one class have been classified to other class. Thus analysis based on accuracy measure can be misleading.

Kappa statistic is less discriminative. Also, Kappa doesn't take the least value (= -1) for matrix J where none of the items are correctly classified. The value NaN in the R_k columns corresponding to matrix C and matrix H indicates that it is not a number (NaN). Another observation is that Kappa and R_k measures are less correlated with accuracy measure.

For matrix B in Table 2, the value of (1-CEN) is 0, signifying the worst classification. However, Matrix B has half the items in each class correctly classified. Moreover, the value of CEN goes out of range in some cases. For matrix D, CEN value is 1.06 which is out of the range. Such cases arise when the ratio of correct cases to incorrect cases is less than 1 for both the categories. Among the matrices G, H, I, and J, we would expect the measure to indicate G as the best classification and J as the worst classification. The CEN values indicate matrix H to be the best among the four and matrix I to be the worst. Definitely, the performance measure for J should have the worst value as it has all the items misclassified. This shows that CEN is not a reliable measure.

There are several cases where the F-measure is undefined. Such cases arise when there are some categories for which no correct classification is made. Moreover, we can observe that NaN does not necessarily occur when the confusion

Table 3. Classification results for 3-class problem

	O_0	O_1	O_2	O_0	O_1	O_2	O_0	O_1	O_2	O_0	O_1	O_2
	A			B			C			D		
C_0	60	0	0	40	0	20	30	30	0	30	15	15
C_1	0	60	0	0	60	0	0	60	0	0	60	0
C_2	0	0	60	0	0	60	0	0	60	0	0	60
	E			F			G			H		
C_0	40	10	10	0	30	30	20	20	20	0	0	60
C_1	10	40	10	0	60	0	20	20	20	0	60	0
C_2	10	10	40	0	0	60	20	20	20	60	0	0

Table 4. Performance measures for matrices A to H in Table 3

MATRIX	ACC	KAPPA	R_k	1-CEN	FMEAS	PACC
A	1.00	1.00	1.00	1.00	1.00	1.00
B	0.89	0.83	0.85	0.86	0.89	0.90
C	0.83	0.75	0.78	0.84	0.82	0.84
D	0.83	0.75	0.78	0.76	0.82	0.83
E	0.67	0.50	0.50	0.40	0.67	0.67
F	0.67	0.50	0.58	0.72	NAN	0.63
G	0.33	0.00	0.00	0.14	0.33	0.33
H	0.33	0.00	0.00	0.67	NAN	0.33

matrix has low accuracy. As can be seen in Table 2, the value of FMEAS is NaN for matrix H and matrix J. The corresponding accuracy for H and J are 0.80 and 0, respectively.

For the binary cases with balanced distribution, our *Pacc measure* is consistent with the accuracy. For the unbalanced cases, *Pacc* is more discriminative and closer to accuracy than other measures.

5.2 Analysis of Measures for 3-class Classification

Consider 3-class classification results (confusion matrices A to H) in Table 3 with balanced distribution of items in each class. The performance measures for these matrices are provided in Table 4. Figure 1 shows the plot of these measures for the corresponding matrices. F-measure is not included in this graph as some values are undefined for this measure. As we go from matrix A to matrix H, the number of misclassified items is increased. Therefore, we expect similar changes in the performance measure. From the performance measures in Table 4 and graph in Fig. 1, we observe the following.

- Accuracy measure is less discriminative. For example, matrices C and D, E and F, and G and H have the same accuracy. Therefore, we cannot rank these matrices based on accuracy. Likewise, Kappa statistic is less discriminative than our *Pacc measure*.

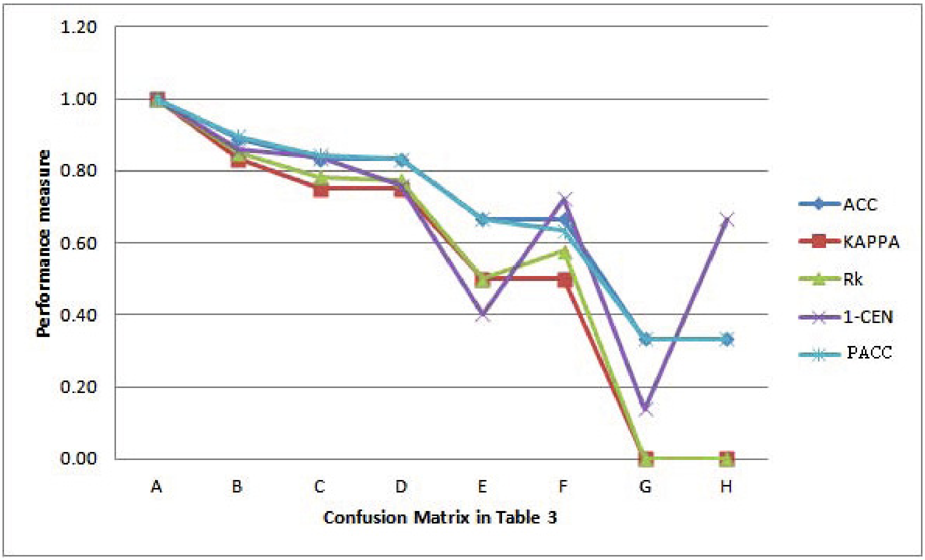


Fig. 1. Graph showing the plot of performance measures from Table 4

- R_k measure and our *Pacc measure* behave almost the same way in terms of ranking the matrices. However, the applicable range of -1 to +1 for R_k makes it difficult to correlate with accuracy measure.
- Confusion entropy measure is not consistent with other measures. Confusion entropy measure suggests confusion matrices F and H being better than E which does not look correct. Likewise, the results are not as expected when we compare matrix E and H. Matrix E has 40 items in each category correctly classified. On the other hand, Matrix H has none of the items in the first category and third category correctly classified. CEN suggests matrix H to be better result compared to matrix E. Therefore, the result is not as desired.
- F-measure is not computable for the confusion matrices F and H and is less discriminative compared to *Pacc measure.*
- *Pacc* follows the decreasing trend of values as we go from error matrix A to H. At the least, *Pacc measure* is as discriminative as any other measure.

5.3 Comparative Evaluation of Performance Measures

Discriminative property: One important requirement for a performance measure is the ability to distinguish confusion matrices. From the examples presented in earlier sections, it is seen that measures like accuracy and Kappa statistic have low discriminative power. To analyze the discriminative power of the various measures, we considered a 3-class problem with 5 items in each category. 21 combinations are possible for the distribution of 5 items into different categories. Therefore, a total of 9261 (21 x 21 x 21) confusion matrices are possible.

Table 5. Table showing the count of distinct values from 9261 possible confusion matrices in a 3-class problem with 5 items in each category

Measure	Num distinct values	Avg(abs diff with acc)
ACC	16	-
KAPPA	16	0.166
R_k	183	0.166
CEN	1504	0.359
FMEAS	368	0.169
Pacc	669	0.029

We calculated all the measures for these confusion matrices. Table 5 shows the count of distinct values obtained for 9261 confusion matrices. Accuracy and Kappa statistic have the lowest discriminative power as both of these measures have only 16 possible values for all of these matrices. Confusion entropy has the largest number of distinct values. *Pacc measure* also has high discriminative power.

NaN values and our resolution: Measures like F-measure and correlation coefficient may produce NaN as the result. For the 9261 possible confusion matrices in a 3-class problem with 5 items in each category, F-measure is NaN for almost 63% of the confusion matrices. If both precision and recall are 0, F-measure becomes undefined or NaN. Therefore, when these measures are used for the evaluation of classification results, necessary patches should be applied so that NaN is not an output. One approach to solve this would be to take the measure to be equal to 0. Nonetheless, there are multiple cases for the measure to be 0 making it difficult to distinguish/rank classification results. Correlation coefficient can produce NaN in case all the items are classified to a single class. If all items are classified into a single class, the variance for that class is 0. Since there are no items that are classified into other classes, the variance for those classes are also 0. This corresponds to a column in confusion matrix with non-zero values where the rest of the values are 0 in the confusion matrix.

Accuracy correlation: Table 5 provides the average of absolute difference between accuracy and other measures for the 9261 confusion matrices. For the correlation coefficient and Kappa statistic, the final average value is divided by 2 since its original range is from -1 to +1. The difference is the least for *Pacc measure*, thus revealing a high correlation of *Pacc measure* with accuracy . The inconsistency in confusion entropy measure is also reflected by the low correlation with accuracy. Kappa, R_k, and F-measure also have lesser correlation with accuracy compared to *Pacc measure*.

Scale invariance: A new set of confusion matrices were created by scaling the confusion matrices A to H provided in Table 3. For each of these matrices, the second row is doubled and the third row is increased by 5 times. The performance measures for the modified matrices are presented in Table 6. Figure 2 shows the

Table 6. Performance measures for the matrices A to H in Table 3 with second row increased twice and 3rd row increased by 5 times

MATRIX	ACC	KAPPA	R_k	1-CEN	FMEAS	PACC
A	1.00	1.00	1.00	1.00	1.00	1.00
B	0.96	0.92	0.92	0.92	0.92	0.93
C	0.94	0.88	0.89	0.93	0.85	0.88
D	0.94	0.88	0.88	0.89	0.86	0.89
E	0.67	0.44	0.46	0.46	0.61	0.65
F	0.88	0.75	0.77	0.85	NAN	0.73
G	0.33	0.00	0.00	0.23	0.30	0.35
H	0.25	0.04	0.06	0.76	NAN	0.33

Table 7. Table listing the properties of various measures

MEASURE	DISCRIMINATIVE	NAN VALUES	ACCURACY CORRELATION	SCALE INVARIANCE
ACC	LOW	NO	-	LOW
KAPPA	LOW	NO	LOW	LOW
R_k	MEDIUM	YES	LOW	MEDIUM
CEN	HIGH	NO	VERY LOW	HIGH
FMEAS	MEDIUM	YES	LOW	HIGH
PACC	HIGH	NO	HIGH	HIGH

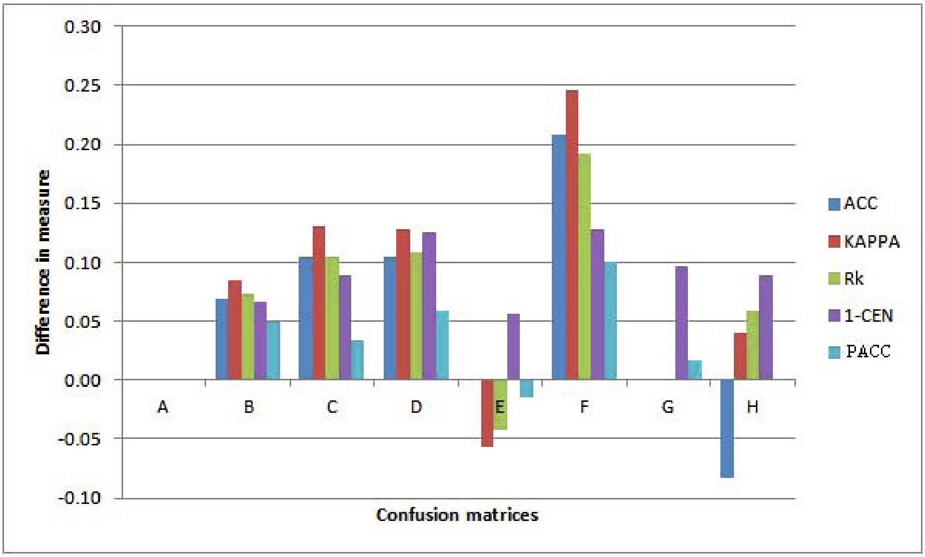


Fig. 2. Graph showing the difference in the values of performance measures in Table 4 and Table 6

plot of the difference in the performance measures in Table 4 and Table 6 (i.e., the difference in the measures between original and scaled matrices). F-measure is not included in this plot as there are some values for F-measure that are undefined. From the figure, we can see that accuracy and K-category correlation coefficient (R_k) are the most affected measures by the scaling. CEN measure and *Pacc measure* are comparatively less affected.

Table 7 provides a summary of various properties exhibited by the different measures. The level of discrimination and the level of scale invariance for accuracy is considered to be low. The levels for other measures are assigned relative to the level for accuracy. The accuracy correlation column is derived from the values in third column of table 5. The ranges for absolute difference with accuracy as ≤ 0.05, 0.05 - 0.1, 0.1 - 0.2, and ≥ 0.2 are considered to be high, medium, low, and very low levels respectively. *Pacc measure* compares best among the others as it has high level of discriminancy, does not result NaN values, is highly correlated with the accuracy measure, and the scale invariance is high.

6 Conclusion

In this paper, we explained the difficulties in comparing two classification experiments and highlighted the need for a good performance measure. We listed expected qualities in a good classifier performance measure and introduced a novel measure, *probabilistic accuracy (Pacc)*, which is based on the difference between probabilities of correct and incorrect classification. We made a comparative analysis of five multiclass performance measures and our proposed method considering different cases of confusion matrices. Correlation coefficient and Macro-averaged F-measure can produce NaN and this does not necessarily happen when the performance is very low. The K-category correlation coefficient (R_k) and Kappa statistic exhibit low correlation with accuracy. Likewise, we showed that confusion entropy measure is not consistent. The results show that the proposed *Pacc measure* is relatively consistent with the accuracy measure and also is more discriminant than others. The *Pacc measure* was shown to be less affected by the scaling of data. Therefore, our *Pacc measure* fairs best among others for the evaluation of classification results.

Choice of a performance measure and its analysis can be domain/problem dependent. Also, analysis based on a single measure can be misleading as different measure can produce contrasting decision for the selection of a classifier. The measures may have specific biases and hence should be carefully used and analyzed. This follows that results of classification experiments should be accompanied by the confusion matrix.

A classifer can be said to be useful only if it performs better than a random classifer. As future work, we plan to investigate methods to analyze performance based on how well a classifier is performing compared to a random classifier. Similarly, we would like to formulate methods to evaluate performance where the order of classification category is important. Likewise, we also plan to investigate matrix normalization techniques to our proposed method.

Acknowledgments. This research was supported by National Institutes of Health (GM-090453) grant.

References

1. Cumbaa, C., Jurisica, I.: Protein crystallization analysis on the world community grid. Journal of Structural and Functional Genomics, 61–69 (2010)
2. Espndola, R.P., Ebecken, N.F.F.: On extending F-measure and G-mean metrics to multi-class problems. In: Data Mining, Text Mining and their Business Applications (2005)
3. Özgür, A., Özgür, L., Güngör, T.: Text categorization with class-based and corpus-based keyword selection. In: 20th Inter. Conf. on Comp. and Inf. Sci., pp. 606–615 (2005)
4. Landgrebe, T., Duin, R.: Efficient multiclass roc approximation by decomposition via confusion matrix perturbation analysis. IEEE Trans on Patttern Analysis and Machine Intelligence 30, 810–822 (2008)
5. Rees, G.S., Wright, W.A., Greenway, P.: Roc method for the evaluation of multi-class segmentation classification algorithms with infrared imagery (2002)
6. Yang, B.: The extension of the area under the receiver operating characteristic curve to multi-class problems, vol. 2, pp. 463–466 (2009)
7. Wei, J., Yuan, X., Hu, Q., Wang, S.: A novel measure for evaluating classifiers. Expert Systems with Applications 37, 3799–3809 (2010)
8. Gorodkin, J.: Comparing two k-category assignments by a k-category correlation coefficient. Computational Biology and Chemistry 28, 367–374 (2004)
9. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. 45, 427–437 (2009)
10. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: An overview (2000)
11. Garca, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694 (2009)
12. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
13. Perner, P.: How to interpret decision trees? In: Perner, P. (ed.) ICDM 2011. LNCS, vol. 6870, pp. 40–55. Springer, Heidelberg (2011)
14. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20, 37–46 (1960)

A Single-Domain, Representation-Learning Model for Big Data Classification of Network Intrusion

Shan Suthaharan

Department of Computer Science, University of North Carolina,
Greensboro, NC 27412, USA,
ssuthaharan@uncg.edu

Abstract. Classification of network traffic for intrusion detection is a Big Data classification problem. It requires an efficient Machine Learning technique to learn the characteristics of the rapidly changing varieties of traffic in large volume and high velocity so that this knowledge can be applied to a classification task. This paper proposes a supervised-learning technique called the Unit Ring Machine which utilizes the geometric patterns of the network traffic variables to learn the traffic characteristics. It provides a single-domain, representation-learning technique with a class-separate objective for the network intrusion detection. It assigns a large volume of network traffic data to a single unit-ring and categorizes them based on the varieties of network traffic, making it a highly suitable technique for the Big Data classification of network intrusion traffic.

Keywords: Machine learning, big data classification, intrusion detection, unit-circle representation, supervised learning, labeled datasets.

1 Introduction

The term Big Data describes the data that cannot be managed efficiently by the current techniques, tools and devices. The Big Data is defined as the data that must be studied in whole rather than in parts using the standard sampling techniques. This is because the sampling may ignore some of the important properties that may play a negative role in the formulation of long-term models. The paper [1] has recently defined the Big Data as a dataset that consists of data points represented by a mathematical relationship between three independent variables, Volume, Velocity and Variety. With this definition the complexity in analyzing and visualizing this space, and extracting useful patterns from this space can lead to Big Data problem. The network traffic information collected at an intermediate system in a distributed network show high volume of data with large varieties of traffic types arriving in high velocity. These Big Data properties of network traffic lead to data with multiple distributions within a single dataset. Hence the application of a standard Machine Learning (ML) technique to this Big Data environment of network intrusion can be challenging [2], [3]. There are two

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 296–310, 2013.

main problems associated with the Big Data classification of network traffic. The first problem is the processing power required for the analysis and visualization of the massive size data. This problem can be addressed by using the emerging Big Data platforms like the Hadoop Distributed File System (HDFS) [4]. The HDFS takes the input data with an associated task and divides them into smaller fragments (data and task) using the two concepts called the Map and Reduce. It then distributes the tasks and data to multiple nodes in a cluster and executes these tasks in those nodes. This distribution scheme helps to achieve bandwidth requirements and a fault-tolerant system.

The second problem is to find a robust representation for the data to learn the characteristics of the traffic distribution. It can help develop classifiers using training dataset and validate the classifiers using the test dataset. This data representation problem can be addressed by representation-learning techniques which are used in ML research [5]. The representation-learning can help learn robust data representations that can be used to classify regular and intrusion traffic using Big Data platforms. The intrusion datasets are high dimensional and hence the representation-learning can be used to select features that contribute to the classification of the traffic types (i.e. feature selection), transform the high dimensional dataset into several manageable and meaningful low dimensional datasets (i.e. feature extraction), and define a suitable measure that can calculate the class separation distances adequately (i.e. distance metric learning) [6].

In a classification task a representation-learning model is trained using a labeled (training) dataset. This trained model is then validated using cross-validation ap-proaches and applied to a classification task of new incoming data. This process is carried out under the assumption that the pattern of the distribution is the same for all datasets, although multiple distributions are possible within a dataset. In this case the representation-learning is called single domain, where the multiple datasets (training, test and incoming datasets) represent the same domain. Although the single-domain, representation-learning will not give a complete solution to Big Data classification, it must be studied to understand the suitability of the new concept, the unit-circle algorithm (UCA) [7] for the development of a network intrusion detection system.

This paper proposes a single-domain, representation-learning technique with a class-separate objective which characterizes the geometric representation properties of the network intrusion traffic using the concept of unit-circle algorithm and helps to classify the network intrusion and regular traffic. This approach is called the unit-ring machine (URM) because it catches the distribution similarity using the unit-circle representation and determines the distribution with unit-rings, where a unit-ring con-tains a collection of unit-circles. It provides a theory and results focusing on feature extraction and distance metric learning for a predetermined pair of features, hence the feature selection learning is not discussed in this paper. However a simple feature selection approach is presented for learning the selection of feature variables. Since the Big Data and Visualization go hand-in-hand, the results and findings are discussed mostly using the visual examples rather than numerical examples.

2 Background

The importance of the representation-learning has been discussed in detail in a recent paper by Bengio et al [5]. They addressed its applications to speech recognition, object recognition, natural language processing and signal processing. However its application to network security is also very important and that is the focus of this paper. The intrusion datasets depend on many feature variables and extracting robust features under the constraints of a large number of feature variables is a challenging problem. Several supervised learning approaches [7], [8], [9], [10], [11] have been proposed for the classification of network intrusion traffic. However the learning technique based on the Support Vector Machine (SVM) became popular for intrusion detection because of its classification accuracy. The SVM technique is computationally expensive and thus several versions of SVM have been proposed subsequently to address this problem [12], [13]. These techniques do not provide generalized solutions with associated representation learning and class-separate objectives. Recently simultaneous classification and feature selection process is integrated into SVM [13]. However it is not suitable for dynamically growing large amount of data like the network intrusion traffic because of its requirements for a subset of prior knowledge. Hence other alternative approaches must be explored for Big Data situation.

Several representation learning approaches, that might be suitable for a dynamically growing dataset, have been proposed for intrusion detection in recent years [14], [15], [16] and they mainly focus on the feature selection aspect of the representation learning. The representation learning generally carried out independently and they do not include associated class-separate objectives. Because of the lack of coordination between the learning and classification, the classification accuracy is affected negatively in many applications. Therefore it is appropriate to develop an associated class-separate objective to each representation learning technique. Most recently Tu and Sun [6] addressed this issue and proposed a cross-domain representation learning framework with an associated class-separate objective. This approach is not developed for Big Data classification that can utilize the emerging HDFS technology.

A recent study with an intrusion dataset indicates that the intrusion can be represented by unit-circles and this representation can be used for classification of traffic types [7]. This study introduced the Unit Circle Algorithm mentioned earlier. It is a two-class, two-dimensional technique and it provides a static classifier for the classification of intrusion and regular traffic with no representation-learning. The static classifier introduces a fuzzy boundary in the UCA-based classification. When the Big Data is considered and analyzed, the classification need to deal with datasets with rapidly changing varieties of traffic types in large volume and high velocity. The fuzzy boundary problem of UCA also causes severe inaccuracies in the classification results. Therefore it is important to adopt a learning technique to train the machine to find a suitable boundary parameter. In this paper a boundary parameter is introduced to the UCA approach to represent the intrusion data in unit-rings and allow representational learning with class-separate objective. The proposed URM technique adopts the concept of unit-circle representation of intrusion datasets, but suggests some modifications which will enable representation learning with class-separate objective. The motivation behind this approach is to represent the multidimensional intrusion data into multiple two-dimensional representation using unit-circles and help the HDFS technology to solve Big Data intrusion classification. In this paper a suitable representation learning technique with an associated class-separate objective is developed and evaluated using a label dataset.

3 Unit-Circle Algorithm

The UCA approach presented in a short-paper [7] is discussed in this section with additional examples. The UCA approach allows two feature variables as input and constructs the concentric unit-circles with radius less than or equal to 1. As stated in the development of UCA classifiers, the observations of the feature variables are normalized to satisfy the formulation of unit-circles. As an example, if we represent two feature variables by X and Y, and their normalized observations by xi and yi (where i=1..n) respectively, then the following inequalities can be satisfied [7]:

$$0 \leq x_i, y_i \leq 1, where, \forall i = 1..n \tag{1}$$

The UCA then constructed the unit-circle region according to the following mathematical expression presented in [7]:

$$R = \big\{(x_i, y_i)|0 \leq x_i^2 + y_i^2 \leq 2, i = 1..n\big\} \tag{2}$$

This unit-circle region was then divided into two disjoint regions to generate suita-ble classifiers for two types of data [7]:

$$R_1 = \big\{(x_i, y_i)|0 \leq x_i^2 + y_i^2 \leq 1\big\} \tag{3}$$

$$R_2 = \big\{(x_i, y_i)|1 < x_i^2 + y_i^2 \leq 2\big\} \tag{4}$$

This class separation was carried out based on the observation that the back attack and regular traffic in the NSL-KDD [17] dataset satisfy this property. If we divide the inequalities by 2 both sides and take square root then we have classifiers that are separated by 1/2. However a fuzzy boundary was noticed and hence a tolerance value of 0.005 was used to reduce this fuzziness. Thus the classifiers were modified in [7] as:

Regular traffic class

$$C_1 = \Big\{(X, Y)|0 \leq \sqrt{(X^2 + Y^2)/2} \leq 0.7121\Big\} \tag{5}$$

Intrusion traffic class

$$C_2 = \Big\{(X, Y)|0.7021 < \sqrt{(X^2 + Y^2)/2} \leq 1\Big\} \tag{6}$$

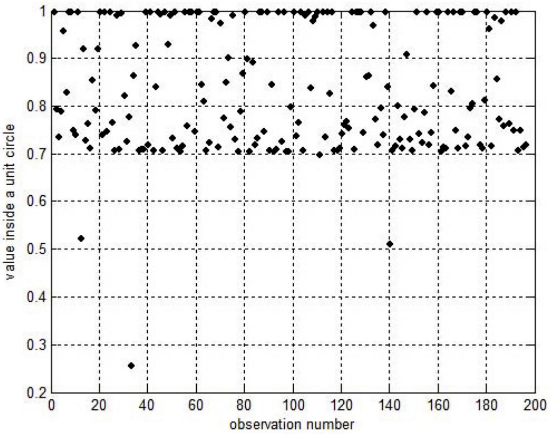


Fig. 1. Normalized values of back attack traffic

These classifiers were applied to the classification of back attack and normal traffic in the NSL-KDD datasets [7]. The features 5 and 32 were used to construct the unit-circles of the UCA (features 23 and 24 were used for Neptune attack and normal traffic). These features were selected based on a distribution-based feature selection algorithm. The results indicated that the distinct properties of the intrusion and regular traffic activities can be obtained with these classifiers.

Similar results are obtained with different random samples of the back attack and regular traffic and hence these new results are presented in Figures 1 and 2 to support the classification capabilities of the UCA concept. These figures show a sharp boundary between the classes and the tolerance of 0.005 can handle the fuzziness reasonably. In addition the results obtained using the UCA classifiers for the Neptune attack and its corresponding normal traffic are presented in Figure 3 and Figure 4. In this case the features 23 and 24 are used to construct the unit-circles of the UCA. This simulation shows no sharp boundary between the classes and the tolerance level 0.005 cannot handle, thus high fuzziness occurred in this case. Based on the results presented in these figure, we may conclude that the UCA classifiers can classify the attacks and normal, however fuzzy boundary is still a problem. This problem can be increased significantly with the data that satisfy the definition of Big Data. Hence this approach is not suitable for the classification of Big Data. In addition, the same features cannot be used to construct the unit-circles for dif-ferent combinations of attack and normal traffic classes. It requires a feature selection module. To reduce fuzzy boundary effect and to use HDFS technology, a combined representation learning and class-separate objective is required. The UCA provides a static boundary as a classifier and hence representation learning is not possible. However UCA supports the concept of representing intrusion data by unit-circles and it play a major role in the proposed Unit Ring Machine for representation-learning in which a ring represents a collection of related unit-circles.

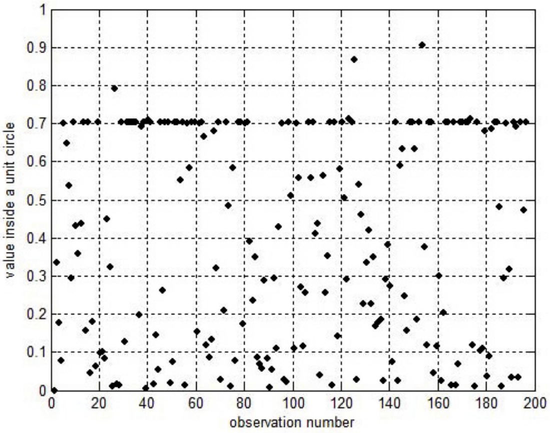


Fig. 2. Normalized values of regular network traffic

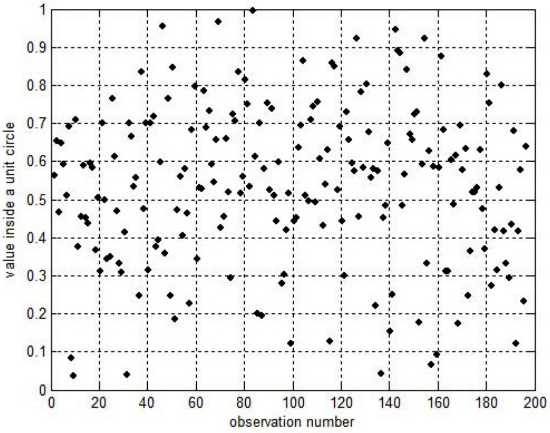


Fig. 3. Normalized values of Neptune attack traffic

4 Unit Ring Machine

The concept of URM is to represent the data by unit-rings and classify the data by classifying the unit-rings. Compared to UCA, the URM has adjustable widths for representation learning. The adjustable ring width will help feature extraction learning and distance metric learning with class-separate objective. The URM approach, like UCA, accepts two feature variables and constructs unit-rings. The feature variables are denoted by X and Y, and their normalized observation are represented by xi and yi, respectively, hence they satisfy the following inequalities:

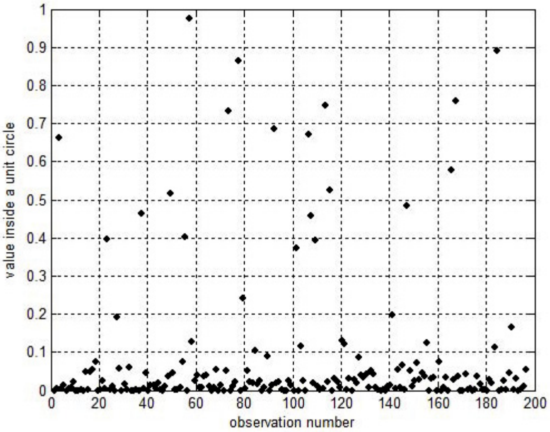


Fig. 4. Normalized values of regular network traffic

$$0 \leq x_i, y_i \leq 1, where, \forall i = 1..n \tag{7}$$

These normalized observations of the feature variables can form the following:

$$R = \left\{(x_i, y_i)|0 \leq \sqrt{(x_i^2 + y_i^2)/2} \leq 1, i = 1..n\right\} \tag{8}$$

In the URM approach, this circular region is divided into k unit-rings with equal widths as follows:

$$R_j = \left\{(x_i, y_i)|(j-1)/k \leq \sqrt{(x_i^2 + y_i^2)/2} \leq j/k\right\} \tag{9}$$

In this equation j=1..k, which represents a ring number, and i=1..n. Based on the concept of UCA these rings can be divided into two disjoint classes as follows:

Regular traffic class

$$C_1 = \cup R_j, where, 0 \leq j \leq \left\lceil k/\sqrt(2)\right\rceil \tag{10}$$

Intrusion traffic class

$$C_2 = \cup R_j, where, \left\lceil k/\sqrt(2)\right\rceil < j \leq k \tag{11}$$

These are the classifiers of URM, where the operator provides the rounded in-teger value for the operand z, and they will be trained using the parameter k.

Representation Learning: The proposed representation learning involves the ex-traction of features (e.g. geometric patterns, statistical distributions), and the devel-opment of a distance metric that estimates the distance between the unit-rings of data point for a pair of feature variables. The proposed learning mechanism selects a pair of features X and Y by sorting their normalized observations xi and yi and measuring their Euclidean distances. If the distance shows a significant separation then these features are selected for the input to the URM. In general the distance metric learning is for maximizing the distance between the points in different classes. In the proposed approach each ring has a set of data points and thus the distance metric is between the unit-rings to reduce the fuzzy boundary. In other words the distance metric defines the width of a unit-ring. In the proposed approach the feature extraction learning relates to the learning of data distribution within each unit-ring. Therefore the representation-learning selects the parameter width (1/k) for the unit-rings and then calculates the percentage of data points that fall inside each circle with respect to the data points fall in the other circles together. Thus the URM technique will learn these properties of the data as a representation-learning approach.

5 Simulation and Results

In this section the proposed representation learning and class-separate capability are evaluated using specific examples. The URM technique is applied to the subsets of the NSL-KDD training and test datasets and the results are presented. The repre-sentation-learning involves feature extraction learning and distance metric learning for a fixed pair of features selected by the Euclidean distance metric for a traffic class.

Training Results: In the training phase, the back attack and regular traffic pair; and the Neptune attack and regular traffic pair are used. For back attack and regular traffic, features 5 and 32 are used and they are mapped to unit-circles to demonstrate the learning technique. The radius of these unit-circles are sorted and plotted in Figure 5, which demonstrates the Euclidean between these traffic classes.

Figure 5 allows us to calculate the Euclidean distance between the back attack (red) and regular traffic (blue). A significant separation between the back attack and regular traffic can be clearly seen. It also shows that both back attack and regular traffic can have the same or similar radius values and it will be a significant problem when the dataset grows to Big Data.

Similarly Figure 6 demonstrates the distance between the Neptune attack (red) and regular traffic (blue) using the features 23 and 24. A significant separation between the Neptune attack and regular traffic can also be seen, and the same conflict can occur for the Neptune attack and regular traffic as well when the dataset grows to Big Data. For the distance metric learning, the width for a unit-ring is selected as 0.05 using n-fold cross-validation test [18], and the unit-rings that have less than 5 data points within these rings will be eliminated.

Figure 7 shows the unit-rings of the back attack and regular traffic calculated using the width of 0.05. Each point in this figure represents a concentric ring corresponding to the radius mapped on the y-axis. The space between two points represents the width of the ring. For example if we consider the red point at (4, 0.4) then it is the fourth concentric ring of the back attack class with

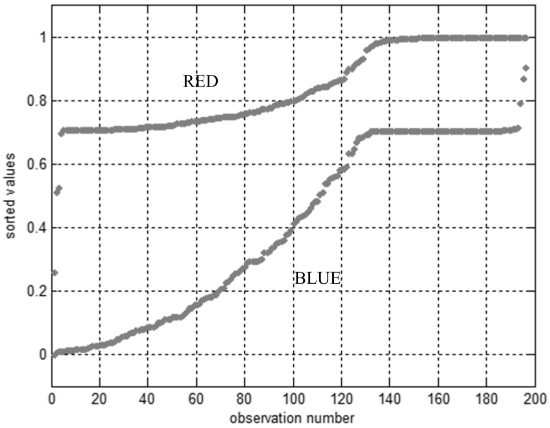


Fig. 5. Sorted representation of back attack and regular traffic

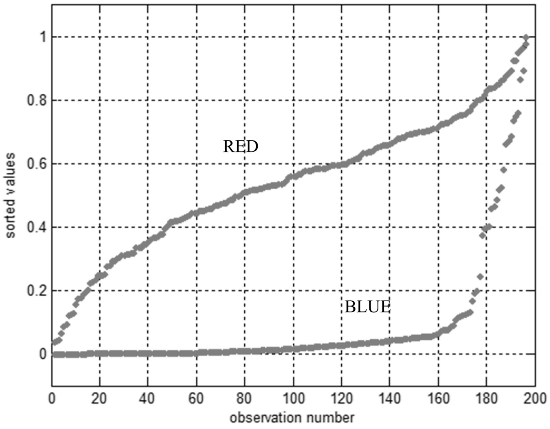


Fig. 6. Sorted representation of Neptune attack and regular traffic

radius 0.4 and the width of 0.05 (i.e. 1/20). Similarly if we consider the blue point at (9, 0.4) then it is the ninth concentric circle of the regular traffic class, its radius and widths are also 0.4 and 0.05. As a whole, by drawing a horizontal line at a particular radius, we can say that there is a high chance of having a single ring with data points from both the intrusion class and the regular traffic class. However the feature learning can eliminate the rings that have significantly low number of records. In this experiment the rings with less than 5 records (data points) are eliminated from the computation of the classifiers. Figure 8 demonstrates the results of this process with unit-circles. If we

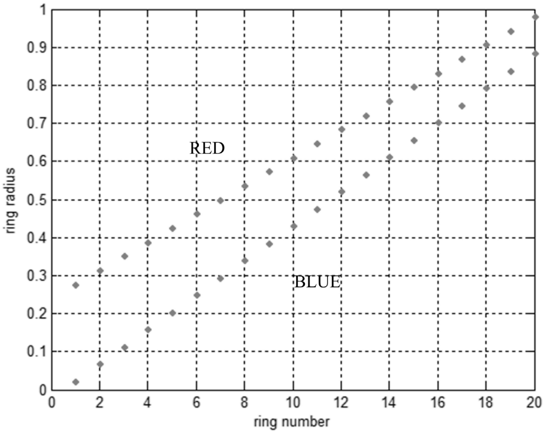


Fig. 7. Rings assigned by URM for back attack and regular traffic in the training data

compare the Figures 7 and 8, and consider the blue points (i.e. regular traffic class) then all the blue points above 0.7 (radius) are eliminated. Note that some blue points below 0.7 are also eliminated. Similarly if we consider the red points (i.e. intrusion traffic class) then all the red points below 0.7 are eliminated. This process completely removes the fuzziness at the class boundary 0.7 and this effect can be clearly seen in Figure 8, and thus meets the class-separate objective.

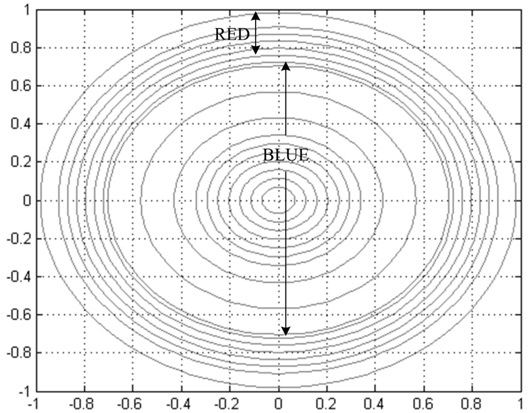


Fig. 8. Rings detected by URM for back attack and regular traffic in the training data

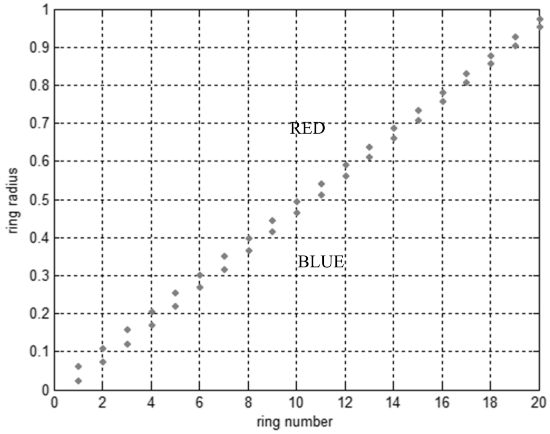


Fig. 9. Rings assigned by URM for Neptune and regular traffic in the training data

Similar results of Neptune attack and regular traffic are obtained and also presented in Figures 9 and 10. These figures demonstrate that the learning can help to remove fuzziness at 0.2 and classify the Neptune attack and regular traffic robustly. Hence the results of training phase show that the URM can be trained successfully to achieve class-separate objective with the proposed representation learning. In summary, the knowledge acquired by the URM-based representation learning for the intrusion datasets are with (i) features 5 and 32 for back attack and regular traffic pair, and (ii) features 23 and 24 for Neptune attack and regular traffic pair are (i) twenty rings; each has the width of 0.05; and (iii) rings with less than 5 records (data points) should be eliminated to achieve an acceptable class-separate objective.

Testing Phase Results: In this testing phase, the leaned and retained knowledge at the training phase will be transferred to testing phase. For this purpose the test dataset of NSL-KDD is used. Firstly the back attack and regular traffic records, with features 5 and 32, of this test dataset are analyzed with the URM technique and the results are presented in Figures 11 and 12. Figure 11 shows all the possible 20 rings calculated for the test dataset and it demonstrates similar properties as of training dataset pre-sented in Figure 7. Figure 12 shows a perfect class-separate for the back attack and regular traffic and these results are comparable with the results in Figure 8. Hence the class-separate objective is achieved.

In the next simulation, Neptune attack and regular traffic records, with features 23 and 24, of the NSL-KDD test dataset are analyzed using the URM technique. The results are presented in Figures 13 and 14. Figure 13 shows all the possible 20 rings calculated for the test dataset and it shows some dissimilarity to the results of the training dataset presented in Figure 9. Also,

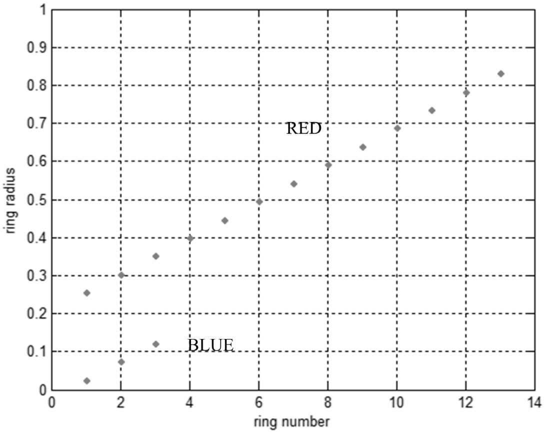


Fig. 10. Rings detected by URM for Neptune and regular traffic in the training data, similar unit-circle separation in Figure 8 can be seen

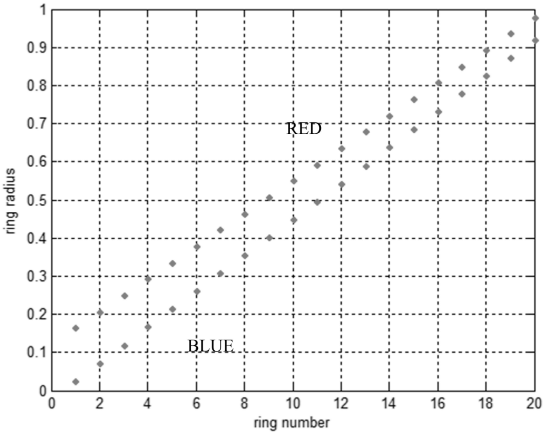


Fig. 11. Rings assigned by URM for back and regular traffic in the test data

the classification results presented in Figure 14 does not show perfect class-separate. However the isolated point (blue, regular traffic) can be removed with outlier detection techniques. Further training (i.e. multiple instance learning) using the proposed URM technique is required to separate other overlapping points.

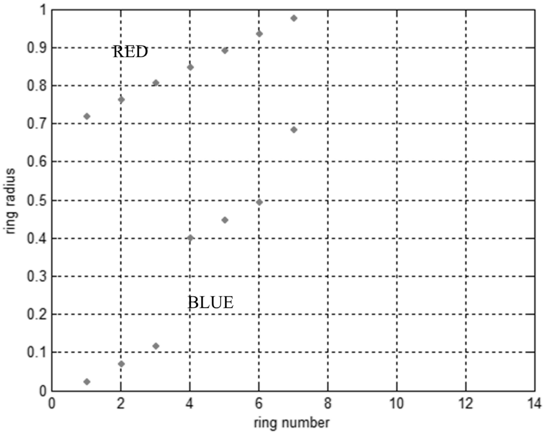


Fig. 12. Rings detected by URM for back and regular traffic in the test data, similar unit-circle separation in Figure 8 can be seen

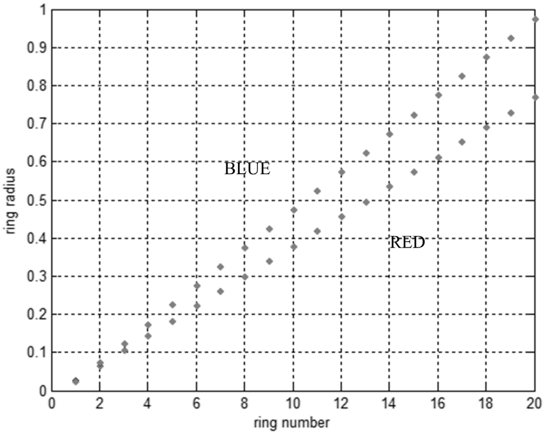


Fig. 13. Rings assigned by URM for Neptune and regular traffic in the test data

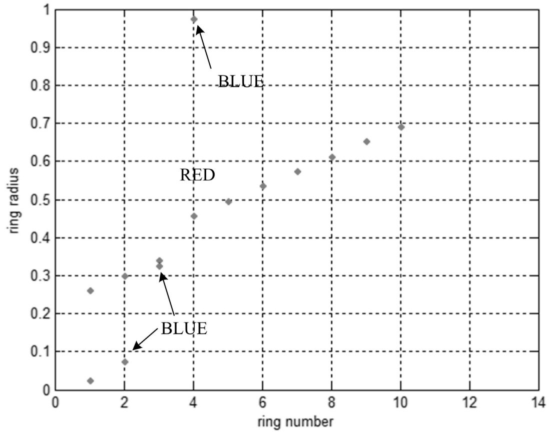


Fig. 14. Rings detected by URM for Neptune and regular traffic in the test data, similar unit-circle separation in Figure 8 can be seen

6 Conclusion

The concept of UCA helped build the URM technique for intrusion detection. The proposed URM technique used the geometric properties of the intrusion and regular network traffic, and assigned many traffic records to a single ring to handle Big Data representation and accomplish class-separate objective. The results are evaluated by the visual tools only, because it provides the sufficient information to understand and assess the performance of the proposed representation-learning model. The evaluation of the URM technique, using a subset of the NSL-KDD dataset, showed that this technique can be used to manage Big Data classification tasks. It also showed that it can be used to reduce the fuzzy boundary between network traffic classes through a single-domain, representation-learning. However, it can be improved to get better classification accuracy for every intrusion and regular traffic class tested hence further research is required with multiple-domain, representation-learning with knowledge-transfer and class-separate objectives. The proposed technique will be implemented and validated using the emerging HDFS platforms as a part of the on-going research.

References

1. Gartner, `http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf`
2. Laskov, P., Dussel, P., Schafer, C., Rieck, K.: Learning intrusion detection: supervised or unsupervised? In: Proceedings of the 13th ICIAP Conference, pp. 50–57 (2005)

3. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. Informatica 31, 249–268 (2007)
4. White, T.: Hadoop: The Definitive Guide, 3rd edn. O' Reilly Media Inc. (2012)
5. Bengio, Y., Courville, A., Vincentar, P.: Representation Learning: A Review and New Perspectives. arXiv:1206.5538v2 [cs.LG] (2012)
6. Tu, W., Sun, S.: Cross-domain representation-learning framework with combination of class-separate and domain-merge objectives. In: Proceedings of the CDKD 2012 Conference, pp. 18–25 (2012)
7. Suthaharan, S.: A unit-circle classification algorithm to characterize back attack and normal traffic for intrusion detection. In: Proc. of the IEEE International Conference on Intelligence and Security Informatics, pp. 150–152 (2012)
8. Laskov, P., Schafer, C., Kotenko, I.: Intrusion detection in unlabeled data with quarter-sphere support vector machines. In: Proceedings of the DIMVA Conference, pp. 71–82 (2004)
9. Huang, G., Chen, H., Zhou, Z., Yin, F., Guo, K.: Two-class support vector data description. Pattern Recognition 44, 320–329 (2011)
10. Corona, I., Giacinto, G., Roli, F.: Intrusion detection in computer systems using multiple classifier systems. Studies in Computational Intelligence (SCI) 126, 91–113 (2008)
11. Giacinto, G., Perdisci, R., Roli, F.: Network intrusion detection by combining one-class classifier. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 58–65. Springer, Heidelberg (2005)
12. Mangasarian, O.L., Musicant, D.R.: Lagrangian support vector machine classification. TR 00-06, Data Mining Institute, Department of Computer Science, University of Wisconsin, USA (2000), `ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.pdf`
13. Jeyakumar, V., Li, G., Suthaharan, S.: Support vector machine classifiers with uncertain knowledge sets via robust convex optimization. Optimization the Journal of Mathematical Programming and Operations Research, 1–18 (2012)
14. Chen, Y., Li, Y., Cheng, X., Guo, L.: Survey and taxonomy of feature selection algorithms in intrusion detection system. In: Lipmaa, H., Yung, M., Lin, D. (eds.) Inscrypt 2006. LNCS, vol. 4318, pp. 153–167. Springer, Heidelberg (2006)
15. Kayacik, H.G., Zincir-Heywood, A.N., Heywoo, M.I.: Selecting features for intrusion detection: A feature relevance analysis on KDD 99 Intrusion Detection Datasets. Association of Computer Machinery Press, 85–89 (2006)
16. Li, Y., Wang, J., Tian, Z., Lu, T., Young, C.: Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. Computers and Security 28(6), 466–475 (2009)
17. NSL-KDD, `http://www.iscx.ca/NSL-KDD/`
18. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data mining, Inference, and Prediction. Springer, New York (2001)

Relation Decomposition: The Theory

Robert Martin Haralick, Ligon Liu, and Evan Misshula

Computer Science, Graduate Center
City University of New York
New York, NY 10016, USA
haralick@aim.com
http://haralick.org

Abstract. Data Mining explanatory models must deal with relevance: how values of different data items are relevant to the values of other data items. But to be able to construct explanatory models, and in particular causal explanatory models, we must do so by first understanding irrelevance and exactly how irrelevance plays a role in explanatory models. The reason is that the conditional irrelevance or conditional no influence relation defines the boundaries of the ballpark within which an explanatory model lives.

This paper reviews the theory of no influence in the mathematical relation data structure. We discuss the relationship this theory has to graphical models and we define a coefficient of no influence and give a method for the estimation of its p-value.

1 Introduction

Informally, A is irrelevant to B, or A has no influence on B, means A is unrelated, extraneous and not pertinent to B. Formally, a set of variables A has no influence on a set of variables B in the context of a set of variables C if and only if for each tuple of values for C, every tuple of values taken by A can co-occur with every tuple of values taken by B. This is equivalent to the qualitative independence of Shafer et. a.[8] and the multivalued dependence of Fagin and Vardi[3] and it bears a direct similarity to the causal irrelevance definition of Galles and Pearl[4]. Lauritzen (1996)[6] states it this way: knowing C, reading A is irrelevant for reading B. Consider a causal context in which C is the direct cause of A. If B is a non-descendant of A, then A has no influence on B given C. This is called the causal Markov condition.[5] Other phrases for this is that C shields B from A[9] or that C screens off B from A.

When A has no influence on B in the context of C, then the observed data relation can be decomposed into two factor relations whose relational join is the observed data relation. Each factor relation is a projection of the observed data relation. One factor is the projection on $A \cup C$ and the other factor is the projection on $B \cup C$. With many no influences, there will be multiple factors. Each factor relation will be a projection of the observed data relation. Then a relation join decomposition can be computed in which the factor relations of the join take on the role similar to prime factors in an arithmetic prime factor decomposition and all the tuples of the observed relation are determined by the relation join of the factor relations.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 311–324, 2013.

The relation join decomposition can lead to explanatory models. If the small factor relations are metaphorically thought of as little stories, the relation join of the small factor relations creates that big story consistent with all the small stories. This process essentially threads or chains the small stories together. This chaining or threading is, in fact, equivalent to propositional logic reasoning. The small factor relations are essentially relational constraints and the tuples that belong to the join decomposition are tuples that satisfy all the constraints.

Real observed data relations typically would not have perfect decompositions, the same way that in multi-variate analysis it would be rare to observe a sample correlation coefficient to be zero or a sample regression coefficient to be zero. The way this is handled is for approximate models to be set up in which the correlation coefficients or regression coefficients that are statistically insignificantly different from zero are set to zero. We follow the analogous methodology with relation decomposition.

The first four sections define the concepts we will need, the notation we use and the theory. The last section explains how we use the theory to form relation decompositions.

2 Preliminary Concepts

2.1 Index Sets and Indexed Relations

Let $X_1, \ldots, X_N$ be the N variables associated with a relation. Let L_n be the set of possible values variable X_n can take. Let R be a data set to be mined. As our methodology does not deal with probability, we can assume that each data tuple is unique: it occurs precisely once. Hence the observed data can be represented as a relation R.

$$R \subseteq \bigtimes_{n=1}^{N} L_n$$

We will be working with many relations associated with different and overlapping variable sets and therefore over different domains. For this purpose we will carry an index set along with each relation. The index set indexes the variables associated with the relation. An index set is a totally ordered set.

Definition 1. *$I = \{i_1, \ldots, i_K\}$ is an* **index set** *if and only if $i_1 < i_2 < \cdots < i_K$.* ♠

For a natural number N, we use the convention that $[N] = \{1, \ldots, N\}$ and $|A|$ designates the number of elements in the set A. Next we need to define Cartesian product sets with respect to an index set.

Definition 2. *If $I = \{i_1, \ldots, i_K\}$ is an index set, we define the* **Cartesian product**

$$\bigtimes_{i \in I} L_i = \bigtimes_{k=1}^{K} L_{i_k} = L_{i_1} \times L_{i_2} \times \cdots \times L_{i_K}$$

♠

The definition tells us that the order in which we take the Cartesian product $\bigtimes_{i \in I} L_i$ is precisely the order of the indexes in I.

Now we can define the relation with its index set as a pair called the indexed relation.

Definition 3. *If I is an index set with $|I| = N$ and $R \subseteq \bigtimes_{i \in I} L_i$, then we say (I, R) is an* **indexed N-ary relation** *on the range sets indexed by I. We also say that (I, R) has dimension N. We take the range sets to be fixed. So to save writing, anytime we have an indexed relation (I, R), we assume that that $R \subseteq \bigtimes_{i \in I} L_i$, the sets L_i, $i \in I$, being the fixed range sets.* ♠

We can perform the usual set operations of union and intersection with the structures (I, R) and (J, S) and when $I = J$:

$$(I, R) \cup (I, S) = (I, R \cup S)$$
$$(I, R) \cap (I, S) = (I, R \cap S)$$

The subset relation also has the usual meaning.

If $R \subseteq \bigtimes_{i \in I} L_i$ and $S \subseteq \bigtimes_{i \in I} L_i$, then

$$R \subseteq S \text{ if and only if } (I, R) \subseteq (I, S)$$

2.2 Projection

Next we need the concept of projection. If (J, R) is an indexed relation and $I \subseteq J$, the projection of $(J.R)$ onto the ranges sets indexed by I is the indexed set (I, S) where a tuple $(x_1, \ldots, x_{|I|})$ is in S whenever for some $|J|$-tuple $(a_1, \ldots, a_{|J|})$ of R, x_i is the value of that component j of $(a_1, \ldots, a_{|J|})$ where the variable associated with place i of the tuple $(x_1, \ldots, x_{|I|})$ is the same as the variable associate with place j of $(a_1, \ldots, a_{|J|})$. The projection operator defined here is the same as the projection operator in the relational database world.

Let I and J be index sets with $I \subseteq J$. The **projection** operator projecting a relation on the range sets indexed by J onto the range sets indexed by I is written as

$$\pi_I(J, R) = (I, S)$$

Projection operators are idempotent. For $J \subseteq I$, $\pi_J(I, R) = \pi_J(\pi_J(I, R))$. More generally, if $K \subseteq J \subseteq I$, then $\pi_K(\pi_J(I, R) = \pi_K(I, R)$.

2.3 Relation Join

It follows from the definition of join that the join of projections of an indexed relation (K, R) must be a superset of (K, R) when the union of the index sets of the projections equals the index set K.

There is an join absorption law. If (M, R) is an indexed relation and $I \subseteq J \subseteq M$ are index sets, then

$$\pi_I(M, R) \otimes \pi_J(M, R) = \pi_J(M, R)$$

.

The relation join has a number of properties as stated in the following proposition.

Proposition 1. *Let (I,R), (I,U), (J,S), and (K,T) be indexed relations with $R \subseteq \bigtimes_{i\in I} L_i$, $S \subseteq \bigtimes_{j\in J} L_j$, $T \subseteq \bigtimes_{k\in K} L_k$, and $U = \bigtimes_{i\in I} L_i$. Then,*

$$\begin{aligned}
(I,R)\otimes(J,S) &= (J,S)\otimes(I,R)\\
((I,R)\otimes(J,S))\otimes(K,T) &= (I,R)\otimes((J,S)\otimes(K,T))\\
(I,R)\otimes(I,U) &= (I,R)\\
(I,R)\otimes(I,R) &= (I,R)\\
(I,R)\otimes[(J,S)\cup(J,T)] &= [(I,R)\otimes(J,S)]\cup[(I,R)\otimes(J,T)]\\
[(I,R)\cup(I,T)]\otimes(J,S) &= [(I,R)\otimes(J,S)]\cup[(I,T)\otimes(J,S)]
\end{aligned}$$

$$(J,S)\subseteq(J,T) \textit{ implies } (I,R)\otimes(J,S)\subseteq(I,R)\otimes(J,T)$$

$$(I,R)\subseteq(I,T) \textit{ implies } (I,R)\otimes(J,S)\subseteq(I,T)\otimes(J,S)$$

This makes the relation join operator one which is an operation of an idempotent commutative groupoid and as well it is union preserving on both its left and right operands. The union preserving property makes it increasing on both its left and right operands.

We now define the restriction operator on indexed relations. This is the same as the selection operator in the relational database world.

Definition 4. *Let (I,R) be an indexed relation with $R \subseteq \bigtimes_{i\in I} L_i$. Let $J \subseteq I$ and $a \in \bigtimes_{j\in J} L_j$. Then the restriction of (I,R) to (J,a) is denoted by $(I,R)|_{(J,a)}$ and is defined by*

$$(I,R)|_{(J,a)} = (I,\{r\in R \mid \pi_J(I,r) = (J,a)\})$$

♠

Having the concept of restriction, we can state the Join Representation Theorem.

Theorem 1. *Let $M = I \cap J \neq \emptyset$. Then*

$$(I,R)\otimes(J,S) = \bigcup_{(M,c)\in\pi_M(I,R)\cap\pi_M(J,S)} (I,R)|_{(M,c)}\otimes(J,S)|_{(M,c)}$$

The representation of an indexed relation join can be completely expressed in terms of the join of projections as stated in the following corollary.

Corollary 1.

$$(I,R)\otimes(J,S) = \bigcup_{(M,c)\in\pi_M(I,R)\cap\pi_M(J,S)} \pi_{I-M}(I,R)|_{(M,c)}\otimes(M,c)\otimes\pi_{J-M}(J,S)|_{(M,c)}$$

3 No Influence

The idea of no influence of an index set I on an index set J in an indexed relation (K,R) is that the tuple values in (K,R) taken on the domain indexed by I do not constrain or limit in any way the tuple values in (K,R) taken on the domain indexed by J in each block (K,R). For any $(M,c)\in\pi_M(K,R)$, the (M,c) block of (K,R) is defined by $(K,R)|_{(M,c)}$. The tuples on the domain indexed by I in the (M,c) block are given by the restriction $\pi_I(K,R)|_{(M,c)}$. The tuples on the domain indexed by J in the (M,c) block are given by $\pi_J(K,R)|_{(M,c)}$. No influence means that every tuple in $\pi_I(K,R)|_{(M,c)}$ can form a join with every tuple in $\pi_J(K,R)|_{(M,c)}$ and then form a join with (M,c). So if there are m tuples in $\pi_I(K,R)|_{(M,c)}$ and n tuples in $\pi_J(K,R)|_{(M,c)}$, there will be mn tuples in $\pi_J(K,R)|_{(M,c)} \otimes (M,c) \otimes \pi_I(K,R)|_{(M,c)}$ in the case of no influence.

Definition 5. *Let (K,R) be an indexed relation and $\{I,J,M\}$ a non-trivial partition of K. We say conditioned on M, I and J have* **No Influence** *on each other if and only if*

$$\cup_{(M,c)\in\pi_M(K,R)}\pi_I((M,R)|_{(M,c)}) \otimes (M,c) \otimes \pi_J((M,R)|_{(M,c)}) \subseteq (K,R)$$

We also use the language that I has no influence on J.

♠

Figure 1 shows a simple relation $(\{1,2,3\},R$ in which 1 has no influence on 2 given 3. a_1 must be different from a_2 and a_3 must be different from a_4. c_1 must be different from c_2. b_1, b_2, b_3 must all be different and b_4 and b_5 must be different. In the context of the block defined by $(\{3\},c_1)$, i.e. $(\{1,2,3\},R)|_{\{3\},c_1)}$ each occurring value for the variable of index 1 pairs with each occurring value for the variable of index 2.

1	**3**	**2**
a_1	c_1	b_1
a_1	c_1	b_2
a_1	c_1	b_3
a_2	c_1	b_1
a_2	c_1	b_2
a_2	c_1	b_3
a_3	c_2	b_4
a_3	c_2	b_5
a_4	c_2	b_4
a_4	c_2	b_5

Fig. 1. Shows a no influence example. 1 has no influence on 2 given 3. See all possible pairings in the block of c_1 and all possible pairings in the block of c_2.

Based on corollary (1) we have a specialization when both sets of a join decomposition are projections of the same indexed relation.

Proposition 2. *Let $\{I', J', M\}$ be a partition of K. Then,*

$$\pi_{I'\cup M}(K,T) \otimes \pi_{J'\cup M}(K,T) = \bigcup_{(M,c)\in\pi_M(K,T)} \pi_{I'}(K,T)|_{(M,c)} \otimes (M,c) \otimes \pi_{J'}(K,T)|_{(M,c)}$$

The definition of no influence immediately leads to the no influence theorem.

Theorem 2. No Influence Theorem
Let (K, R) be an indexed relation and $\{I', J', M\}$ be a non-trivial partition of K. Then I' has no influence on J' if and only if

$$\pi_{I' \cup M}(K, R) \otimes \pi_{J' \cup M}(K, R) = (K, R)$$

In the database world, the relation join decomposition of the no influence theorem is called a lossless decomposition. The concept of a relation being a join decomposition of two relations is related to the generalization of functional dependency in the database world. The generalization is called multivalued dependency.[2]

Definition 6. *Let (M, R) be an indexed relation and $\{I, J, K\}$ be a partition of M. The multivalued dependency $I \to\to J$ holds in (M, R) if and only if for every $(I \cup K, d), (I \cup K, d') \in \pi_{I \cup K}(M, R)$, where $\pi_I(I \cup K, d) = \pi_I(I \cup K, d')$,*

$$\pi_J(M, R)|_{(I \cup K, d)} = \pi_J(M, R)|_{(I \cup K, d')}$$

♠

Theorem 3. *(Fagin 1977)*
Let (M, R) be an indexed relation. Then $I \to\to J$ holds in (M, R) if and only if $(M, R) = \pi_{I \cup J}(M, R) \otimes \pi_{I \cup K}(M, R)$

We adopt the notation that Dawid[1] uses for conditional independence to indicate no influence.

Definition 7. *Let (K, R) be an indexed relation and $\{I, J, M\}$ be a partition of K. If I has no influence on J we will write $I \perp\!\!\!\perp J \mid M : (K, R)$. If the context (K, R) is clear we will just write $I \perp\!\!\!\perp J \mid M$.*

If the context is the indexed relation (K, R) and if I, J, M are mutually exclusive subsets of K that do not cover K, then we will write $I \perp\!\!\!\perp J \mid M : (K, R)$ to mean

$$\pi_{I \cup M}(K, R) \otimes \pi_{J \cup M}(K, R) \subseteq \pi_{I \cup J \cup M}(K, R)$$

♠

There are some additional properties of the no influence relation: it satisfies the properties of what has been called a semi-graphoid (Pearl and Paz[7] called it a graphoid). A semi-graphoid is a set of triples in which each component is an index set and satisfies the properties stated in the following definition.

Definition 8. *Let $G \subseteq \mathcal{P}(M)^3$. G is a semi-graphoid if and only if*

- *Exclusivity: $(A, B, C) \in G$ implies A, B, C are mutually exclusive sets M*
- *Symmetry: $(A, B, C) \in G$ implies $(B, A, C) \in G$*
- *Decomposition: $(A, B \cup C, D) \in G$ implies $(A, B, C \cup D) \in G$*
- *Weak Union: $(A, B \cup C, D) \in G$ implies $(A, B, D) \in G$*
- *Contraction: $(A, B, D) \in G$ and $(A, C, B \cup D) \in G$ imply $(A, B \cup C, D) \in G$*

4 Factoring

Now we begin the development of the full join decomposition of a relation. Each of the component relations in the join decomposition we call a factor relation. When the join of two factor relations equals a given relation we say that the given relation is factored. The index set on which a factor relation is defined we call its index factor set. In this section we develop the properties relating to this kind of factoring.

The first property is that if the join of two factor relations equals the given relation, then any factor relations defined on supersets of the index factor sets will also factor the given relation.

Proposition 3. *Suppose* $\pi_I(M,R) \otimes \pi_J(M,R) = (M,R)$*. If* $I \subseteq I'$ *and* $J \subseteq J'$*, and* $I' \cup J' = M$*, then* $\pi_{I'}(M,R) \otimes \pi_{J'}(M,R) = (M,R)$*.*

If a given relation is factored into a decomposition of N factor relations, then grouping these factor relations into two possibly overlapping groups will also factor the given relation.

Proposition 4. *Let* $(M,R) = \otimes_{n=1}^{N} \pi_{M_n}(M,R)$ *and* $S \cup T = \{1,\ldots,N\}$*. Then*

$$\pi_{\cup_{s\in S} M_s}(M,R) \otimes \pi_{\cup_{t\in T} M_t}(M,R) = (M,R)$$

Because of the relationship between relation join decomposition and no influence, we can see that in any possibly overlapping grouping of the factor relations, where U is the union of the index factor sets in the first group and V is the union of the index factor sets in the second group, $U - V \perp\!\!\!\perp V - U \mid U \cap V$.

Corollary 2. No Influence Corollary
Let $(M,R) = \otimes_{n=1}^{N} \pi_{M_n}(M,R)$*. Define* $\mathcal{T} = \{T \mid$ *for some* $S \subseteq [N], T = \cup_{s\in S} M_s\}$*, then* $U, V \in \mathcal{T}$ *implies*

$$U - V \perp\!\!\!\perp V - U \mid U \cap V$$

Now we begin to explore the relationship between a join decomposition of a indexed relation (M,R) and the various no influence relationships it implies. Suppose that $M_1,\ldots,M_N$ are the index factor sets of the factoring so that $M = \cup_{n=1}^{N} M_n$, and $I, J \subset M$, $I \cap J = \emptyset$. If I has a nonempty intersection with some index factor set implies that J's intersection with that index factor set is empty, then this forces I to have no influence on J given the remaining index set: $I \perp\!\!\!\perp J \mid M - (I \cup J)$

Proposition 5. *Let* $(M,R) = \otimes_{n=1}^{N} \pi_{M_n}(M,R)$*. Suppose* $I, J \subseteq M$ *and* $I \cap J = \emptyset$*. If* $I \cap M_n \neq \emptyset$ *implies* $J \cap M_n = \emptyset, n = 1,\ldots,N$*, then* $I \perp\!\!\!\perp J \mid M - (I \cup J)$

Definition 9. No Influence Graph and Relation
Let (M,R) *be an indexed relation. The* **no influence graph** G *associated with* (M,R) *is given by* $G = (M,E)$ *where* $E = \{\{i,j\} \mid i \perp\!\!\!\perp j \mid M - \{i,j\}\}$*. The* **no influence relation** F *associated with* (M,R) *is defined by* $F = \{(i,j) \in M \times M \mid i \perp\!\!\!\perp j \mid M - \{i,j\}\}$*. The complement of the no influence graph is called the* **influence graph***.* ♠

The next proposition states that the Cartesian product of the two set differences between the index factor sets of a join decomposition must constitute a block of the no influence relation.

Proposition 6. *Let (M,R) be an indexed relation with no influence relation $F = \{(i,j) \mid i \perp\!\!\!\perp j \mid M - \{i,j\}\}$. If $\pi_A(M,R) \otimes \pi_B(M,R) = (M,R)$, then*

$$(A - B) \times (B - A) \subseteq F$$

Definition 10. Rectangular Block
Let $E \subseteq M \times M$. (A,B) is called a **rectangular block** *of E if and only if $A \times B \subseteq E$.*

♠

Any pair (I,J) of no influence sets of an indexed relation (M,R) constitute a rectangular block of the no influence relation of (M,R).

Proposition 7. *Let (M,R) be an indexed relation and its no influence relation $F = \{(i,j) \mid i \perp\!\!\!\perp j \mid M - \{ij\}\}$. Then $I \perp\!\!\!\perp J \mid M - (I \cup J)$ implies $I \times J \subseteq F$.*

The next result we discuss is concerned with block covers of relations and how they relate to no influence: if $I \perp\!\!\!\perp J \mid M - (I \cup J)$, then each set of the cover is a subset of $M - I$ or a subset of $M - J$.

Definition 11. Block and Block Cover
Let $E \subseteq M \times M$. Then a subset $A \subseteq M$ is called a block of E if and only if $A \times A \subseteq E$. A collection of sets $\{A_1, \ldots, A_N\}$ is called a block cover of E if and only if $\cup_{n=1}^{N} A_n \times A_n = E$.

Clearly, if $\cup_{n=1}^{N} A_n \times A_n = E$, then $M \times M - \cup_{n=1}^{N} A_n \times A_n = E^c$. And if $I \times J \subseteq M \times M - \cup_{n=1}^{N} A_n \times A_n$, then $M_n \subseteq I$ or $M_n \subseteq J$.

Proposition 8. *Let $\{M_n\}_{n=1}^{N}$ be a cover of M. If*

$$I \times J \subseteq M \times M - \cup_{k=1}^{N} M_k \times M_k$$

Then for every $n \in \{1, \ldots, N\}$, $M_n \subseteq M - I$ or $M_n \subseteq M - J$

If $\{M_n\}_{n=1}^{N}$ is a block cover of the influence relation E. Then $M \times M - \cup_{n=1}^{N} M_n \times M_n$ is a block cover of the no influence relation E^c. And if (M,R) is factored with index factor sets A and B, then $M_n \subseteq A$ or $M_n \subseteq B$.

Proposition 9. *Let (M,R) be an indexed relation and $\{M_n\}_{n=1}^{N}$ be a cover for M satisfying*

$$\{(i,j) \mid i \perp\!\!\!\perp j \mid M - \{i,j\} = M \times M - \cup_{n=1}^{N} M_n \times M_n\}$$

If $\pi_A(M,R) \otimes \pi_B(M,R) = (M,R)$, then for every $n \in \{1, \ldots, N\}$, $M_n \subseteq A$ or $M_n \subseteq B$.

Definition 12. Clique
Let $G = (M,H)$ be a graph. A subset $I \subseteq M$ is called a **clique** *of G if and only if*

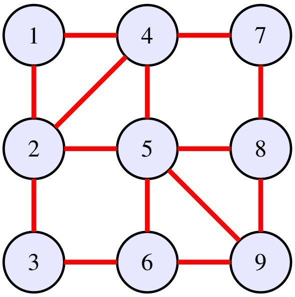
1. *$i, j \in I$ implies $\{i,j\} \in H$*

2. *if $J \supseteq I$ and $i, j \in J$ implies $\{i, j\} \in H$, then $J = I$*

Let $P \subseteq M \times M$. A subset $I \subseteq M$ is called a **clique** *of P if and only if*

1. *Block: $I \times I \subseteq P$*
2. *Maximal: $J \supseteq I$ and $J \times J \subseteq P$ implies $J = I$*

Figure 2(b) shows what happens when an indexed relation is join decomposed into factors whose index factor sets are the cliques of the influence graph. When the factor relations are grouped into two factor groups, each factor must be in at least one of the groups. Each of the two group index factor sets is a superset of the union of the factor sets they contain.



(a) Influence Graph

$$\mathbf{M} = \{\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7},\mathbf{8},\mathbf{9}\}$$
$$(\mathbf{M},\mathbf{R}) = \otimes^{\mathbf{8}}_{\mathbf{n=1}} \pi_{\mathbf{M_n}}(\mathbf{M},\mathbf{R})$$
$$\mathbf{I} = \{\mathbf{1},\mathbf{2}\},\ J = \{\mathbf{8},\mathbf{9}\},\ \mathbf{M} - (\mathbf{I} \cup \mathbf{J}) = \{\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7}\}$$
$$\mathbf{I} \cap \mathbf{J} = \emptyset, \mathbf{I} \perp\!\!\!\perp \mathbf{J} \mid \mathbf{M} - (\mathbf{I} \cup \mathbf{J})$$
$$\mathbf{I} \cup (\mathbf{M} - (\mathbf{I} \cup \mathbf{J})) = \mathbf{M} - \mathbf{J}$$
$$\mathbf{J} \cup (\mathbf{M} - (\mathbf{I} \cup \mathbf{J})) = \mathbf{M} - \mathbf{I}$$
$$(\mathbf{M},\mathbf{R}) = \pi_{\mathbf{M-I}}(\mathbf{M},\mathbf{R}) \otimes \pi_{\mathbf{M-J}}(\mathbf{M},\mathbf{R})$$
$$\mathbf{I} \times \mathbf{J} \subseteq \mathbf{M} \times \mathbf{M} - \cup^{\mathbf{8}}_{\mathbf{n=1}} \mathbf{M_n} \times \mathbf{M_n}$$
$$\mathbf{S} = \{\mathbf{n} \mid \mathbf{M_n} \subseteq \mathbf{M} - \mathbf{J}\} = \{\mathbf{1},\mathbf{2},\mathbf{5},\mathbf{6},\mathbf{7}\}$$
$$\mathbf{T} = \{\mathbf{n} \mid \mathbf{M_n} \subseteq \mathbf{M} - \mathbf{I}\} = \{\mathbf{3},\mathbf{4},\mathbf{6},\mathbf{7},\mathbf{8}\}$$
$$\mathbf{M} - \mathbf{J} = \mathbf{M_1} \cup \mathbf{M_2} \cup \mathbf{M_5} \cup \mathbf{M_6} \cup \mathbf{M_7} = \cup_{\mathbf{s \in S}} \mathbf{M_s}$$
$$\mathbf{M} - \mathbf{I} = \mathbf{M_3} \cup \mathbf{M_4} \cup \mathbf{M_6} \cup \mathbf{M_7} \cup \mathbf{M_8} = \cup_{\mathbf{t \in T}} \mathbf{M_t}$$

Clique		$\mathbf{M} - \mathbf{J}$	$\mathbf{M} - \mathbf{I}$
Symbol	**Cliques**	$\{\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7}\}$	$\{\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7},\mathbf{8},\mathbf{9}\}$
M_1	$\{1,2,4\}$	1	0
M_2	$\{2,4,5\}$	1	0
M_3	$\{5,6,9\}$	0	1
M_4	$\{5,8,9\}$	0	1
M_5	$\{2,3\}$	1	0
M_6	$\{3,6\}$	1	1
M_7	$\{4,7\}$	1	1
M_8	$\{7,8\}$	0	1

(b) Clique Containment Table

Fig. 2. Shows relationship between cliques and no influence pair of sets $I = \{1,2\}, J = \{8,9\}$

Theorem 4. No Influence Decomposition Theorem
If $(M,R) = \otimes_{n=1}^{N}\pi_{M_n}(M,R)$ *and* $I \times J \subseteq M \times M - \cup_{n=1}^{N} M_n \times M_n$*, then* $I \perp\!\!\!\perp J|M - (I \cup J)$*.*

The collection of cliques of an influence graph of an indexed relation (M,R) cover M. There is a relationship that can be defined between covers of a set. Let C be a collection of covers of a set M. Define a binary relation $\mathcal{P} \subseteq C \times C$ by

$$\mathcal{P} = \{(C_1,C_2) \in C \times C \mid C \in C_1 \text{ implies that for some } D \in C_2, C \subseteq D\}$$

When $(C_1,C_2) \in \mathcal{P}$ we say that C_1 is a refinement of C_2 or equivalently, that C_2 is a coarsening of $(C_1$.

It is easy to see that the binary relation $\mathcal{P}$ is reflexive and transitive and is therefore a pre-order. However, if the cover has the property that no two sets in the cover can be in a subset relation, then the binary relation $\mathcal{P}$ is antisymmetric and therefore a partial order.

Definition 13. *Let C be a cover of a set M. C has the* **Subset Property** *if and only if $A,B \in C$ and $A \subseteq B$ imply $A = B$.*

Notice that any collection of cliques of a graph constituting a cover has the subset property.

Proposition 10. *Let C be a collection of covers of a set M where each cover in C has the subset property. Define a binary relation $\mathcal{P} \subseteq C \times C$ by*

$$\mathcal{P} = \{(C_1,C_2) \in C \times C \mid A \in C_1 \textit{ implies that for some } B \in C_2, A \subseteq B\}$$

Then $\mathcal{P}$ is a partial order.

Definition 14. *Let $\mathcal{P}$ be a partial order on a set C. C_M is called a* **Largest Element** *if and only if for every $C_1 \in C$, $(C_1,C_M) \in \mathcal{P}$.*

Proposition 11. *Largest elements of a partial order are unique.*

Now it is clear that if $\mathcal{P}$ is defined on the collection of all block covers of the influence graph having the subset property, the cover defined by the set of cliques of the influence graph is the unique largest element. Since it is the case that the larger the index sets are in a join composition, the smaller the resulting join. Therefore, the smallest resulting join composition is the join composition resulting from projections using index sets that are the cliques of the influence graph. Therefore, if the collection C is the collection of the cliques of the influence graph of (M,R), then it must be the case that for any other block cover $\mathcal{B}$ of the influence graph of (M,R),

$$\otimes_{C \in C}\pi_C(M,R) \subseteq \otimes_{B \in \mathcal{B}}\pi_B(M,R)$$

Even though $\otimes_{C \in C}\pi_C(M,R)$ produces the smallest possible decomposition, it still may even properly contain (M,R). So the strongest statement that can be made is $(M,R) \subseteq \otimes_{C \in C}\pi_C(M,R)$.

Definition 15. Irreducibility
Let (M, R) be an indexed relation. (M, R) is called an **irreducible indexed relation** *if and only if for every possible distinct $\{M_1, M_2\}$ cover of M*

$$(M, R) \neq \pi_{M_1}(M, R) \otimes \pi_{M_2}(M, R)$$

A join decomposition $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$ is called an **irreducible decomposition** *if and only if for each $n \in [N]$, $\pi_{M_n}(M, R)$ is an irreducible indexed relation.* ♠

It immediately follows from the definition of irreducibility that an indexed relation (M, R) is irreducible if and only if its no influence relation is empty. So it is certainly the case that for any $\{I, J\}$ cover of M it is not the case that $I \perp\!\!\!\perp J \mid M - (I \cup J)$. In particular this implies that for any $i, j \in M$, it is not the case that $i \perp\!\!\!\perp j \mid M - \{i, j\}$. Thus in an irreducible join decomposition $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$, each of the factor projections, $\pi_{M_n}(M, R)$, is a maximal projection, maximal in the set M_n, having an empty no influence relation. So if we define the influence graph $G = (M, E)$ arising from the indexed relation (M, R), by $E^c = \{\{i, j\} \mid i \perp\!\!\!\perp j \mid M - \{i, j\}\}$, then $\{M_n\}_{n=1}^{N}$ are the cliques of G.

Definition 16. Complete Irreducible Decomposition
An irreducible decomposition $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$ is called a **complete irreducible decomposition** *if and only if C a clique of the influence graph of (M, R) implies $C \in \{M_n\}_{n=1}^{N}$.* ♠

In arithmetic, irreducible numbers are numbers that cannot be further factored. Such numbers are called prime. We might think that like in arithmetic, a prime factor decomposition of relations is unique up to the order of the factors. But for relation decomposition this is not the case. There can be multiple irreducible factor decompositions of the same composite relation.

When $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$, we say that the M_n sets are the index factor sets. If for the same relation $\pi_A(M, R) \otimes \pi_B(M, R) = (M, R)$ we say that A and B are group factor sets. As stated in the next proposition, the group factor sets are bounded above by unions of the index factor sets.

Proposition 12. *Let $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$ be a complete irreducible join decomposition. Let $\{A, B\}$ be a cover of M satisfying $A - B \neq \emptyset$ and $B - A \neq \emptyset$ and $\pi_A(M, R) \otimes \pi_B(M, R) = (M, R)$. Then there exists $S, T \subseteq [N]$ such that $S \cup T = [N]$ and*

$$A \subseteq \cup_{s \in S} M_s$$
$$B \subseteq \cup_{t \in T} M_t$$

and

$$(\pi_{\cup_{s \in S} M_s}(M, R)) \otimes (\pi_{\cup_{t \in T} M_t}(M, R)) = (M, R)$$

Recall that the decomposition proposition implies that if $I \perp\!\!\!\perp J \mid M - (I \cup J)$, then for any subsets $I' \subseteq I$ and $J' \subseteq J$, $I' \perp\!\!\!\perp J' \mid M - (I' \cup J')$. So a no influence pair of sets can be decomposed into a smaller pair of sets having no influence. But the inverse does not hold. It does not follow that if $I \perp\!\!\!\perp J \mid M - (I \cup J)$ and $I \perp\!\!\!\perp K \mid M - (I \cup K)$, then we can build up the no influence pair of sets and obtain $I \perp\!\!\!\perp J \cup K \mid M - (I \cup J \cup K)$. However, if the indexed relation (M, R) has a decomposition $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$ and the factor sets of the decomposition satisfy $\{(i, j) \mid i \perp\!\!\!\perp j \mid M - \{i, j\}\} = M \times M - \cup_{n=1}^{N} M_n \times M_n$, then

$$I \perp\!\!\!\perp J \mid M - (I \cup J) \text{ and } I \perp\!\!\!\perp K \mid M - (I \cup K) \text{ imply } (I \perp\!\!\!\perp J \cup K \mid M - (I \cup J \cup K)$$

Proposition 13. Intersection Property
Let (M, R) be an indexed relation and its no influence relation

$$F = \{(i, j) \mid \quad i \perp\!\!\!\perp j \mid M - \{ij\} \ \}$$

If $\{M_n\}_{n=1}^{M}$ is a cover of M satisfying

$$\begin{aligned} (M, R) &= \otimes_{n=1}^{N} \pi_{M_n}(M, R) \\ F &= M \times M - \cup_{n=1}^{N} M_n \times M_n \end{aligned}$$

then, $I \perp\!\!\!\perp J \mid M - (I \cup J)$ and $I \perp\!\!\!\perp K \mid M - (I \cup K)$ imply $I \perp\!\!\!\perp (J \cup K) \mid M - (I \cup J \cup K)$

Thus an indexed relation (M, R) that has factors satisfies the intersection property and this makes the collection of triples

$$\mathcal{G} = \{(A, B, D) \in \mathcal{P}(M) \mid A \ \perp\!\!\!\perp B \mid C : (M, R)\}$$

a graphoid.

Figure 3 shows how conditional no influences can be seen from the maximally complete bipartite subgraphs of the no influence graph G_N of a relation (M, R) whose factors are the cliques of its influence graph. If (I, J) is pair of sets associated with a maximally complete bipartite subgraph, then $I \ \perp\!\!\!\perp J \mid M - (I \cup J)$ and in the influence graph I is separated from J by $M - (I \cup J)$.

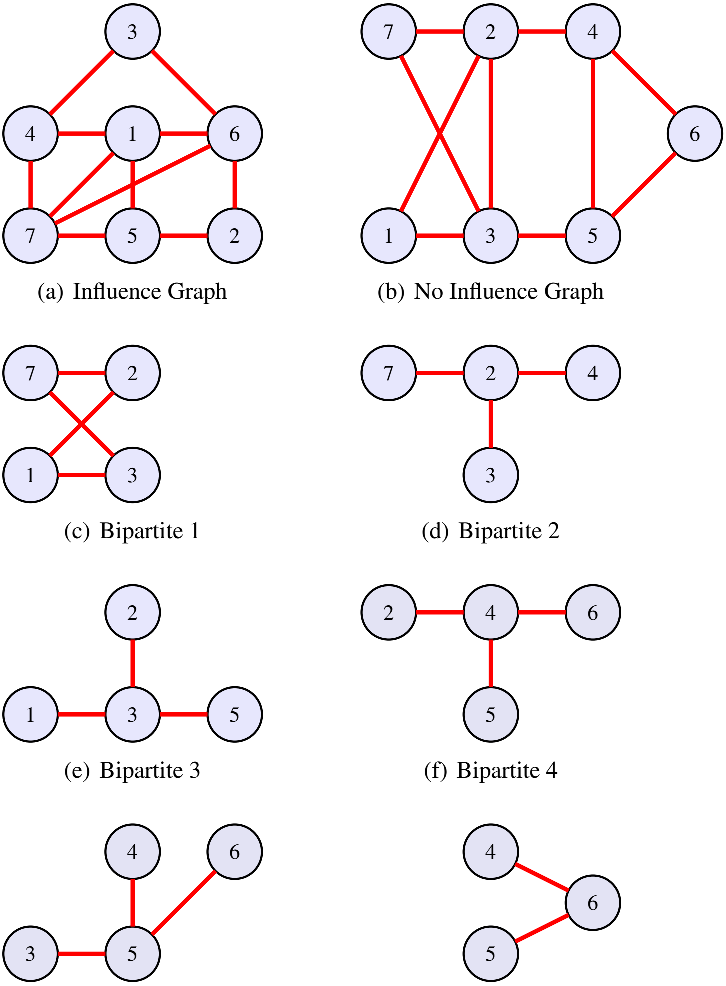
5 Forming the Decomposition

Let (M, R) be an indexed relation and $i, j \in M$. If $i \ \perp\!\!\!\perp \ j \mid M - \{i, j\}$, then it would be the case that $(M, R) = \pi_{M-\{i\}}(M, R) \otimes \pi_{M-\{j\}}(M, R)$. Since it always is the case that $(M, R) \subseteq \pi_{M-\{i\}}(M, R) \otimes \pi_{M-\{j\}}(M, R)$, it is reasonable to base a coefficient of no influence on $|\pi_{M-\{i\}}(M, R) \otimes \pi_{M-\{j\}}(M, R) - (M, R)|$. Normalized,

$$\rho_{ij} = \frac{|\pi_{M-\{i\}}(M, R) \otimes \pi_{M-\{j\}}(M, R) - (M, R)|}{|\pi_{M-\{i\}}(M, R) \otimes \pi_{M-\{j\}}|}$$

takes the value 0 when $i \ \perp\!\!\!\perp \ j \mid M - \{i, j\}$ and takes a value greater than 0 otherwise.

To determine the statistical significance of ρ_{ij} we do a permutation test. Thinking of the tuples of (M, R) arranged as a matrix with each row representing one tuple, we can take the column associated with the index i and randomly shuffle all the values of the column. Similarly, we can take the column associated with the index j and randomly shuffle all the values of that column. Call the permuted relation (M, R^1). Let us do this Z times forming the permuted relations $(M, R^1), \ldots, (M, R^Z)$. Associated with each permuted relation (M, R^z) is its coefficient of no influence ρ^z_{ij}. We define the p-value



(a) Influence Graph

(b) No Influence Graph

(c) Bipartite 1

(d) Bipartite 2

(e) Bipartite 3

(f) Bipartite 4

(g) Bipartite 5

(h) Bipartite 6

Fig. 3. Shows an example no influence graph and its complement, the influence graph. The maximally complete bipartite subgraphs of the no influence graph are shown. In the influence graph: (Bipartite 1) $\{1,7\}$ is separated from $\{2,3\}$ by $\{4,5,6\}$. (Bipartite 2) $\{2\}$ is separated from $\{3,4,7\}$ by $\{1,5,6\}$. (Bipartite 3) $\{3\}$ is separated from $\{1,2,5\}$ by $\{4,6,7\}$. (Bipartite 4) $\{4\}$ is separated from $\{2,5,6\}$ by $\{1,3,7\}$. (Bipartite 5) $\{5\}$ is separated from $\{3,4,6\}$ by $\{1,2,7\}$. (Bipartite 6) $\{6\}$ is separated from $\{4,5\}$ by $\{1,2,3,7\}$.

associated with the Null Hypothesis that $i \perp\!\!\!\perp j \mid M - \{i, j\}$ by

$$p - value_{ij} = \frac{|\{z \mid \rho^z_{ij} > \rho_{ij}\}| + \frac{1}{2}|\{z \mid \rho^z_{ij} = \rho_{ij}\}|}{Z + 1}$$

We reject the hypothesis that $i \perp\!\!\!\perp j \mid M - \{i, j\}$ at significance level α when $p - value_{ij} < \alpha$. Now we can define the relation $F \subseteq M \times M$ by

$$F = \{(i, j) \in M \times M \mid p - value_{ij} < \alpha\}$$

Associated with the collection of cliques C of F, we can form the relation decomposition

$$\otimes_{A \in C} \pi_A(M, R)$$

6 Concluding Discussion

Let (M, R) be the observed data relation and let $\otimes_{A \in C} \pi_A(M, R)$ be its relation decomposition. Then we can asert that for the variables indexed in each $A \in C$, the variables are directly jointly dependent and the action of the joint dependence is given by the tuples in $\pi_A(M, R)$. Note that because of the dependency, $\pi_A(M, R) \subset \times_{a \in C} L_a$ Proper subsets indicate constraints and constraints define dependency.

For any $A, B \in C$, such that $A \cap B = \emptyset$, we can assert that the variables indexed by A have no influence on the variables indexed by B given the remaining variables indexed by $M - (A \cup B)$. Further, we can define the full collection of explanatory causal models which are consistent with the relation decomposition. These causal models can be feedforward like Bayesian Networks or can be feedback causal models. Each such model constitutes a causal explanation of the process by which the observed data (M, R) arises.

References

1. Phillip Dawid, A.: Conditional independence in statistical theory. Journal of the Royal Statistical Society Serial B 41, 1–31 (1979)
2. Fagin, R.: A new normal form for relational databases. ACM Transactions on Database Systems 2(3), 262–278 (1977)
3. Fagin, R., Vardi, M.: The theory of data dependencies - a survey. Proceedings of Symposia in Applied Mathematics 34, 19–71 (1986)
4. Galles, D., Pearl, J.: Axioms of causal relevance. Artificial Intelligence 97, 9–43 (1997)
5. Hausman, D., Woodward, J.: Independence, invariance, and the causal markov condition. British Journal of Philosophy 50, 521–583 (1999)
6. Lauritzen, S.: Graphical Models. Oxford University Press, New York (1996)
7. Pearl, J., Paz, A.: On the logic of representing dependencies by graphs. In: Proceedings 1986 Canadian AI Conference, Montreal, Canada, pp. 94–98 (1986)
8. Shafer, G., Shenoy, P., Mellouli, K.: Propagating belief functions in qualitative markov trees. International Journal of Approximate Reasoning, 349–400 (1987)
9. Spohn, W.: Stochastic independence, causal independence, and shieldability. Journal of Philosophical Logic 9, 73–99 (1980)

Using Turning Point Detection to Obtain Better Regression Trees

Paul K. Amalaman, Christoph F. Eick, and Nouhad Rizk

Department of Computer Science,
University of Houston, Houston, Texas 77204-3010, USA
{pkamalam,ceick,nrizk}@uh.edu

Abstract. The issue of detecting optimal split points for linear regression trees is examined. A novel approach called Turning Point Regression Tree Induction (TPRTI) is proposed which uses turning points to identify the best split points. When this approach is used, first, a general trend is derived from the original dataset by dividing the dataset into subsets using a sliding window approach and a centroid for each subset is computed. Second, using those centroids, a set of turning points is identified, indicating points in the input space in which the regression function, associated with neighboring subsets, changes direction. Third, the turning points are then used as input to a novel linear regression tree induction algorithm as potential split points. TPRTI is compared in a set of experiments using artificial and real world data sets with state-of-the-art regression tree approaches, such as M5. The experimental results indicate that TPRTI has a high predictive accuracy and induces less complex trees than competing approaches, while still being scalable to cope with larger datasets.

Keywords: Prediction, Linear Regression Tree, Model Tree, Turning Point Detection.

1 Introduction

Regression trees are widely used machine learning techniques for numerical prediction. Among the regression trees, we may distinguish those that associate a constant to each leaf node, for instance CART [2], and those that fit a less trivial model to each leaf node. Among the latter, we further distinguish a class of regression trees that fit a linear model to each leaf node, such as linear regression trees. Linear regression tree induction has been intensely researched. One of the first approach, M5[12], induces the tree using a CART-like splitting decision which is a binary split based on mean values and uses constant regression functions in the nodes; the attribute that best reduces the variance in the nodes is chosen for the split, and its mean value is selected as the split value. After the tree is fully developed M5 fits a linear model to each leaf-node during pruning phase. This splitting method is also used by many regression approaches which associate non-constant models with the leaf-nodes. However, in conjunction with non-constant regression models, using mean values of attributes as split points and using variance reduction as an objective function does not necessarily obtain the best model [7].

To address this issue Karalic proposed RETIS [7] which fits a linear model in each node and uses minimization of the residual sum of squared errors (RSS) instead of the

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 325–339, 2013.

variance as splitting function. However, although the approach yields significantly better accuracy and smaller regression trees than M5 [12], it has been labeled "intractable" because to find the best pair {split attribute/split value} all potential split values for each input attribute need be evaluated, making this approach too expensive even for medium-sized datasets. Therefore, many more scalable algorithms for inducing linear regression trees have been proposed [1, 6, 7, 11, 12, 15, 5, 17] which rely on heuristics, sampling, approximations, and different node evaluation functions to reduce the computational cost of the RETIS algorithm.

Detecting turning points which indicate locations where the general trend changes direction can be useful in many applications. In this paper, we propose a new approach for Linear Regression Tree construction called Turning Point Regression Tree Induction (TPRTI) that infuses turning points into a regression tree induction algorithm to achieve improved scalability while maintaining accuracy and low model complexity. TPRTI induces decision trees using the following procedure. First, a general trend is derived from the dataset by dividing the dataset into a sequence of subsets of equal size using a sliding window, and by associating a centroid with each subset. Second, using those centroids, a set of turning points is identified, indicating points in the input space in which the piecewise linear function associated with neighboring subsets changes direction. Finally, the identified turning points are used in two novel top down linear regression tree induction algorithms as potential split points. The two algorithms which are called TPRTI-A and TPRTI-B are discussed in section 2.

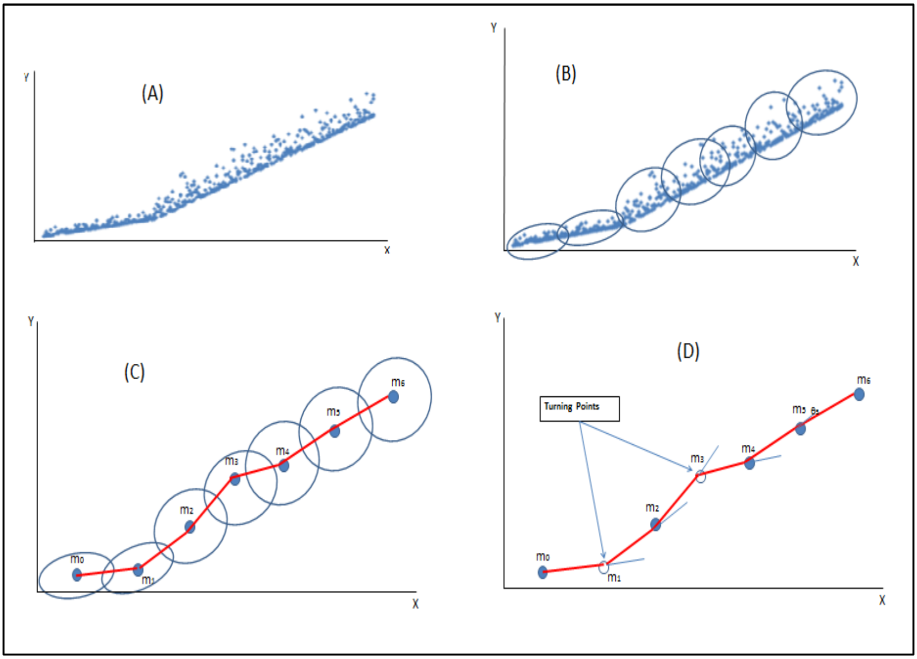


Fig. 1. Panel (A) represents hypothetical dataset in a plane (x,y). (B) The dataset is subdivided into overlapping subsets of equal size. (C) Centroids are joined by straight lines to form a general trend in the dataset. In panel (D) m_1 and m_3 are detected as turning points.

Figure 1 illustrates our proposed approach to detecting turning points. The input dataset is shown in panel (A). In panel (B) the dataset is sorted by the input attribute and divided into subsets of equal size using a sliding window approach. In panel (C) the general trend in the dataset is derived by connecting the centroids of neighboring subsets, obtaining a piecewise linear function. Finally in panel (D), points m1 and m3 are selected as turning points as they exhibit sharp turns in the piecewise linear function. The selected turning points are later fed to a linear regression tree algorithm as potential split points. Algorithm 1 gives the pseudo code of turning point detection algorithm.

Algorithm 1: Determining turning points

```
Inputs
  plane ($X_k$,Y) where $X_k$'s  are input attributes (k=1,2,..p)
  $X_k$: real valued discrete or continuous variable
  y: target variable
Outputs
  Turning points set

Project dataset onto each plane ($X_k$,Y)
For each plane ($X_k$,Y)
  Sort the data per attribute $X_k$
  if $X_k$ discrete attribute then
    Compute centroids for each distinct value of $X_k$
    Label centroids as turning points
  else
    Use a sliding window of fixed size subsets for the input
    attribute to split the data into neighboring subsets from
    small values of $X_k$ to the largest value of $X_k$
    Determine general trends by computing the centroids of
    each subset and connect them to obtain piecewise linear
    functions
    Identify turning points by analyzing the angle $\theta$ between
    neighboring subsets of the piecewise linear function
  Output the set of turning points
```

The main contributions of the paper include:

1. A novel approach for turning point detection which relies on window sub-setting is introduced.
2. Two novel linear regression tree induction algorithms called TPRTI-A and TPRTI-B which incorporate turning points into the node evaluation are introduced.
3. State of the art linear regression tree algorithms are compared with each other and with TPRTI-A and TPRTI-B for a challenging benchmark involving 12 datasets.
4. The experimental results indicate that TPRTI is a scalable algorithm that is capable of obtaining a high predictive accuracy using smaller decision trees than other approaches.

The rest of the paper is organized as follows. Section 2 contains the description of our proposed methods for linear regression trees. In section 3 we show results of experimental study and we conclude in Section 4.

Table 1. Notation used in the remaining of the paper

K	User defined overlapping parameter characterizing the number of examples pertaining to two consecutive subsets.
S	Size of each subset
θ	Angle at a centroid
β	User-defined threshold angle such that if cosθ < cosβ then the centroid with angle θ is a turning point
Stp_{XY}	Set of turning points in the XY-plane
Stp	Union of all Stp_{XY} (for all planes)
Stp_left	Turning Points set for left sub-node such that Stp=Stp_left U Stp_right
Stp_right	Turning Point set for right sub-node such that Stp=Stp_left U Stp_right
RSS	Residual Sum of Squared errors

2 The Tprti Approach

Linear regression algorithms can be divided into three groups: The first group fits a constant regression model to each intermediate node. M5 [12] is an example of this approach. The second group fits a more complex regression model to each node; usually at least a model is needed per input attribute. RETIS [7] is one such example since it fits multiple regression models per input attribute; one regression model for each distinct value of each input attribute. The third group uses linear regression models in the leaves, but at each node transforms the regression problem into a classification problem in order to use more efficient node evaluation function. SECRET [5], GUIDE [9], and SUPPORT [3], are examples of such approaches. Each group can be distinguished by how well it performs with respect to model accuracy, model complexity, and runtime complexity, which is how scalable the approach is when data sizes increase. To evaluate a node, the first group is computationally more efficient since the evaluation function in each node is the simplest; however it often yields complex models and low accuracy. The second group has a much better accuracy, and model complexity but it comes at the expense of a higher cost node evaluation. Between both ends of the spectrum lies the third group. The major limitation of the first group of algorithms is illustrated next.

2.1 Illustrating the Limitations of the Variance Based-Approaches

Many widely used first group algorithms use variance as node evaluation function and we will refer to them as “variance-based approaches”. M5 [12] is one such an example. As pointed out in [7] variance based algorithms have a fundamental imperfection to induce linear regression trees that are optimal because the variance is not a good node evaluation criterion. To illustrate this point let us consider a dataset called dataset#2 whose instances were generated with respect to the function defined below without using any noise, assuming a uniform distribution of input values x1 in [0,250]:

$$x1 \in [0{,}250] \text{ and } x2{=}0, \text{ and } y \in R$$

$$\begin{cases} y = x1; & if\ 0 \le x1 < 50 \\ y = 100 - x1; & if\ 50 \le x1 < 250 \end{cases}$$

It is clear that the best split value is located at x1=50. The variance based approaches, like M5, will split at x1=125 (the mean value of x1) leading to the necessity to introduce further possible split to the data in the left node. RETIS however, which tests each distinct value of the input variable x1, selects the optimal split value 50.

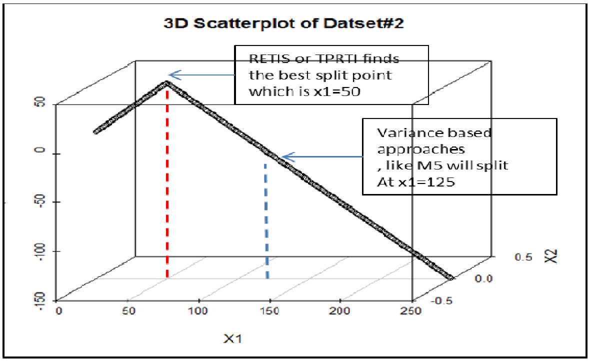


Fig. 2. Illustrating the imperfection of M5

As pointed out earlier, many second group approaches like RETIS yield more accurate, and shorter linear regression trees but do not scale well to large dataset. Likewise the third group approaches resolve the scalability issue of RETIS but at a cost to accuracy and/or model complexity.

In this paper we use the term "look-ahead RSS" to mean the following: First, the dataset associated with current node is split into two sub-sets; one pertaining to each sub-node. Second, each sub-node is fitted with a linear regression model and the Residual Sum of Squared errors (RSS) are computed for both nodes. Next, a weighted sum of both RSS is computed. This is done for all potential split pairs {attribute variable, attribute value} available. One such split is retained and the rest discarded. Likewise, we use the term "split point" to designate the pair {split attribute, split attribute value}. We also use it to refer to the actual point in the plane, or point in the input space. When used as a point in a plane it refers to the pair (x_q ,y) where x_q is one of the input variables, q ∈ {1,..p}. When used to mean a point in the input space, it refers to a tuple; (**x**,y) where **x** is an input vector (x_1, ..x_p) and y the associate response. Key notations used in the remaining of the paper are provided in table1. Algorithm1 presents the turning point detection approach. We provide next, in section 2.2 detailed description of how turning points are computed.

2.2 Centroids and Turning Points Computation

First, a general trend is derived from the dataset by sorting the examples based on the input attributes, by dividing the dataset into subsets of equal size using a sliding window, and by associating a centroid with each subset. Second, using those centroids, a set of turning points is identified, indicating points in the input space in which the piecewise linear function—which was obtained by connecting centroids of neighboring subsets—significantly changes direction.

The input attributes to the algorithm are real valued attributes that are either discrete or continuous. Lines 8, 9, and 10 in algorithm 1 project the dataset onto the p x_ky-planes (k=1,..,p). For the remaining of the lines, line 11 to 17 we consider one plane at a time. Line 10 ensures that the dataset associated with the plane is sorted with respect to the input attribute. Lines 11, 12, and 13 treat the case of discrete attributes. First, the algorithm queries the distinct values of the discrete attribute. It then computes the centroids for each attribute. Next, each centroid is labeled turning point.

Lines 14 to 17 treat the case where the input attribute is continuous. There are three user-defined parameters K, S, β that need to be set. Let assume that a centroid has angle θ. β is a user-defined angle such that if $\cos\theta < \cos\beta$ then the centroid is a turning point. K is the overlapping parameter characterizing the number of examples pertaining to two consecutive subsets. S is the size of each subset. In line 15 subsets of equal size S are created using the sorted dataset as follows: S_0 is composed of the first S examples. At any given step i, (i>0), the examples in subset S_i are determined by dropping the first K examples in S_{i-1} and by adding the next K new examples. When K=S, the subsets are disjoint. When 0<K< S the subsets overlap. In line 17 turning points are computed for each plane by analyzing the angle at each centroid.

2.3 Node Evaluation

We introduce TPRTI-A, which is a mixture of first group and second group approach in that its node evaluation avoids exhaustive search by evaluating only a supplied list of turning points. We also introduce TPRTI-B which is a mixture of all 3 groups in that it uses a two-step node evaluation. It avoids exhaustive search by evaluating only a supplied set of turning points. It first fits a model to current node, and uses a simple evaluation function which is the distance of each turning point to the fitted model, to select the turning point which distance is the largest. TPRTI-A and TPRTI-B differ by their node evaluation function. They both use as input, a set of predetermined turning points.

2.3.1 Node evaluation for TPRTI-A

The first approach, TPRTI-A, evaluates all turning points by a look-ahead strategy and selects the one that yields the minimum RSS.

Algorithm 2: Node evaluation for TPRTI-A

```
Inputs
  $Stp_{XY}$: Set of turning points in the XY-plane
  Stp: Union of all $Stp_{XY}$ (for all planes)
  $TP_{XY}$(x,y): Turning point in the XY-plane with
  coordinate (x,y)
 If stopping criteria is reached then
   Return
 For each $Stp_{XY}$ in Stp
  For each $TP_{XY}$(x,y)in $Stp_{XY}$
      Split data in $S_{Left}$ and $S_{right}$
      Compute look-ahead weighted RSS
 Select the turning point (x ,y )that minimizes
 Weighted RSS for the split
 Split Stp into Left_Stp, and Right_Stp based on x
```

2.3.2 Node Evaluation for TPRTI-B

The second approach, TPRTI-B, is a multi-step evaluation approach. With this approach, first the current node is fitted with a model and the distances of the turning points (actual tuples) to the fitted model are computed. The turning point for which the distance to the model is the largest is selected as split point. Next, each coordinate (value of input attributes) of the split point needs be evaluated by a look-ahead RSS minimization method to determine the best pair {split variable, split value}. That is, in contrast to TPRTI-A, only a single split value per input attribute and not a set of split values is considered; reducing runtime complexity. Figure 3 illustrates the general idea. In figure 3, the dotted line represents a linear model F fitted to current node. In this hypothetical node, the turning point set has three turning points TP1, TP2 and TP3. We assume d1 < d2 and d3 < d2. Hence TP2(x_{21},x_{22},y_2) is chosen as split point. Its coordinates x_1= x_{21},and x_2= x_{22} need next, to be evaluated to select the pair {split variable, split value} that best minimizes RSS. Algorithm 3 summarizes the concept.

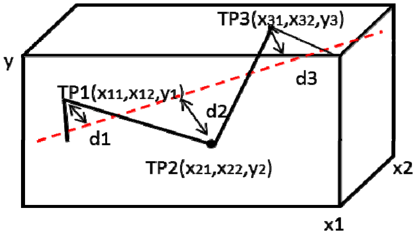


Fig. 3. To illustrate the node evaluation of TPRTI-B; the dotted line is a model fitted to current node. TP2 is selected as split point because it is the turning point further away from the fitted model F.

Algorithm 3: Node evaluation for TPRTI-B

```
1  If stopping criteria is reached then
2    Return
3  Fit current node with linear model F
4  For each turning point tp in Stp
5    Compute distance to F
6  Select tp_max the point with the largest distance to F
7  For each input coordinate of tp_max
8   Split data in $S_{Left}$ and $S_{right}$
9   Compute look-ahead weighted RSS
10 Select the turning point that minimizes weighted RSS as split
    point
11  split Stp in Left_Stp, and Right_Stp
```

In line 3 a model F is fitted to the node. Lines 4, 5, and 6 select the point with the largest distance to F. Line 7 to 11 are similar to what was presented for TPRTI-A in 2.3.1 except that here the points are represented in the actual space; not in a plane. Hence the turning points set Stp, although computed as presented previously in algorithm 1, is formed of points with their coordinates in the data space. This is so because line 3 needs to compute the distance to F in the entire space.

2.4 Runtime Complexity

We compute the runtime cost to evaluate a node. Let N be the total number of training examples in the node, m the number of subsets, p the number of input attributes, and t total number of turning points. Assuming N>>p, evaluating each split candidate costs $O(p^2N)$; If TPRTI-A method is used, all the turning points are evaluated and the cost to evaluate a node is $O(p^2Nt)$. If TPRTI-B is used the distance of each turning point in the data space to the fitted curve cost O(p) which leads to O(pt) for the t turning points. $O(p^2N)$ is needed to evaluate each of the p input attribute, and $O(p^2N)$ is as well needed to fit a model to current node; obtaining: $O(pt+p^2N+Np^3) = O(p^2N(p+1))$. With M5, the split point is the mean point of each p variable. Hence, t=p obtaining $O(p^3N)$; In the worst case, RETIS will test each value of each variable leading to t=pN ; thus $O(p^3N^2)$. TPRTI-A worse case happens when each centroid is a turning point; which leads to t=pm hence $O(p^3Nm)$. Table 2 summarizes the runtime complexity of each approach.

Table 2. Node Runtime complexity of TPRTI in comparison to M5 and RETIS

	RETIS	TPRTI-A	TPRTI-B	M5
Runtime complexity	$O(p^3N^2)$	$O(p^3Nm)$	$O(p^2N(p+1))$	$O(p^3N)$

2.5 Stopping Criteria

RETIS, M5 and TPRTI share the following stopping criteria

- The algorithm stops if the number of examples in the node is less than a minimum number set by user.

- The algorithm stops if the subsequent split sub-nodes do not improve the error function more than a minimum value set by user.
- However, TPRTI has an additional stopping condition: The algorithm may terminate if there is no more turning point in the node dataset to evaluate.

3 Empirical Evaluation

In this section results of extensive experimental study of TPRTI-A and TPRTI-B are provided. We compare both algorithms with each other and against the well-known M5 [12], SECRET [5], GUIDE [9], and RETIS [7] algorithms in term of accuracy, scalability and model complexity. The experimental result published in [5], for GUIDE and SECRET, two state-of-the-art scalable linear regression tree induction algorithms are used in this study for comparison. The GUIDE and SECRET algorithms were built to be both fast and accurate. Model complexity results for previously used datasets were not available for comparison. We performed all the experiments reported in this paper on a 64-bit PC i7-2630 CPU at 2Ghz running Windows 7. Datasets and extended version of this paper are available at [19].

3.1 Datasets

Five artificial and 7 real-world datasets were used in the experiments; Table 3 and 4 give a summary for the 12 datasets. The last column in both tables contains in parenthesis the number of attributes.

Table 3. Artificial datasets

Dataset	Description	Number of examples
dataset #1	x1, x2 are the input variables and y the output variable; $x1 \in R$,$x2 \in R$, $y\in R$ $\begin{cases} Y = -2 * x1 & if\ \ 0 \leq x1 < 100\ and\ x2 = 0 \\ Y\ =\ -700 + 5 * x1\ if & 100 \leq x1 < 200\ and\ x2 = 0 \\ Y = 300 - 3 * x2 & if\ \ 0 \leq x2 < 100\ and\ x1 = 200 \end{cases}$	300 (3)
dataset #2	$x1 \in [0,250]$ and x2=0, and $y \in R$ $\begin{cases} y = x1; & if\ 0 \leq x1 < 50 \\ y = 100 - x1; & if\ 50 \leq x1 < 250 \end{cases}$	2500 (3)
CART dataset	This dataset was found in [2]with 10 predictor attributes:X1 ∈ {−1, 1} with equal probability that X1 =1 or X1 = -1; Xi ∈ {−1, 0, 1}, with i ∈ {2 . . . 10} and the predicted attribute y determined by $\begin{cases} Y\ =\ 3\ +\ 3X2\ +\ 2X3\ +\ X4 & if\ x1 = 1 \\ Y\ =\ -3 +\ 3X5\ +\ 2X6\ + X7 & if\ x = -1 \end{cases}$ A random noise ε between [-2 and 2] was added to Y	750 (11)
3DSin dataset	This dataset has two continuous predictor attributes x1, x2 uniformly distributed in interval [−3, 3] determined by Y = 3 sin(X1) sin(X2).	500(3)
Fried dataset	This dataset has 10 continuous predictor attributes with independent values uniformly distributed in the interval [0, 1] Y = 10 sin(πX1X2) + 20(X3−0.5)^2 + 10X4+5X5;A random noise ε between [-1;1] was added	700(11)

Table 4. Real world dataset

Dataset	Description	Number of examples (number of attributes)
Abalone	This dataset was obtained from UCI [16] machine learning repository.	4177 (8)
Auto-mpg	This dataset obtained from UCI [16] repository. Tuples with missing data were removed.	392 (8)
Boston Housing	This dataset obtained from UCI[16] repository	506 (14)
Kin8nm	This dataset was obtained from the DELVE [4]repository.	8192 (9)
Normalized Auto-mpg	This is the auto-mpg dataset from UCI [16] repository that has been normalized with z-score values	392 (8)
STOCK	This dataset is from SatLib[14]. The dataset contains 950 examples. However, the first tuple was removed because it did not appear correctly formatted	949 (10)
Tecator Dataset	This dataset originated from the StatLib [14] repository.	240 (11)

3.2 Experimental Methodology

All the experiments were done with a 10-fold cross validation and repeated 10 times with different seeds for each run. The average values are reported in table 7 along with corresponding standard deviation. Five artificial datasets were used three of which were previously used and results for GUIDE and SECRET are available in [5]. We also used 7 real world datasets of which 6 were previously used and results for GUIDE and SECRET available in [5]. The datasets which have not been previously used are dataset#1, dataset#2 and the normalized auto mpg dataset. We implemented TPRTI making use of R software [13] libraries. We run M5 using Weka [18]. R and Weka are publicly available software. Our implementation of RETIS relies on running TPRTI-A with all input attributes set as discrete attributes.

3.2.1 Input Parameters and Stopping Rules Used for the Experiments

For each dataset, different input parameters and stopping rules were used for TPRTI-A, and TPRTI-B. Table 5 summarizes the parameters used for each dataset where cosβ is the cosine threshold used to determine the turning points, “Min. Node Size” is the minimum required size of a node expressed as 100*(current node size)/(initial dataset), and “Min. RSS imp.” is the minimum improvement of the sum of the weighted RSS of both sub-nodes required to split a node. It is expressed as 100*(Parent RSS - weighted sum of children nodes RSS)/Parent RSS.

The input parameter “Subset Size” is used to subdivide the input data into subsets of equal size in order to compute the centroids. RETIS was run without post pruning.

Table 5. Input and stopping parameters for TPRTI

	TPRTI-A				**TPRTI-B**			
	Input parameters		**Stopping rules*		*Input parameters*		*Stopping rules*	
	Subset Size	cos β	Min. Node Size (in %)	Min. RSS imp. (in%)	Subset size	cos β	Min. Node Size (in %)	Min. RSS imp. (in %)
Dataset #1	3	0.8	10	10	3	0.8	10	10
Dataset #2	9	0.8	10	10	9	0.8	10	10
CART	5	0.8	10	10	5	0.8	10	10
3DSin	4	0.8	10	10	5	0.8	10	10
Fried	3	0.85	10	10	3	0.85	10	10
Abalone	55	0.8	10	10	21	0.8	10	10
Auto mpg	4	0.85	10	10	4	0.85	10	10
Boston Housing	14	0.8	12	12	14	85	12	12
Normalized auto mpg (z-score)	4	0.85	10	10	4	0.85	10	10
Stock Data	4	0.8	10	10	13	0.85	10	10
Tecator	8	0.8	10	10	21	0.7	10	10
Kin8nm	250	0.95	3	1	250	0.95	3	1

**Stopping rules:* The stopping parameters set for TPRTI-A are same parameters used for RETIS.

3.3 Results

Accuracy was measured by the MSE (Mean Squared Error). Model Complexity was measured by the number of leaf-nodes. However, a bad model may have small number of leaf-nodes. Thus complexity was slightly redefined as number of times an approach obtained the combination (best accuracy, fewest leaf-nodes). Both the number of leaf-nodes and MSE are provided as μ ± c where μ is the average MSE (or number of leaf-node) and c the standard deviation over several runs. Let $\mu_1 \pm c_1$ and $\mu_2 \pm c_2$ be two results such that $\mu_1 < \mu_2$. We consider a tie between the two results if $\mu_1 + c_1 > \mu_2$. Both Accuracy and number of leaf-nodes are reported in table 7 with the number of leaf-nodes in parenthesis. The main findings of our study are:

3.3.1 On Accuracy

TPRTI-A, and TPRTI-B are compared with the approaches in the columns. The number in each cell denotes (number of wins/number of ties/number of losses). For example, (6/5/1) in the first column of the first row means that TPRTI-A is more

Table 6. Comparison between TPRTI-A, TPRTI-B and state-of-the-art approaches with respect to accuracy

	M5	TPRTI-A	TPRTI-B	RETIS	GUIDE	SECRET
TPRTI-A	(6/5/1)	-	(4/6/2)	(4/6/0)	(5/1/2)	(4/2/2)
TPRTI-B	(4/6/2)	(2/6/4)	-	(3/7/0)	(5/1/2)	(1/4/3)

accurate than M5 on 6 datasets, ties M5 on 5 datasets and loses on 1 dataset. Overall, TPRTI yields comparable result as or slightly better result than RETIS. It has better accuracy than GUIDE and SECRET.

Table 7. Accuracy results

	M5	**RETIS**	**GUIDE**	**SECRET**	**TPRTI-A**	**TPRTI-B**
Dataset #1	446.8996 ±36.45 (11±0.00)	**0.000 ±0.0000** **(3±0.00)**	N.A	N.A	0.089 ±0.0000 **(3±0.00)**	0.089 ±0.0000 **(3±0.00)**
Dataset #2	4.75 ±0.239 (11±0.00)	**0.000 ±0.0000** **(2±0.00)**	N.A	N.A	**0.000 ±0.0000** **(2±0.00)**	**0.000 ±0.0000** **(2±0.00)**
CART	0.0932 ±0.0009 **(2±0.00)**	0.085 ±0.0125 (4.1±0.32)	N.A	N.A	**0.07614 ±0.0100** (4.1±0.32)	0.098±0.33 (6.1±0.32)
3DSin	**0.0015 ±0.0002** (20±0.00)	0.01 ±0.0074 (4±0.00)	0.0448 ±0.0018	0.0384 ±0.0026	0.0074 ±0.01 (4±0.00)	0.0063 ±0.01 **(3±0.00)**
Fried	4.888 ±0.1536 **(3±0.00)**	4.773 ±0.3893 **(3±0.00)**	**1.21** **±0.0000**	1.26 ±0.010	3.1114 ±0.80 (4±0.00)	1.4968 ±0.60 (6.7±0.48)
Abalone	4.6910 ±0.586 **(2±0.00)**	*N.A	4.63 ±0.04	4.67 ±0.04	4.3806 ±2.71 (4±0.00)	**4.1527±2.59** (5.1±0.45)
Auto mpg	8.5073 ±0.3105 (5±0.00)	8.8470 ±7.2183 **(3.1±0.32)**	34.92 ±21.92	15.88 ±0.68	**7.6021 ±6.33** (5±0.00)	8.4493 ±6.39 (4.6±0.52)
Boston Housing	28.8397 ±30.8896 (10±0.00)	24.569±20.090 **(4.2 ±0.92)**	40.63 ±6.63	24.01 ±0.69	**16.0922±10.29** (5.5±0.53)	19.6237 ±9.24 (4.8±0.92)
Normalized Auto mpg (z-score)	0.1396 ±0.0051 (5±0.00)	0.1186±0.0895 (4.0 ±0.00)	N.A	N.A	**0.1169 ±0.07** **(3.8±0.63)**	0.1342 ±0.09 (4.7±0.82)
Stock Data	1.0389 ±0.1008 (19±0.00)	11.977±7.884 (3.9 ±0.32)	1.49 ±0.09	1.35 ±0.05	**0.2067 ±0.10** **(3±0.47)**	4.8867 ±3.09 (4.9±0.88)
Tecator	9.4513 ±2.9519 (6±0.00)	6.6310±6.3036 (5.4 ±0.51)	13.46 ±0.72	12.08 ±0.53	**2.8315 ±1.412** **(3.1 ±0.31)**	7.1266 ±8.20 (6.4±0.70)
Kin8nm	0.0303 ±0.0009 (24±0.00)	*N.A.	0.0235 ±0.0002	**0.0222** **±0.0002**	0.0303 ±0.001 (5.33±0.57)	0.0227±0.0020 (25.5±0.17)

*N.A is used to express the fact that the program runs more than 3 hours without outputting a result or runs out of memory whereas N.A is used to express the fact that the result is not available.

3.3.2 On Model Complexity

In this study we consider a linear regression model to have a good model complexity when it is both accurate and has a small number of leaf-nodes. Table 8, which is compiled from table 7, presents the number of cases where an approach obtained both best accuracy and fewest nodes at the same time.

Table 8. Number of time an approach obtained the combination (best accuracy, fewest leaf nodes) for a dataset

M5	TPRTI-A	TPRTI-B	RETIS	GUIDE	SECRET
0	5	3	5	N.A	N.A

RETIS and TPRTI-A won the combination (best accuracy, fewest leaf-nodes) five times while M5 never won, and TPRTI-B won 3 times. This suggests that TPRTI hold comparable model complexity as RETIS.

3.3.3 On Scalability with Respect to Dataset Size

We use a direct comparison of runtime in seconds for TPRTI-A, TPRTI-B, and RETIS because they were implemented and run in the same machine. We use an indirect comparison to compare the different approaches. The indirect comparison consists of setting a baseline dataset size, and measuring the percent increase in runtime in relation to percent increase in baseline dataset size. Figure 4 summarizes our findings. TPRTI-B outperforms M5 consistently on all dataset sizes and number of input attribute. This suggests that TPRTI-B is a more scalable approach than M5. This is because models generated by TPRTI tend to have fewer nodes. On small to medium size dataset there is no significant difference between TPRTI and SECRET. Overall SECRET outperforms TPRTI consistently on all dataset size. Figure 5 summarizes our result for the direct comparison. In figure 5, Panel (A) shows that RETIS has the worst performance even on dataset with small number of input attribute. Panel (B) provides evidence that as the number of input attribute increases, performance decreases. Panel(C) and (D) demonstrate that TPRTI-B consistently outperform TPRTI-A.

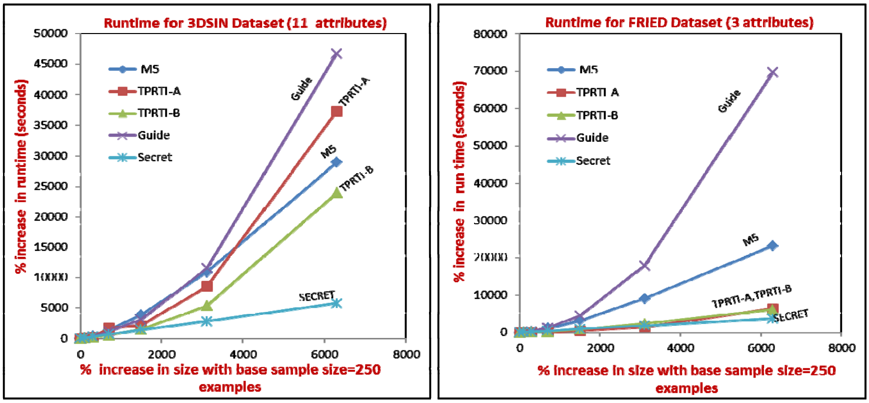


Fig. 4. Indirect Runtime comparison: Percent increase in baseline dataset size and the resulting percent increase in runtime with baseline size set to 250 examples

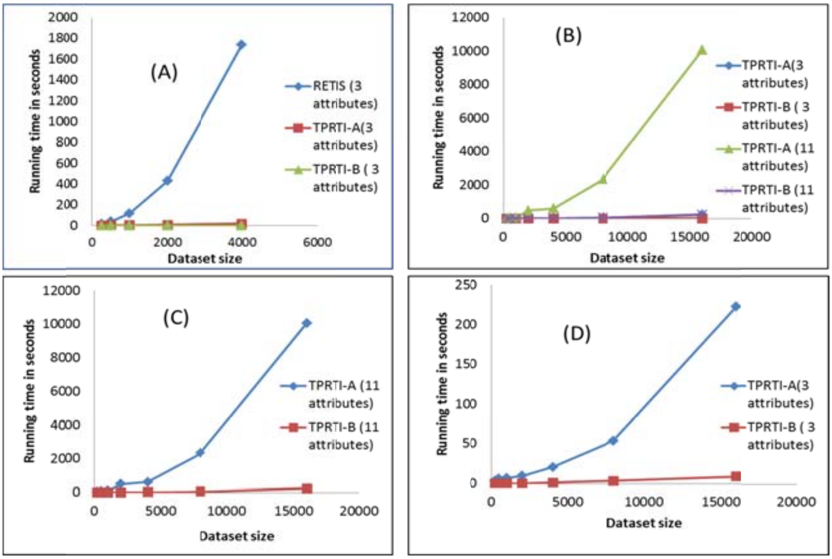


Fig. 5. Direct runtime comparison among TPRTI-A, TPRTI-B and RETIS

4 Conclusion

The paper proposes a new approach for Linear Regression Tree construction called Turning Point Regression Tree Induction (TPRTI) that infuses turning points into a regression tree induction algorithm to achieve improved scalability while maintaining accuracy and low model complexity. Two novel linear regression tree induction algorithms called TPRTI-A and TPRTI-B which incorporate turning points into the node evaluation were introduced and experimental results indicate that TPRTI is a scalable algorithm that is capable of obtaining a high predictive accuracy using smaller decision trees than other approaches.

Future Work. We are investigating how turning point detection can also be used to induce better classification trees.

References

1. Alexander, W.P., Grimshaw, S.D.: Treed regression. Journal of Computational and Gra aphical Statistics 5, 156–175 (1996)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth, Belmont (1984)
3. Chaudhuri, P., Huang, M.-C., Loh, W.-Y., Yao, R.: Piecewise-polynomial regression trees. Statistica Sinica 4, 143–167 (1994)

4. DELVE repository of data (April 12, 2012), `http://www.cs.toronto.edu/~delve/`
5. Dobra, A., Gehrke, J.: SECRET:a scalable linear regression tree algorithm SIGKDD 2002 (2002)
6. Friedman, J.: Multivariate adaptive regression splines (with discussion). Annals of Statistics 19, 1–142 (1991)
7. Karalic, A.: Employing linear regression in regression tree leaves. In: European Conference on Artificial Intelligence, pp. 440–441 (1992)
8. Li, K.C., Lue, H.H., Chen, C.H.: Interactive Tree-Structured Regression via Principal Hessian Directions. Journal of the American Statistical Association 95, 547–560 (2000)
9. Loh, W.-Y.: Regression trees with unbiased variable selection and interaction detection. Statistica Sinica 12, 361–386 (2002)
10. Loh, W.-Y., Shih, Y.-S.: Split Selection Methods for Classification Trees. Statistica Sinica 7, 815–840 (1997)
11. Malerba, D., Esposito, F., Ceci, M., Appice, A.: Top-down induction of model trees with regression and splitting nodes. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 612–625 (2004)
12. Quinlan, J.R.: Learning with Continuous Classes. In: 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348 (1992)
13. The R Project for Statistical Computing official website as of August 8, 2012, `http://www.r-project.org/`
14. StatLib repository (Dataset Archive), as of April 12, 2013, `http://lib.stat.cmu.edu/datasets/`
15. Torgo, L.: Functional models for regression tree leaves. In: Proc. 14th International Conference on Machine Learning, pp. 385–393. Morgan Kaufmann (1997)
16. UCI repository as of April 12, 2012, `http://archive.ics.uci.edu/ml/datasets.html`
17. Vogel, D.S., Asparouhov, O., Scheffer, T.: Scalable Look-Ahead Linear Regression Trees KDD 2007, San Jose, California, USA, August 12-15 (2007)
18. Weka software official website as of August 8, 2012, `http://www.cs.waikato.ac.nz/ml/weka/`
19. `http://www2.cs.uh.edu/~UH-DMML/index.html`

Automatic Classification of Web Databases Using Domain-Dictionaries

Heidy M. Marin-Castro[1], Victor J. Sosa-Sosa[1],
Ivan Lopez-Arevalo[1], and Hugo Jair Escalante-Baldera[2]

[1] Center of Research and Advanced Studies of the National Polytechnic Institute
Information Technology Laboratory
Victoria City, Tamaulipas. Mexico
[2] National Institute for Astrophysics, Optics and Electronics
Tonantzintla, Puebla. Mexico

Abstract. The identification, classification and integration of databases on the Web (also called web databases) as information sources is still a great challenge due to their constantly growing and diversification. The classification of such web databases according to their application domain is an important step towards the integration of deep web sources. Moreover, given the design and content heterogeneity that exists among the different web databases, their automatic classification become a great challenge and a highly demanded task, requiring techniques that allow to cluster web databases according to the domains they belong to. In this paper we present a strategy for automatic classification of web databases based on a new supervised approach. This strategy uses the visible information available on a group of specific-domain Web Query Interfaces (WQIs) to construct a dictionary or lexicon that will allow to better describe a particular domain of interest. The dictionary is enriched with synonyms. In our experiments, the dictionary was built from a set of randomly selected specific-domain WQIs. The automatic WQI classification based on dictionaries generated in this way showed efficient and competitive results compared against related work.

Keywords: deep web, web databases, web query interfaces, classification, domain-dictionary, models of vectorial representation.

1 Introduction

Nowadays the Web remains as one of the most important sources of information, which can be divided in surface web and deep web. Compared with the surface web, the huge amount of information stored on the deep web cannot be easily crawled and indexed by conventional Web search engines as *Google*, *Yahoo*, *Bing* among others [3]. Furthermore, deep web is conformed by web databases, that store information related to a specific application domain, for example molecular biology, job postings, flight schedules, accommodation reservation, etc. The information coming from the deep web and presented to the user is built dynamically as an HTML response to a query that was sent through a

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 340–351, 2013.

Web Query Interface (WQI). A WQI is a special type of HTML form designed to access information of a specific web database. As new WQIs are designed, others are frequently removed or changed. Since a WQI is designed to access a specific web database, it would be desired that a single WQI could access all the web databases related to a specific domain available in the deep web. For example, it would be desirable for an user interested on flights to use a single WIQ to search for flights across all companies or airlines So, in this suggested integration process, a task that must be carried out is the classification of web databases according to the domain they represent.

One first challenge that a user faces when trying to access web databases is to know where they can be found (discovery), because they are distributed and appear in all over the web. As a difference with traditional databases, the web database schemes (meta data) are unknown for users, so they must be deduced from their superficial data, which is showed through the WQIs. This visible information usually includes a brief description of attributes associated with the corresponding web database design. Since WQIs are not created neither designed to be used and understood by computer applications, simple tasks (from the position of an user) such as content identification or fill out HTML forms turn out to be a computationally hard problem [16]. Moreover, the structure of WQIs of web databases change frequently without notification and their query capabilities can be limited due to the amount of data received by the user as well as the number of attributes contained in the schema itself. After the identification (discovery) of web databases, one important issue is to determine the application domain of such web databases, which is challenging because the homonymous terms that exist in the WQIs. For example, WQIs of different domains use similar terms for different purposes. In other cases, there may be such homonym even inside a particular domain. Homonymy (semantic ambiguity) is a well known issue in information integration tasks. However it has not been fully solved and keeps as a research topic mainly for smaller and static scenarios as the case of WQIs.

In this work we present a strategy for automatic classification of web databases into domains based in the pre-query approach [15]. The key part in our proposed strategy is the building of domain-dictionaries for classification of new examples of web databases. Several works reported in the literature for classification of web databases [8,2,11] use a clustering model that is not able to automatically assign a class name to the generated clusters. Moreover, for some samples that share properties with members of different clusters, it could be hard to assign the correct class name, leading to an ambiguity problem. Our proposed strategy identifies representative terms contained in WQIs that belong to a specific domain and uses them for building domain-dictionaries. Terms are cleaned applying a pre-processsing task, which consider the elimination of stop-words using the rules described in [4]. Afterwards, the terms are transformed to their root words by a stemming process. The resulting terms are sorted according to their relevance, determined using the TF-IDF weighting scheme. The WQIs are represented using a vectorial representation model called bag of words (BOW), and faced the homonymy problem of terms by using a weighted terms schema.

The rest of this paper is organized as follows: Section 2 describes works related to classification of web databases. Section 3 presents the proposed strategy based on the building of domain dictionaries. Section 4 presents and discusses the experimental results obtained by our proposed strategy. Finally, Section 5 concludes this work.

2 Related Work

Most of the reported works to solve the problem of automatic classification of web databases can be categorized into two approaches: Pre-Query and Post-Query [15]. The Pre-Query approach [2,8,11,17] exploits the internal content provided by WQIs such as control elements, attributes, values of attributes and user's tags. This approach only relies on visible features of WQIs. On the contrary, the Post-Query approach [10,6,7] is based on the identification of web databases from the resulting pages retrieved by submitting probing queries through WQIs. This approach depends on the correct construction of queries for a given domain. The Post-Query approach is difficult to apply to WQIs with multi-attributes because they require to be automatically filled out with valid values.

In [8], Kabisch used two different methods for domain classification: a pattern-based classification method and a neighbor-based classification technique. The first approach is based on a set of mappings among WQIs from a learning set to derive interface domain-patterns and cluster all elements by their concepts. The pattern-based classification method uses a representation model of tree for unknown WQIs schemes. It matches the nodes of the schema tree to the pattern of each known domain. Then, it assigns the interface to the domain containing the best matching pattern. Kabisch derives a useful domain pattern from a sample set of interfaces if (1) the number of relevant concepts in a specific application domain is rather small and (2) the elements on the query interfaces can be clustered to their concepts by exploiting similarities among their terms used, structures and layouts. In the neighbor-based classification method, it is assumed that those interfaces that contain similar elements and have a similar structure most likely belong to the same domain. Therefore, the classification problem is transformed into a similarity estimation problem at interface level.

Barbosa and Freire [2] proposed a clustering strategy called CAFC-C (Context-Aware Form Clustering). This strategy is based on a *k-means* clustering algorithm which uses the form-page model and obtains clusters that are homogeneous. To improve the quality of CAFC-C, the authors exploit a set of features provided by: the hyperlink structure, e.g., anchor text and the quality of hub pages; and form contents, e.g., structural information and automatically extracted labels. The authors use a broad set of meta data in the form context, instead of just the form content. This allows CAFC to uniformly handle web databases accessible through both single- and multi attribute forms.

In [11] the authors propose a new approach for clustering e-commerce search engines (ESEs) on the Web, which utilizes the features available on the WQI, including the label terms and value terms appearing in the search form, the number of images, normalized price terms as well as other terms. They report the design of WISE-Cluster, a tool that aims to automatically cluster *ESEs* into different groups according to their domains. However, the authors do not solve the problem of ambiguity of terms.

Wang, et. al [17] present a framework for locating Hidden-Web databases based on a focused crawler assisted by an ontology. Their framework is composed by three classifiers: a Web Page Classifier (WPC), a Form Structure Classifier (FSC) and a Form Content Classifier (FCC). The WPC searches interesting Web pages by means of analyzing their features. FSC determines whether these pages contain searchable forms by analyzing their structural characteristics and the FCC identifies whether searchable forms belong to a given domain. The authors implemented an ontology which contains concepts and relations of DOCM (Domain Ontology Concept Model) from searchable forms and resulting pages. To locate interesting Web pages that may contain searchable forms belonging to a given domain, the crawler follows two strategies: The crawler starts from a Web page which is classified as belonging to a topic. From that Web page the crawler follows the hyperlinks found on that page until a specified level of depth is reached. During the search of interesting Web pages the crawler builds two vectors, a page feature vector and a topic vector to find similarity between the page found and the source page. FCC is proposed to identify domain-specific databases by analyzing domain-specific form content.

In [5] the authors transform the problem of sources organization into clustering of categorical data. Their approach hypothesizes that homogeneous sources share the same hidden generative model, which probabilistically generates schemes from a finite vocabulary of attributes. The authors propose a new objective function, model-differentiation, which employs an hypothesis to maximize statistical heterogeneity among clusters. They derived a new similarity measure for the general HAC (Hierarchical Agglomerative Clustering) algorithm.

Table 1 shows a summary of related works that face the automatic classification problem of web databases using an pre-query approach. In table 1, the second column indicates the technique used for classification, the third column shows the number of domains used for experimental evaluation, the fourth column represents the average percentage of web databases classified correctly over the distinct domains and finally, the last column, indicate the weighted harmonic mean of precision and recall of the classification process.

3 Proposed Strategy for Automatic Classification of WQIs

In this section, we present a strategy for automatic classification of WQIs based on domain dictionaries. This strategy uses textual labels of fields present in the WQIs as the main elements to determine the WQI's domain. We consider

Table 1. Reported Works for automatic domain classification of web databases

Ref.	Techniques	Domains	Accuracy	F-measure
[8]	Pattern based classification	7	78%	80%
[8]	Neighbor based classification	7	90%	–
[2]	Clustering based on the textual contents (CAFC-C)	8	–	88%
[17]	Form Content Classifier (FCC) assisted by an ontology	1	–	89%
[11]	ESEs (e-commerce search engines) Clustering	12	–	90%

the problem of web databases classification as a text-classification task, where there is a restricted vocabulary of terms that appear in a WQI, which describes its semantics. Figure 1 shows an overview of the proposed strategy, which is formed by four components: an automatic extractor of terms, a pre-processing module, a dictionary-generator and classification module. We start from a set of WQIs detected and extracted form a set of Web pages belonging to a specific domain. Then, textual-labels associated to fields are extracted from WQIs by using a heuristic. These textual-labels called terms are cleaned, and finally, the homonymy problem is addressed over the resulting set of terms, by assigning weights to the terms according to their frequency. This weight assignment allows us to select those terms that better characterize WQIs for a specific domain and they are used in the classification process.

We represent the terms extracted from the WIQs using the distributional term representation model BOW (Bag of Words), which is the mos used in the literature. BOW assumes that a document d_i is represented as a vector of terms

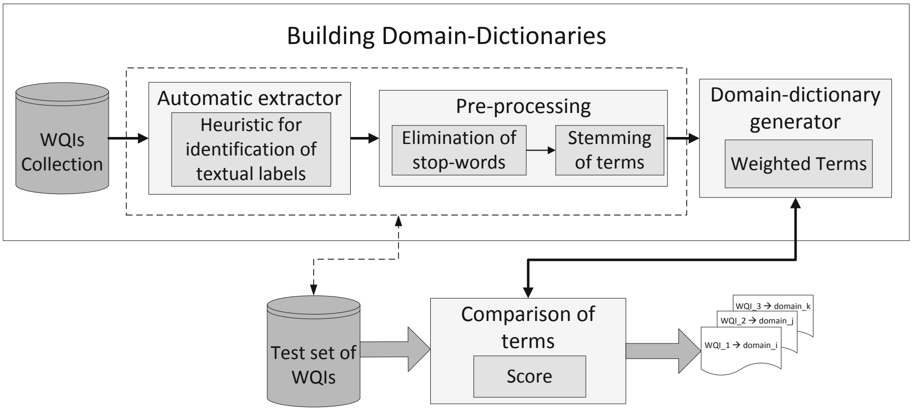


Fig. 1. Block diagram of the proposed strategy for automatic classification of web databases

weights $d_j = (w_{1j}, ..., w_{rj})$ where r is the cardinality of the dictionary δ and $0 \leq w_{kj} \leq 1$ represents loosely speaking the contribution of term t_k to the specification of the semantics of d_j [9].

In the next section we describe with details each component in our proposed strategy shown in figure 1.

3.1 Automatic Term Extractor

After the WQIs are identified, gathered and organized in the WQIs repository, they are processed to extract their textual labels of fields (o terms) associated to each control element present on them. This extraction process is challenging due to the heterogeneity in the design, structure, and domain values of the WQIs. In addition, it is not possible to assume that some terms present in a WQI will be present in other WQIs even for the same domain. That is why it is difficult to identify those terms that provide useful information about the WQI's domain. Textual-labels of fields are identified by a heuristic, based on the spatial position of the labels. The proposed heuristic is as follows. The segment of HTML document corresponding to the WQI identified is cleaned, removing white spaces and newlines, above, below, before and after each control element in the WQI. Afterwards, the row number where each control element in the WQI appears is located. Each control element has a field, which is defined as the triplet <*name, type, value*>, where name represents the name of the field, type of control element *radio, text box, check box* and value represents the domain value of field. Once the row is identified, the textual-label is extracted taking into consideration two aspects:

1. If the control element corresponds to a text-box, the textual-label is commonly located before, up, or inside the control element.
2. If the control element is a radio-button, the label is commonly located after the control element. The same applies for list-down.

In the heuristic we propose, only three types of control elements are considered: text-box, list-down, and check-box, because those are some control elements that better characterize a WQI [13].

3.2 Pre-processing

In information retrieval the elimination of stop-words is a common process. A stop-word is a extremely common word (such as articles, prepositions, conjunctions or any other term that is not considered as relevant in an language) with little value for documents matching, that bear no semantic content by itself [12]. The identification of stop-words in textual terms on WQIs can be tricky, for example the words *from* and *to* in a WQI schema for the traveling (airline or bus) domain can denote an *origin* and *destination city*, respectively. If these words were removed, an empty label will result which is unacceptable. Similarly, if these words are found in the labels *from city* and *to city* and they are removed, then the resulting labels are undistinguished (city in this case), which leads to the homonym problem [4]. The pre-processing that we applied to text terms on WQIs consists of three normalized steps:

- First, all the terms on the WQIs are identified, extracted and stored in a collection S as it was explained in section 3.1. We identified and removed true stop-words from S using the analysis done in [4], where the authors establish two assertions: first, a true stop-word is one that does not generate empty labels and second the elimination of a stop-word must not generate a homonym or hypernym problem with another text-label in the WQIs.
- Next, we normalized all remaining terms in S by applying a steaming process, which consists in reducing inflected (or sometimes derived) words to their stem, base or root form, generally a written word form. The stem is not necessarily identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if that stem is not a valid root.
- Finally we identified synonymous for the remaining terms in S using the lexicon-conceptual dictionary WordNet. These synonymous are aggregated to domain dictionaries.

3.3 Domain-Dictionary Generator

A problem faced in the classification of WQIs is the existence of homonymous terms, that is, similar terms for different purposes depending on the application domain of the WQI, considering also, that there are terms that have multiple synonym terms on different WQIs. This is not new in information retrieval, and its solution would improve significantly the classification task of web databases.

To disambiguate the sense of terms, we compute the relevance of each term using the TF-IDF vector-space model [12]. Let the frequency of a term t to be defined as the number of occurrences of t contained in the WQI schema Q, denoted as $freq(Q; t)$. The (weighted) term-frequency matrix TF(Q; t) measures the association of a term respect to Q. It is generally defined as 0 if the WQI does not contain the term, and nonzero otherwise. There are many ways to define the term-weighting for the nonzero entries in such vector. In our case, we can simply set TF(Q; t) = 1 if the term t occurs in the WQI schema Q.

Besides the term frequency measure, there is another important measure called inverse document frequency (IDF), which represents the scaling factor or the importance of a term t. If a term t occurs in many WQIs, its importance will be scaled down due to its reduced discriminative power. We defined $IDF(t)$ by the following formula:

$$IDF(t) = \log \frac{N}{n_i} \tag{1}$$

where N represent the total collection of WQIs and n_i is the set of WQIs containing term t. If $n_i << N$, the term t will have a large IDF scaling factor, otherwise will be negligible.

Finally, the weight of a term is obtained by the relation between the metrics TF and IDF in equation 2

$$TF - IDF(Q_j, t_i) = TF(Q_j, t_i) \times IDF(t_i) \tag{2}$$

For each training set of WQIs for specific application domain we built a domain dictionary of weighted terms. We define a domain dictionary as:

Definition. A domain dictionary D is a set of interrelated terms to represent the topic of a set of WQIs. Each domain dictionary can be seen as a set of triplets of the form $< t_i, f_i, w_i >$ where t_i is the ith term of domain dictionary δ; f_i is the frequency of this ith term and w_i correspond to assigned weight using the vector-space model TF-IDF.

3.4 Comparison of Terms

A domain dictionary contains relevant terms associated to a given domain together with the synonyms for each term. These dictionaries are then used to classify new instances of WQIs, assigning the domain name to the given WQI. The classification consist in comparing the terms of the incoming new WQI, each term t_i in the WQI is looked up in the domain dictionary d_j and if it matches, the corresponding weight associated to t_i in the dictionary is accumulatively added to obtain the score s_j of the WQI for the domain represented by d_j. Afterwards, once all of the dictionaries have been considered, the domain assigned for the WQI is that corresponding to the highest score s_j.

The next section describes the experimental results using the strategy for web databases classification proposed in this work.

4 Experimental Results

For building our domain-dictionaries we used the WQIs contained in ICQ dataset from the University of Illinois at Urbana-Champaign UIUC repository [1]. We built three domain dictionaries using the datasets of reference: $airfares$, *books*, and *automobiles*. Additionally, the Tel-8 dataset from the UIUC repository was used as test set to evaluate our proposal.

We carried out three experiments with the aim of evaluating the effectiveness of our domain-dictionaries-based proposal:

1. We evaluated the precision of our strategy by computing the number of true positives (i.e. the sum of true positives and false positives that are WQIs incorrectly labeled as belonging to the positive class) divided by the total number of WQIs labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are WQIs incorrectly labeled as belonging to the class) [14], that is the ratio of number of WQIs that are correctly classified divided by the number of WQIs with a class name assigned.

2. Also, we evaluated the proportion of true positives of WQIs cases divided by the total number of WQIs cases that actually belong to the positive class (recall or sensitivity) [14], that is the ratio of number of WQIs that are correctly classified divided by the number of WQIs that should have been classified.
3. Finally, we computed the $f - measure$ to normalize the precision and recall generated since higher precision is generally sacrificed by lower recall and vice versa.

Table 2. Comparison of the domain-dictionary based strategy against related works for web database classification

Strategy	Accuracy	F-measure
Domain-Dictionaries (Proposed strategy)	92%	94%
Pattern based classification [8]	78%	80%
Neighbor based classification [8]	90%	–
FCC assisted by an ontology [17]	–	89%

In each experiment we used training sets of different sizes (twenty, ten and five WQIs samples) from three datasets: $airfares$, *books*, and *automobiles*. The instances of the training set were selected randomly, considering a real scenario where new WQIs can be classified from a small number of sample WQIs. To evaluate the proposed approach we used as test set one hundred WQIs from the TEL-8 dataset.

Figures 2, 3, 4 show the precision, recall and f-measure respectively obtained using different training set sizes. Figure 2 shows that the average precision using

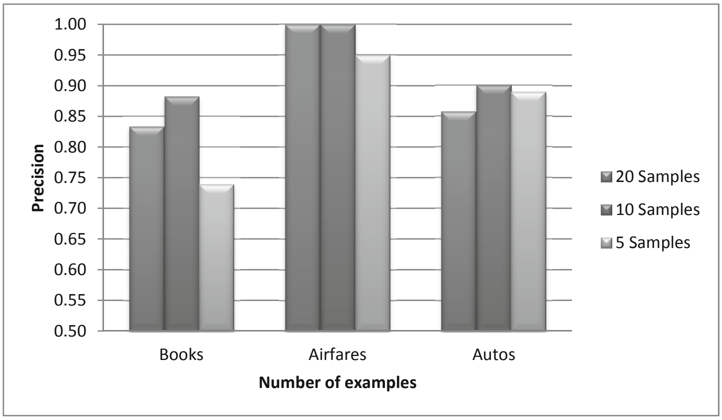


Fig. 2. Precision of the domain-dictionaries strategy for automatic classification of web databases

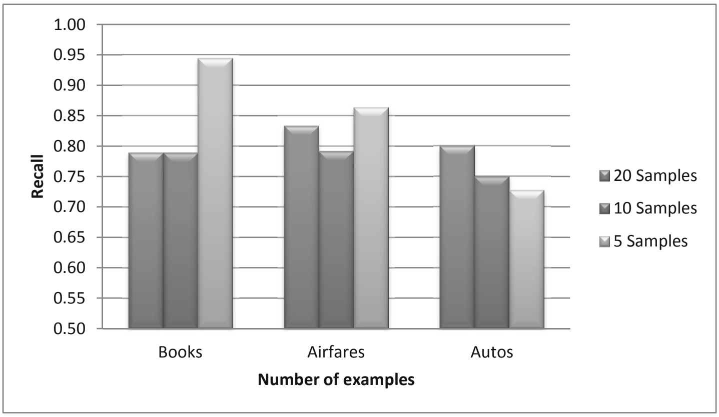


Fig. 3. Recall of the domain-dictionaries strategy for automatic classification of web databases

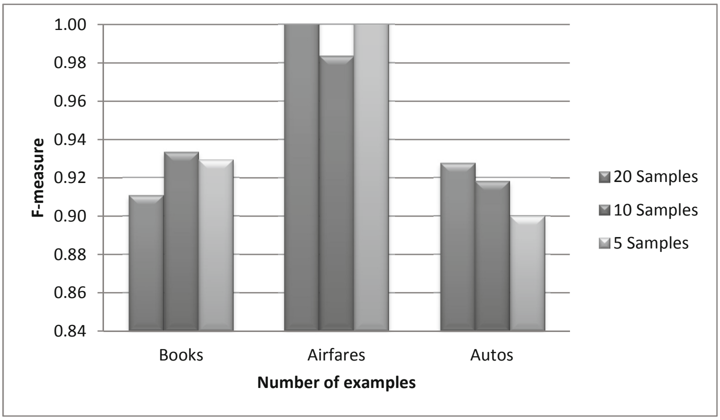


Fig. 4. F-measure of the domain-dictionaries strategy for the automatic classification of web databases

the domain dictionaries strategy is approximately 90% for most of the considered domains. It demonstrates that the proposed strategy is highly competitive compared against other approaches in the literature (see table 1). As it can be observed from figure 3, the average recall is close to 0.8. The f-measure shown in figure 4 allows to weigh the precision and recall, showing the performance of our proposed approach for web databases classification. In the graph shown in figure 4, f-measure is about 84%, a value that outperforms the one obtained in [8].

Table 2 compares our domain-dictionaries proposal against two related works that also use a supervised strategy. In [8], Kabich used a dataset that contain 140 WQIs of seven domains. They pick up one interface, learn the domains using a subset of the remaining interfaces and apply the classification algorithm for the selected interface. On the other hand, in [17], the authors used only one domain to carry out one test of precision, f-measure and recall of their word. They used 160 WQIs from the domain of Books. In table 2 we can see that the proposed strategy of domain-dictionaries can identify WQIs with higher precision compared to other works.

5 Conclusions

Web Query Interfaces (WQIs) allow to access web databases and retrieve useful information that is not reachable by traditional search engines. The automatic classification of web databases is a complex task given the heterogeneity in their design, style, semantic content among others. This work presented an strategy for automatic classification of web databases. The performance of the proposed strategy was evaluated by performing experiments using Tel-8 and ICQ datasets. Our strategy addresses the homonymy problem by the assignation of weights to those relevant terms and their synonyms contained in WQIs. The results achieved give evidence of the effectiveness and usefulness of the domain-dictionaries based proposed strategy.

Acknowledgements. This research was partially supported by project number 203595 from Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas

References

1. The UIUC web integration repository. Computer Science Department. University of Illinois at Urbana-Champaign (2003), http://metaquerier.cs.uiuc.edu/repository
2. Barbosa, L., Freire, J., da Silva, A.S.: Organizing hidden-web databases by clustering visible web documents. In: ICDE, pp. 326–335 (2007)
3. Bergman, M.K.: The deep web: Surfacing hidden value (white paper). Journal of Electronic Publishing 7(1) (2001)
4. Dragut, E.C., Fang, F., Sistla, A.P., Yu, C.T., Meng, W.: Stop word and related problems in web interface integration. PVLDB 2(1), 349–360 (2009)
5. He, B., Tao, T., Chang, K.C.-C.: Organizing structured web sources by query schemas: a clustering approach. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM 2004, pp. 22–31. ACM, New York (2004)
6. Jiang, L., Wu, Z., Feng, Q., Liu, J., Zheng, Q.: Efficient deep web crawling using reinforcement learning. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS, vol. 6118, pp. 428–439. Springer, Heidelberg (2010)
7. Jiang, L., Wu, Z., Zheng, Q., Liu, J.: Learning deep web crawling with diverse features. In: Web Intelligence, pp. 572–575 (2009)
8. Kabisch, T.: Extraction and integration of Web query interfaces. PhD thesis (2011)

9. Lavelli, A., Sebastiani, F., Zanoli, R.: Distributional term representations: an experimental comparison. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM 2004, pp. 615–624. ACM, New York (2004)
10. Lin, L., Zhou, L.: Web database schema identification through simple query interface. In: Lacroix, Z. (ed.) RED 2009. LNCS, vol. 6162, pp. 18–34. Springer, Heidelberg (2010)
11. Lu, Y., He, H., Peng, Q., Meng, W., Yu, C.T.: Clustering e-commerce search engines based on their search interface pages using wise-cluster. Data Knowl. Eng. 59(2), 231–246 (2006)
12. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
13. Marin-Castro, H.M., Sosa-Sosa, V.J., Martinez-Trinidad, J.F., Lopez-Arevalo, I.: Automatic discovery of web query interfaces using machine learning techniques. Journal of Intelligent Information Systems 40(1), 85–108 (2013)
14. Powers, D.M.W.: Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. Journal of Machine Learning Technologies 2(1), 37–63 (2011)
15. Ru, Y., Horowitz, E.: Indexing the invisible web: a survey. Online Information Review 29(3), 249–265 (2005)
16. Shestakov, D.: Search Interfaces on the Web:Querying and Characterizing. PhD thesis, University of Turku Department of Information Technology (2008)
17. Wang, Y., Li, H., Zuo, W., He, F., Wang, X., Chen, K.: Research on discovering deep web entries. Comput. Sci. Inf. Syst. 8(3), 779–799 (2011)

An Efficient and Scalable Algorithm for Mining Maximal High Confidence Rules from Microarray Dataset

Wael Zakaria Abd Allah[1], Yasser Kotb El Sayed[1,2],
and Fayed Fayek Mohamed Ghaleb[1]

[1] Ain Shams University, Faculty of Science,
Mathematics/Computer Science Department-Abbassia, Cairo, Egypt
[2] Information Systems Department, College of Computer and Information Sciences,
Al-Imam Muhammad ibn Saud Islamic University, Riyadh, KSA
{Wael_Abdallah,yasser_kotb}@sci.asu.edu.eg,
fmghaleb@yahoo.com

Abstract. DNA microarrays allow simultaneous measurements of expression levels for a large number of genes within a number of different experimental samples. Mining association rules algorithms are used to reveal biologically relevant associations between different genes under different experimental samples. In this paper, we present a new mining association rules algorithm called *Mining Maximal High Confidence Rules* (MMHCR). The MMHCR algorithm is based on a column (gene) enumeration method which overcomes both the computational time and memory explosion problems of column-enumeration method used in many of the mining microarray algorithms. MMHCR uses an efficient data structure tree in which each node holds a gene's name and its binary representation. The binary representation is beneficial in two folds. First, it makes MMHCR easily find all maximal high confidence rules. Second, it makes MMHCR more scalable than comparatives. In our experiments on a real microarray dataset, MMHCR attained very promising results and outperformed other counterparts.

Keywords: Data mining, DNA microarray, mining association rules, closed itemsets, row enumeration, column enumeration, maximal high confidence rules.

1 Introduction

Gene expression is the process of transcribing DNA sequences into mRNA sequences, which are later translated into amino acid sequences called *proteins*. The number of copies of the produced RNA is called the *gene expression level*. The regulation of gene expression level is essential for proper cell function. Microarray technologies provide the opportunity to measure the expression level of tens of thousands of genes in cells simultaneously. Usually, the expression level is correlated with the corresponding protein made under different conditions (samples). Due to the huge amount of data, most studies are focused on the analysis and the extraction of useful and interesting information from microarray data [1-3].

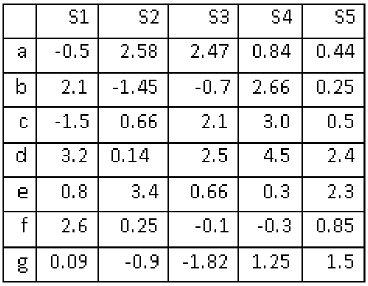
P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 352–366, 2013.

The microarray dataset can be seen as an M×N matrix G of expression values; where the rows represent genes g_1, g_2, …, g_m and the columns represent different experimental conditions or samples s_1, s_2, …, s_n. Each element G[i, j] represents the expression level of the gene g_i in the sample s_j see Table 1. The matrix usually contains a huge data; therefore, data mining techniques are used to extract useful knowledge from such matrices [3-4].

Mining association rules is currently a vital data mining technique for many applications [4-6]. Mining association rules technique is applied to microarray dataset to extract interesting relationships among sets of genes [2], [4], [7]. Let G_1 and G_2 be two sets of genes, then the association rule $G_1 \rightarrow G_2$ (with support 80% and confidence 90%) unmasks the relations among those sets of genes, that is, both sets are expressed in 80% of the microarray experiments and if G_1 is expressed then G_2 is also expressed with probability 90% [3].

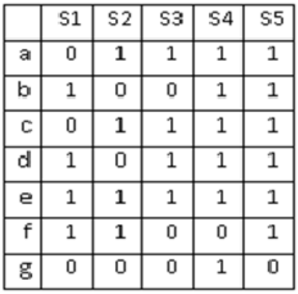
In order to mine association rules in microarray dataset, the data is pre-processed by applying the logarithms procedure to ensure that the data is suitable for analysis. The logarithms procedure transforms DNA microarray data from the raw intensities into log intensities [1]. Then, the transformed dataset is *discretized* [8] into binary matrix such that each value is mapped into 1 or 0 based on whether greater than or equal (≥) or less than (<) a predefined specified threshold respectively-in this paper the predefined threshold is 0.2. Table 2 shows the discretized microarray. Many frequent pattern mining methods are based on one of two perspectives, either *column enumeration* methods or *row enumeration* methods [4].

Table 1. Microarray Dataset



	S1	S2	S3	S4	S5
a	-0.5	2.58	2.47	0.84	0.44
b	2.1	-1.45	-0.7	2.66	0.25
c	-1.5	0.66	2.1	3.0	0.5
d	3.2	0.14	2.5	4.5	2.4
e	0.8	3.4	0.66	0.3	2.3
f	2.6	0.25	-0.1	-0.3	0.85
g	0.09	-0.9	-1.82	1.25	1.5

Table 2. .Discretized Microarray



	S1	S2	S3	S4	S5
a	0	1	1	1	1
b	1	0	0	1	1
c	0	1	1	1	1
d	1	0	1	1	1
e	1	1	1	1	1
f	1	1	0	0	1
g	0	0	0	1	0

In this paper, we present a new algorithm based on the column (gene) enumeration method. The proposed algorithm is called *Mining Maximal High Confidence Rules (MMHCR*) which overcomes both the computational time and memory explosion problems of column-enumeration used in many algorithms for mining microarray datasets [4]. MMHCR scans the microarray dataset to obtain a list of all genes. Each gene is associated with a binary representation where each element in the representation shows whether the gene is expressed in the corresponding sample. For example, the binary representation of the gene *a* is [01111] as in Table 2. This structure helps in finding easily and faster all maximal high confidence rules. The experimental results show that the MMHCR algorithm is faster than the row-enumeration based methods MAXCONF [9] and RERII [10]. Since, RERII and MAXCONF are better than other methods based on column enumeration like CHARM [11], As a result, MMHCR algorithm is also faster than the column enumeration method CHARM.

The rest of the paper is organized as follows. In section 2, we present the background of mining association rules, while in section 3, we present related works. In section 4, the proposed MMHCR algorithm is presented for extracting all maximal high confidence rules. In section 5, we show our experimental results. Finally, we conclude the paper in section 6.

2 Mining Association Rules

Mining association rules technique [12] extracts interesting relationships among sets of items (genes) in a large dataset. One of the most famous applications of this technique is *market basket analysis* [5-6] where the objective is to find the relationships between the purchased items under different transactions. Also, mining association rules is applied in microarray datasets in order to find the relationships between genes under different samples. For example, Table 3 shows the transactions (samples) datasets of the discretized dataset in Table 2. In this section, we introduce some notations we use throughout the paper.

Table 3. (Transactions (Samples) Dataset)

tiD (samples id)	Transactions (samples)
1	b,d,e,f
2	a,c,e,f
3	a,c,d,e
4	a,b,c,d,e,g
5	a,b,c,d,e,f

Definition 1 (Association Rules). Let I= $\{i_1, i_2, \ldots, i_n\}$ be a set of items (genes). A subset T$\subseteq$ I is called a *transaction (sample)*. The transactions (samples) dataset *D* is a set of transactions; where each transaction has a unique id called *tid*. In other words, D= {<tid,T>; T$\subset$I, tid $\in$ {1, 2, ..., k}; k=|D|}. An *association rule* is a pair of itemsets(X, Y) where X, Y $\subseteq$ I and X$\cap$Y=Φ, and is denoted by X→Y. The itemsets (set of items) X and Y are called *antecedent* and *consequent* of the rule X→Y, respectively.

Convention: T$\in$D denotes that $\exists$ tid such that <tid,T>$\in$D.

For example, the transaction (sample) *{b, d, e, f}* (Table 3) at the first row (sample) represents the items (genes) that appear (expressed) at sample 1.

Definition 2 (Measurements of Association Rules). An association rule X→Y has two measurements: *support* and *confidence*. They are defined, with respect to a transactions (samples) dataset D, as follows:

$$\mathrm{supp_D}(X \to Y) = \frac{\mathrm{supp_D}(X \cup Y)}{|D|}$$

$$\mathrm{conf_D}(X \to Y) = \frac{\mathrm{supp_D}(X \cup Y)}{\mathrm{supp_D}(X)}$$

$$\text{where } \mathrm{supp_D}(X) = |\{T; X \subseteq T, T \in D\}|.$$

Definition 3 (Frequent Itemset and Strong (Confident) Association Rules). The itemset X is called *frequent* if supp(X) $\geq$ minsup; where minsup is a user defined threshold. The rule X→Y is called *strong* (or *confident)* if supp(X→Y) $\geq$ minsup and conf(X →Y) $\geq$ minconf; where minconf is another user defined threshold.

The process of mining confident association rules is performed in two steps [5-6], [12]:

1. Generate all frequent n-itemsets (set of n items).
2. Using all frequent n-itemsets, generate all strong (confident) association rules X→Y, where X and Y are frequent n-itemsets.

The dataset such "*market basket analysis*" has the property that the number of items in the dataset is less than the number of transactions. These kinds of dataset called sparse, i.e., the longest frequent itemsets are relatively short. However, there are many real-life datasets such as microarray datasets, that the number of items (genes) in the dataset is greater than the number of transactions (samples). This kind of datasets called *dense*, i.e., they contain very long frequent itemsets (genesets). Therefore, generating all frequent itemsets in such dense dataset requires large memory. Hence, recent algorithms prevent this problem by expanding only frequent closed itemsets [9-10], [13-15].

Definition 4 (Frequent Closed Itemset). The frequent itemset X is called a *frequent closed itemset* if ∄ a frequent itemset Y such that X $\subseteq$ Y and supp(X)=supp(Y).

For example, if ***AB*** and ***ABC*** are two frequent itemsets with supp(AB)=supp(ABC), then ***AB*** is called *non-closed itemset.*

The set of all confident association rules created from frequent closed itemsets might still be very large. Therefore, the recent algorithms for mining microarrays mine only the maximal confident association rules, as in MAXCONF algorithm [9].

Definition 5 (Maximal Confident Association Rules). A confident rule r_1 is called *maximal confident association rule*, if ∄ other confident rule r_2 such that

1) antecedent(r_1) = antecedent(r_2), and
2) consequent(r_1) $\subset$ consequent(r_2).

For example, if the rules A → BCD and A → BC are confident, then A → BC is called *non-maximal confident association rule*.

In section 3, we review two algorithms for mining association rules related to our algorithm.

3 Related Works

Almost mining strong association rules algorithms face the following problems; the repeatedly scanning of the dataset and the explosion of generating all frequent itemsets. Apriori [5] algorithm based on column enumeration method did not solve these problems. Some modification and restrictions are made to Apriori algorithm for overcoming these problems. Max-Miner algorithm [16] mines only all maximal strong association rules, which can be fitted in memory, by pruning all the non-maximal frequent itemsets in early steps. But Max-Miner algorithm still scans the dataset many times to calculate the support of all frequent item sets. In the dense microarray datasets, Aprioir and Max-Miner enumerate the items (genes) which are larger than the transactions (samples). On the other hand, the algorithms like CHARM [11] and RERII [10] algorithms, are based on row enumeration method, overcome the problems of column enumeration method by listing the transactions rather than items. Their experimental results show that these algorithms outperforms than the column enumeration method when applied on microarray datasets. MAXCONF [9] algorithm is a modification of RERII algorithm by setting some restrictions to get only maximal high confidence rules; where recent paper [4] noted that "*A comparative analysis using several known microarray datasets revealed that without using any support threshold MAXCONF provide excellent results*". In this regards, we present the MAXCONF algorithm in the following subsection that is related to our approach for mining only maximal high confidence rules. Also, we present the advantages and disadvantages of this algorithm.

3.1 MAXCONF (Free Support-Based Pruning)

The MAXCONF algorithm [9] depends only on confidence pruning (free support pruning) to produce all maximal high confidence association rules without support pruning. The algorithm creates the rules with only one gene on the LHS; all rules on the form LHS→RHS, where |LHS|=1). MAXCONF uses the following two confidence pruning methods:

1. **Level 1 pruning**

Based on the row enumeration tree's structure, for each node in the tree, the maximum support can be predicted. The maximal support of a node is calculated by the initial support of a node plus the number of right sibling of this node. If maximum support of the *node<minconf*, then this node will be pruned.

2. **Level 2 pruning**

MAXCONF mines all maximal high confident rules, i.e., it prunes the non-maximal rules in earlier steps. I.e., if some rules are created from node n_i, then n_i goes to make some restrictions on its children to prevent the creation of non-maximal rules from them. But any node sibling to n_i may be produce non-maximal rules, this because n_i makes restrictions only on its children not on its siblings.

Advantages of MAXCONF

- Using the two pruning methods reduce the row enumeration space.
- It mines all common relationships and rare interesting relationships.
- It is faster than the row enumeration based algorithms RERII [10], and CARPENTER [13]. Also it is faster than a column enumeration based algorithms CLOSET+ [15] and CHARM [11]. As consequent, it is faster than Apriori [5] and Max-Miner [16] algorithms.

Disadvantages of MAXCONF

1- The intersection amongst the itemsets leads to time and space consumption.
2- It still produces non-maximal rules.

4 Mining Maximal High Confidence Rules Algorithm

This section introduces our *"Mining Maximal High Confidence Rules"* algorithm (*MMHCR*) that based on the column (gene) enumeration method. MMHCR algorithm uses only confidence pruning for mining maximal high confidence. The mined rules have the form LHS → RHS (conf ≥ minconf); |LHS|=1. The sample dataset in Table 2 is used as running example to illustrate the proposed algorithm; where *minconf* is set to be 50%. The overall structure of MMHCR algorithm consists of the following four consecutive steps:

4.1 Discretization

The normalized microarray dataset is usually represented as a series of continuous numbers. Discretization is the process of transformation from continuous data into discrete data. There are many discretization techniques [8, 17]. In this paper we use threshold methods in order to discretize data; where genes with expression values greater than a particular threshold are considered as *expressed*. Therefore in order to mine association rules, microarray matrix G is converted into matrix G' depending on the particular threshold cut value (*c*) [17]:

$$G'[i,j] = \begin{cases} 1 & G[i,j] \geq c, \quad (g_i \text{ is expressed}) \\ 0 & \text{otherwise} \end{cases}$$

For example, Table 2 shows the microarray dataset after discretized; where the threshold c=0.2. Based on this discretization, we can represent each gene with its binary representation. Fro example, the binary representation of gene *b* is 10011; this means that the gene *b* is expressed in samples 1, 4, and 5 but is non-expressed in samples 2 and 3.

4.2 Build Tree

MMHCR algorithm employes tree data structure to enumerate all genes in which the tree will have only three levels:

- **Level 1:** contains only the root of the tree. This root refers to all genes from microarray dataset that contains one field:
  - o ***Children***: refers to all genes at level 2.
- **Level 2:** contains set of nodes, each node contains one or more genes. The genes that are appeared (expressed) in the same samples will be combined into one node. Each node *N* at level 2 consists of the following fields:
  - o ***Antecedents_genes_set(ant_set)*****:** list of all genes' names that always appeared (expressed) together in the same samples. Each gene g_i∈ant_set will produce a rule on the form g_i→RHS; where RHS is set of genes will be created later.
  - o ***Consequents_Set (conseq_set)*****:** list of all genes' names that will be subset from the RHS of the created rules from node N.
  - o ***Binary_Representation_of_ant_set(BR)*****:** the binary representation of the node is equal to the binary representation of any gene belong to ant_set. I.e., N.BR=g_i.BR; g_i is any gene∈ant_set.
  - o ***Support_of_ant_set(supp)*****:** the number of ones in N.BR.
  - o ***Children_Set (children)***: contains set of nodes at level 3.
- **Level 3:** contains sets of nodes which represent all *children* of the nodes at level 2. Note that, all children nodes will be subset of RHS of the created rule. Each child node *C* at level 3 contains the following fields:-
  - o ***Consequents_genes_Set(conseq _set)*****:** the list of all genes' names that will be subset from the RHS of the created rules from its parent node N.
  - o ***Binary_Representation_of conseq_set(BR)*****:** a binary representation of the node= N.BR ∩ BR(g_i); N is the parent node and g_i is any gene ∈ C.conseq_set.
  - o ***Support_of_genes_set(supp)*****:** the number of ones in C.BR.
  - o ***Sub_Node(sub_node)*****:** if this field is set to be true, then this node will produce non-maximal rule (Definition 5). Therefore, this node will be pruned later after generating all its siblings (Definition 6).
  - o ***Participate (part):*** contains all the nodes' indices which participate to generate this node. For example, if node n_i and node n_j will be combined to form a new node called n_{ij}. Therfore, n_{ij}.part={i,j}.

Defintion 6 (**Sub_node**). A child node C_i at level 3 is called *sub_node* if one of the following two cases are hold:

1- ∃ child node C_k at level 3; C_k.BR ⊂ C_i.BR. i.e., the children nodes C_i and C_k are comined into the child node C_k. In this case, we set C_i.sub_node=true, C_k.consq_set= C_k.conseq_set ∪ C_i.conseq_set, and C_k.part=C_k.part∪{i}. I.e., child node C_i participates to form C_k. The created rule using the cild node C_i will be non-maximal rule.

2- ∃ child node C_k, C_i at level 3; d= C_i.BR ∩ C_k.BR and supp(d) / parent(C_i).supp ≥ mincof i.e., a new child node C_{ik} will be created; C_{ik}.BR=d, C_{ik}.supp=supp(d), and C_{ik}.conseq_set=C_i.conseq_set ∪ C_k.conseq_set. In this case, C_i.sub_node=true and C_k.sub_node=true. In addition, C_{ik}.part= C_{ik}.part∪{i,k}. i.e., C_i and C_k participate to form node C_{ik}.

4.3 Mining Maximal High Confidence Rules

To generate all maximal high confidece rules, MMHCR algorithm works as follows:

i. Scan the microarray dataset (Table 2) in which each line contains both a gene and its binary representation. MMHCR algorithm saves each line in a single node at level 2 in the tree.

ii. Sort the nodes in ascending order with respect to its support. The genes with the same binary representation will be combined into only one node (Fig. 1). For example: since genes *a* and *c* have the same binary representation, then the genes *a* and *c* are combined into single node with genes *ac*. The sort process will reduce the number of nodes at level 2. In Addition, the sort process will help us to easy find the nodes of level 3.

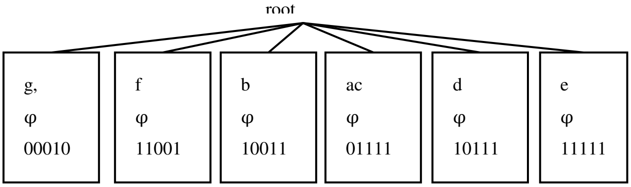


Fig. 1. The Sorted Column (gene) Enumeration Tree for Microarray Dataset

iii. For each node n_i at level 2 (Fig. 1); i=1,2,…, #(root.children)-1, we process the following three steps:

a. Create all children nodes of n_i that produce all high confidence rules

In order to generate all children nodes of n_i, the node n_i will be compared with its all right siblings n_k; k=i+1,…, #(root.children). Lemma 1 shows all cases of comparison between any two nodes at level 2. Fig. 2 and Fig. 3 show the pseudo code of Lemma 1; where for each sibling node n_k the procedure compare (n_i, n_k) in Fig.2 is invoked. Finally, Fig. 4 shows the complete tree after finish all comparisons.

Lemma 1. Let n_i and n_k be two nodes at level 2; d= n_i.BR ∩ n_k.BR, then one of the following cases holds:

1. *if node n_i ⊂ node n_k (n_i.BR=d). Then add n_k.ant_set to n_i.conseq_set.*

2. *if (supp(d) / n_i.supp) ≥ minconf; supp(d) = #ones in d. In this case,* n_i.children= n_i.children ∪ *newNodeAtLevel3 (n_k.ant_set, n_i.BR∩ n_k.BR, supp (d), false,φ).*

Note that, In each case of the Lemma 1, The procudeure compare (Fig. 2) checks the node n_k if it can be updated with node n_i as children or not.

For example, Fig. 4 shows that node n_1 with support 1 is subset from node n_3 with support 3; d= n_1.BR∩n_3.BR=00010 ∩ 10011= 00010 = n1.BR (n1 ⊂ *n3*), (i.e., the rule n_1→ n_2 (conf=100%)), then according to case 1 in Lemma 1 n_1.conseq_set= n_1.conseq_set ∪ n_3.ant_set = {b}. Similarly, node n_1 is subset from node n_4, n_5, n_6 n_1.conseq_set= {b} ∪ {ac} ∪ {d} ∪ {e} = abcde. The node n_4 with supp=4 is not equal n_5 with supp=4; d= n_4.BR ∩ n_5.BR=01111 ∩ 10111= 00111, supp(d) = 3, and supp(d) / n_4.supp= 3/4 = 75% ≥ minconf= 50% (i.e., the rule n_4→n_5 (conf=75%)), then new node n_p is created; n_p.conseq_set = n_5.ant_set={d}, n_p.BR = d = 00111. Also, supp(d) / n_5.supp = 3/4 = 75% ≥ minconf = 50% (i.e., the rule n_5 → n_4 (conf=75%)), then new node n_q is created; n_q.conseq_set = n_4.ant_set ={ac}, n_q.BR=d=00111. Finally, according to case 2 in Lemma 1, n_p and n_q are added to children of node n_4 and n_5 respectively.

```
procudure Compare(node $n_i$ , node $n_k$)
  d=$n_i$.BR ∩ $n_k$.BR;
  if($n_i$.BR = d) // $n_i$.BR ⊂ $n_k$.BR
     ni.conseq_set= $n_i$.conseq_set ∪ $n_k$.ant_set;

     conf=supp(d)/$n_k$.supp;
     if(conf ≥ minconf)
         $n_k$.children=$n_k$.children ∪ newNodeAtLevel3
                 ($n_i$.ant_set,d,supp(d),false,φ).

  else if(supp(d)/$n_i$.supp ≥ minconf)
      ni.children=$n_i$.children ∪ newNodeAtLevel3
                 ($n_k$.ant_set,d,supp(d),false,φ)

      conf= supp(d)/$n_k$.supp;
      if(conf≥minconf)
          $n_k$.children=$n_k$.children ∪ newNodeAtLevel3
                 ($n_i$.ant_set,d,supp(d),false,φ).
    end if
end if
end procedure.
```

Fig. 2. Procedure Compare

```
function newNodeAtLevel3 (conseq_set, BR, supp,
                         sub_node ,part):node at level 3.
    construct new child node C;
    C.conseq_set = conseq_set;
    C.BR = BR;
    C.supp = supp;
    C.sub_node = sub_node;
    C.part= part;
    return C;
end function
```

Fig. 3. Function Create New Node at Level 3

b. Create all children nodes of n_i that produce all maximal high confidence rules

The children nodes at level 3 will be compared with each other in order to produce only the nodes that will produce all maximal high confidence rules. Therefore, MMHCR algorithm makes the following three additional steps:

1. All children nodes of n_i are moved to children_buffer list (i.e., n_i.children will become empty).
2. The first node $n_1 \in$children_buffer will be inserted to n_i.children without any comparison.
3. Each node $n_k \in$children_buffer; k=2, 3, …, #(children_buffer) will be inserted in the right position in n_i.children. Therefore, node n_k is compared with all existed nodes n_j; $n_j \in n_i$.children and n_j.sub_node=false. Lemma 2 shows all cases of comparison between the two nodes n_k and node n_j. Fig. 5 shows the pseudo code of Lemma 2. [Note that, in case n_k.BR=n_j.BR, then n_j will be updated with genes of n_k]. Fig. 6 shows the final column enumeration tree of the dataset on Table 2.

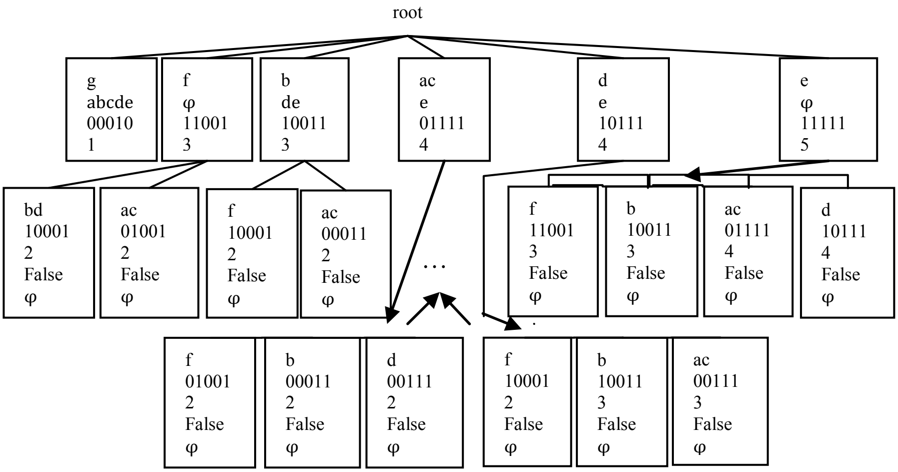


Fig. 4. The Tree After Generating All Children Nodes at Level 3

```
procedure Insert ($n_k$,$n_i$)
//$n_k$ will be added to $n_i$.children
add $n_k$ at position p in $n_i$.children
for each node $n_j \in n_i$.children do
  if ($n_k$.BR =$n_j$.BR) $n_j$.conseq_set=$n_k$.conseq_set; return;
  end if
end for
p= size($n_i$.children);//the paosition of new node $n_k$
newNodes: list of new created nodes"children".
for each node $n_j$ in $n_i$.children do
 if  ($n_i$.sub_node= false)
 d= $n_k$.BR ∩ $n_j$.BR;
 if (supp(d)= $n_k$.supp) // $n_k \subset n_j$
   $n_k$.conseq_set=$n_k$.conseq_set∪ $n_j$.conseq_set
   $n_k$.part =$n_k$.part∪{j} //$n_j$participate to from node
   $n_{k,}n_j$.sub_rule=true;  //nj becomes sub node
 else if (supp(d)=$n_j$.supp) // $n_j \subset n_k$
   $n_j$.conseq_set =$n_j$.conseq_set∪ $n_k$. conseq_set
   $n_j$.part =$n_j$.part∪{p}//k is the position nk
   $n_k$.sub_rule=true; $n_j$.sub_rule= false;  //update nj
 else if (supp(d)/$n_i$.supp>=minconf)
   node $n_{kj}$=newNodeAtLevel3(nj.conseq_set∪
   $n_k$.conseq_set,d,d.length,false,{j,p})
   $n_k$.sub_rule=true; $n_j$.sub_rule=true;
   insert($n_{kj}$,$n_i$). //recursive call
else if ($n_j$.part ≠φ)
   //the node cannot produce new node from $n_j$,then try to compare
   //it with all the nodes that participate to form $n_j$
   for each node $n_q \in n_j$.part do
    //call another procedure with the same 4 steps but without loop
      compare $n_q$ and $n_k$ using the 4 cases of this procedure.
   end if
 end if
end for
end procedure
```

Fig. 5. Procedure Insert

Lemma 2. Let a parent node n_i at level 2 and a node $n_k \in$ buffer_children, firstly n_k is added at position p in n_i.children, secondly, for each node $n_j \in n_i$.children; j=1, 2, ..., k-1; n_j.sub_node = false, and d=n_k.BR ∩ n_j.BR. The comparisons between n_k and all n_j have one of the following cases:

1- *If $n_k \subset n_j$, then*
   - *a node n_k will be updated with genes of n_j, n_j will be pruned after finish generate all children of n_i (i.e., n_j.sub_node=true), and add j to n_k.part.*

2- *If $n_j \subset n_k$ then*

- *a node n_j will be updated with genes of n_k, n_k will be pruned after finish generate all children of n_i (i.e., n_k.sub_node=true), and add p to n_j.part.*

3- *If supp(d)/n_i.supp ≥ minconf then*

- *Recursively add new node, n_{kj} , to n_i.children (again using the steps of this lemma), add the two indices and j to n_{kj}.part, n_k.sub_node= true, and n_j.sub_node=true.*

4- *Otherwise [n_k and n_j cannot be combined], repeat the Lemma 2 to compare node n_k with all the nodes at indices m; m∈n_j.part.*

For example, Fig. 6 (this example shows the case 4 in Lemma 2) shows that the two children nodes n_f and n_b at level 3 of the parent n_d; n_f.BR ⊂ *n_b.BR.* In this case, we set n_f.sub_node= true, n_b.conseq_aset=bf , and n_b.part={2}; 2 is the index of node n_f. The node n_{ac} with gene *ac* will be compared with all rules n_k; n_k.sub_node= false, then the node n_{ac} will be compared with the n_{bf} with genes *nf* only. But n_{ac} and n_{bf} cannot be combined, then n_{ac} will be compared with nodes at indices i; i∈n_{bf}.part. I.e., node n_{ac} will be compared with node n_f to form new node n_{acf} with genes acf; n_{acf}.part={2,3}; 2 and 3 are the indices of node n_f and n_{ac} respectively. Also, n_{ac}.sub_node=true, n_f.sub_node=true.

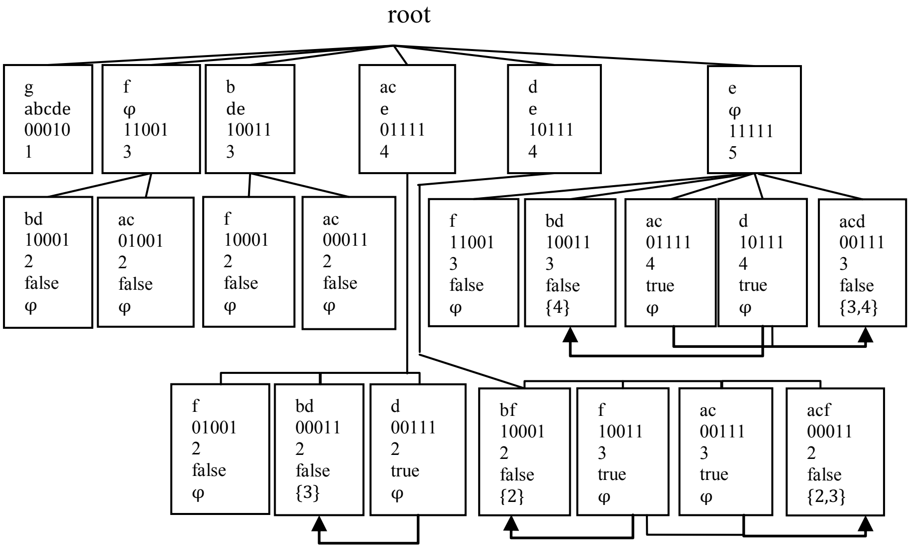


Fig. 6. Tree Contains All Nodes Which Mines Maxima High Confidence Association Rules

c. Generate all maximal high confidence rules of node n_i:

Finally, after all children nodes are created for node n_i (see Fig. 6). All nodes n_k∈ n_i.children with n_i.sub_node=true will be pruned, because these nodes will produce non-maximal rules. After that, each node n_k ∈ n_i.children with n_k.sub_node=false will produce all maximal high confidence rules of node n_i. Fig. 7 shows the procedure *GenerateMaximalHighRules* that generates all maximal high confidence rules of n_i. Fig. 8 shows all the extracted maximal high confidence rules from the tree in Fig. 6.

```
procedure GenerateMaximalHighRules($n_i$)
for each $c_j \in n_i$.childrendo
  for each $a_p \in n_i$.ant_set do
      Form a rule on the form $a_p \rightarrow$
      ($n_i$.ant_set-{$a_p$}) ∪ $n_i$.conseq_set ∪  $c_j$.
end procedure
```

Fig. 7. Procedure to Mine All Maximal High Confidence Rules

g→a b c d e:100.0%,	c→a e f:50.0%,
f→b d e:66.66 %,	c→a b d e:50.0%,
f→a c e:66.66% ,	d→b e f:50.0%,
b→d e f:66.66 %,	d→a b c e:50.0%,
b→a c d e:66.66 %,	e→f :60.0%,
a→c e f:50.0%,	e→b d:60.0%,
a→b c d e:50.0%,	e→a c d:60.0%.

Fig. 8. Extracted Maximal High Confidence Rules

4.4 The MMHCR Algorithm Overall Structure

Fig. 9 shows the overall structure of our proposed MMHCR algorithm.

```
MMHCR algorithm
INPUT: microarray dataset D, minconf.
OUTPUT: All maximal high confidence rules
Algorithm:
//the nodes with the same binary representation are combined.
    // step i and ii
    frequentTree=sorted list of nodes.
    //step iii
    for each node $n_i$ in frequentTree do
        //step iii a
        for each sibling node $n_j$ to $n_i$ do
                Compare($n_i$,$n_j$);
        end
        //step iii b
        buffer_children=$n_i$.children
        for each node $n_j \in$buffer_children do
                Insert ($n_j$, $n_i$);
        end
        //step iii c
        GenerateMaximalHighRules($n_i$);
        clear $n_i$ and their children
    end
end algorithm
```

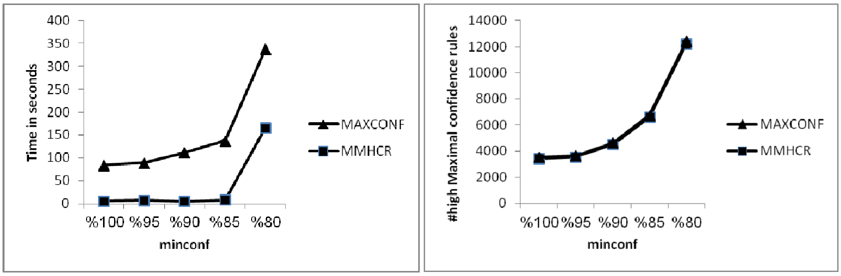
Fig. 9. MMHCR Algorithm

Advantages of MMHCR algorithm:

- Scan the dataset only once.
- The binary representation saves the memory and speeds up the intersection processes.
- The binary representation makes the measurements of confidence easier due to the dataset is scanned only once.
- The tree has 3 levels only; this saves both used space and time.
- It overcomes both the computational time and memory explosion problems of column-enumeration method. Also, it is better than row enumeration like MAXCONF [9] as consequent it is better than Apriori [5], Max-Miner [16], CHARM [11], and RERII [10].

5 Experimental Results

We present our experimental results that ran on PC with Intel(R) core 2 Duo 3.20 GHz, 8.00 GB of RAM, Windows 7 64 bit system using Java compiler JDK jdk-7u3-windows-x64 and netbeans-6.7.1-ml-windows IDE. The, MMHCR algorithm is compared only with the more related algorithm called *MAXCONF* [9] for mining maximal high confidence rules. The two algorithms are tested over the microarray dataset "Hughes et al 2000" of 300 samples and 6316 genes [18]. Fig.10 a. shows the running time in seconds of the two algorithms on the Hughese dataset. Fig. 10 b. shows the number of generated rules in both algorithms. It is clear that, the two algorithms produce the same number of the maximal high confidence rules. The comparative study shows that, MMHCR algorithm is more efficient and scalable than MAXCONF algorithm.



(a) Running Time (b) #Maximal High Confidence Rules

Fig. 10. MMHCR and MAXCONF Algorithms are applied on Hughes Dataset

6 Conclusion

In this paper, we have proposed a new algorithm called *MMHCR* based on column (gene) enumeration. MMHCR algorithm employed an efficient data structure called tree with three levels only; the tree is used to save the gene with its binary representation. Using the binary representation for each gene make the intersection processes easier and faster than the intersection processes between samples as in

MAXCONF algorithm. Moreover, the binary representation saved the used memory; where all association rules are fitted in the available memory. The experimental results on the real microarray dataset showed that our algorithm is an efficient and scalable than MAXCONF algorithm.

References

1. Stekel, D.: Microarray Bioinformatics. Cambridge University Press (2003)
2. Senthil Kumar, A.V.: Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains. In: InformatIon Science Reference (2011)
3. Wang, M., Shang, X.Q., Li, Z.H.: Strong Association Rules Mining without using Frequent Items for Microarray Analysis. In: The 3rd Int. Conf. on Bioinformatics and Biomedical Engineering (iCBBE 2009), pp. 978–984. IEEE, Beijing (2009)
4. Alves, R., Rodriguez-Baena, D.S., Aguilar-Ruiz, J.S.: Gene Association Analysis: a Survey of Frequent Pattern Mining from Gene Expression Data. Brief Bioinform (2010)
5. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proceedings of the 20th Int. Conf. on Very Large Data Bases (VLDB 1994), Santiago de Chile, Chile, pp. 475–486. Morgan Kaufmann (September 1994)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann (July 6, 2011)
7. Piatetsky-Shapiro, G., Tamayo, P.: Microarray Data Mining: Facing the Challenges. SIGKDD Explor. Newsl. 5(2), 1–5 (2003)
8. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.-F., Gandrillon, O.: Strong Association Rule Mining for Large Gene Expression Data Analysis: a Case Study on Human SAGE Data. Genome Biology 12 (2002)
9. McIntosh, T., Chawla, S.: High Confidence Rule Mining for Microarray Analysis. IEEE/ACM TCBB 4(4), 611–623 (2007)
10. Cong, G., Tan, K.-L., Tung, A., Pan, F.: Mining Frequent Closed Patterns in Microarray Data. In: Proc. Fourth IEEE Int'l Conf. Data Mining (ICDM), vol. 4, pp. 363–366 (2004)
11. Zaki, M.J., Hsiao, C.: CHARM: An Efficient Algorithm for Closed Association Rule Mining. In: Proc. SIAM Int'l Conf. on Data Mining, SDM (2002)
12. Agrawal, R., Imielinski, T., Swami, A.N.: Mining Association Rules between Sets of items in Large Databases. In: Proc. of the 1993 ACM SIGMOD Int. Conf. on Management of Data, pp. 207–216 (1993)
13. Pan, F., Cong, G., Tung, K., Yang, J., Zaki, M.J.: Carpenter: Finding Closed Patterns in Long Biological Datasets. In: Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining (KDD), pp. 637–642 (2004)
14. Cong, G., Xu, X., Pan, F., Tung, A., Yang, J.: FARMER: Finding Interesting Rule Groups in Microarray Datasets. In: SIGMOD 2004 (2004)
15. Wang, J., Han, J., Pei, J.: CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. In: Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, KDD (2003)
16. Bayardo, R.J.: Efficiently Mining Long Patterns from Databases. In: ACM SIGMOD Conf. Management of Data (June 1998)
17. Agrwal, J., Ramesh, J.C.: Analysis of Gene Microarray Data using Association Rule Mining. Journal of Computing 4(1) (January 2012)
18. Hughes, T., et al.: Functional Discovery via a Compendium of Expression Profiles. Cell 102, 109–126 (2000)

Partial Discharge Analysis and Inspection Alert Generation in High Power Transformers: A Case Study of an Autotransformer Bank at Eletrobrás-ELETRONORTE Vila do Conde Station

Carlos Takeshi Kudo Yasojima[1,*], Matheus Seribeli Furmigare[2],
Fernando de Souza Brasil[3], Terezinha Ferreira de Oliveira[2],
and Antonio Morais da Silveira[1]

[1] Science Computing Faculty, Pará Federal University, Belém, Brazil
takeshiyasojima@gmail.com, morais@ufpa.br
[2] Statistics Faculty, Pará Federal University, Belém, Brazil
furmigarems@gmail.com, tfo@ufpa.br
[3] Eletronorte Technology Center, Belém, Brazil
fernando.brasil@eletronorte.gov.br

Abstract. This paper presents an exploratory study using statistical and IA techniques in the partial discharge database located in Vila do Conde substation, Barcarena, Pará state, Brazil, ELETRONORTE property. Through ambiental and system variables analysis, it was possible to identify that the 230kV reactive power and period of day have a strong relation to partial discharge measures. With the obtained knowlegde and specialists knowlegde, a initial fuzzy system is proposed for partial discharge classification in diferents operational situations of alert, contributing to the operational status diagnosis of power transformers and amplifying the knowledge about the theme.

Keywords: Partial discharges, Fuzzy Logic, Reactive Power.

1 Introduction

In the electric system, problems in energy supply are common, caused by equipment failures like power transformers, or when performing maintenances, causing losses for companies and consumers. Therefore, it is important to perform periodic and preventive diagnostics to provide a quality service. Researches and experiments indicates that partial discharge phenomenon occurs when there are defects in the equipment[1][2].

Partial discharges (PD) are electrical discharges located at the junction between two conductors through the insulation, which may or may not occur near a conductor [3]. These phenomena are caused by the rupture of localized dielectric strength of the insulating material, which is characterized as one of the possible sources of faults in electrical insulation [4, 5].

[*] Corresponding author.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 367–378, 2013.

In recent years, much has been done to develop and improve the system for detecting partial discharges. Researchers have studied techniques and developed analysis tools to identify patterns in PD data as shown in[6]. Different methods have been developed to select the features that can provide information relevant to the recognition of patterns in PD, such as statistical methods, the analysis of the shape of the pulses, signal processing, image processing and computational intelligence techniques.

Statistical methods, eg, statistical moments [7], independent component analysis (ICA) [8, 9], principal curve analysis (PCA) [10] correlation techniques and simple [11] and multiple [12] linear regression have been used, but also multivariate techniques such as the principal component analysis, factor analysis and cluster analysis have been used to study features of partial discharge[13]. Using computational intelligence techniques, authors [14, 15] have proposed solutions using ANFIS (Adaptive Neuro-Fuzzy Inference System) and neuro-fuzzy techniques, for the recognition and classification of 4 types of partial discharge patterns.

In recent publications, authors [16] and [17] proposed a fault diagnosis method for high power transformers using neural network technique on a partial discharge database from dissolved gas analysis technique, which indicates transformers status and operability.

In this sense, this paper presents the results of an exploratory study using a database on seasonal partial discharges of Eletronorte-Eletrobras, at Vila do Conde Power station. The database was originated from acoustic emission technique, which is an alternative to dissolved gas analysis technique. A fuzzy system is also proposed, based on the knowledge acquired in the research, for classification of partial discharges in different situations, ranging from normality to the generation of inspections alerts on the equipment.

2 Materials and Methods

2.1 Study Area

Vila do Conde is a Brazilian power station is located in Barcarena, a city of Pará state and belongs to the company Centrais Elétricas do Norte do Brasil S/A (Eletronorte) where a database of partial discharge measures from autotransfomers is studied. Autotransformers aims to transform the energy received from 500 kV to 230 kV. Eletronorte has a measurement system that captures values of partial discharges at the autotransformer input (500 kV) and output (230 kV), to obtain indices indicating equipment status and operability .The system operates at a risk, since the loss of one or two 500/230 kV MVA autotransformers may cause an overload of the remaining autotransformers and generate serious consequences, such as the general shutdown of the Vila do Conde substation and, therefore, an interruption in the State power supply.

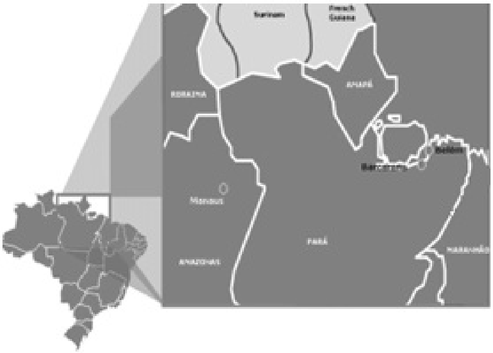


Fig. 1. Location of Barcarena – Pará

2.2 Exploratory Analysis of the Data

Data was collected using the internal Eletronorte - Eletrobras system and the Sinda website (National Environmental Data) during the period from 01/05/2012 to 22/12/2012, totalling 1336 partial discharge measurements in picocoulomb(pC), obtained every three hours. The other variables were also obtained:

- Current of phases A, B, V in 230 kV and 500 kV;
- Voltage of phases AB, BV, VA in 230 kV and 500 kV;
- Active power in 230 kV and 500 kV;
- Reactive Power in 230 kV 500 kV;
- Oil temperature of phases A, B and V;
- Winding Temperature of phases A, B in 13 kV, 230 kV and 500 kV;
- Frequency in 230 kV and 500 kV;
- Power Factor in 230 kV and 500 kV.

The following environmental data were also obtained by means of the Sinda website:

- Rainfall Index
- Air Temperature
- Humidity

2.3 Cluster Analysis

This technique seeks to identify clusters (or families) formed by similar sets. The clustering algorithm used was the k-means, which is based on the averages (centroids) of the clusters minimizing intra-cluster variance and maximizing inter-cluster variance [18] .The folowing variables correlated with partial discharges were used when applying this technique: reactive power (230 kV), temperature (oC) and relative humidity.

2.4 Correpondence Analysis

The correspondence analysis is a multivariate technique used to examine the associations between nominal variables containing several categories. This tool was used to examine possible associations between the intensity of the partial discharges and the time period, that was divided into class intervals. This technique allows a graphical analysis, in a multidimensional space, of the association between variables by means of a contingency table for the calculation of inertia, which is the weighted sum of all distances from the centroid by summing all cells in the table. To investigate the suitability and best interpretation of the correspondence analysis the contingency coefficient C and residual Chi-square analysis were used [19].

2.5 Fuzzy System

Fuzzy logic is a computational intelligence technique that handles verbal expressions, inaccurate, qualitative and inherent in human communication, which have varying degrees of inaccuracy, translating the values in fuzzy terms understandable by computers [20]. These values are placed in an expression with a certain degree of relevance, always in an interval of [0,1], where 1 is the highest possible relevance.

Generally, a fuzzy system consists of four components [21, 22]: the fuzzyfier, that converts the real values of entry into a degree of membership in fuzzy sets to be processed by the fuzzy inference machine, the rule base, which consists a set of rules based on the system inputs and outputs, the fuzzy inference engine, which uses the principles of fuzzy rules for combining the existing fuzzy rule base in a mapping of a fuzzy set input to an fuzzy set output, and the defuzzyfier, that maps a fuzzy set, obtained in the inference engine, into a real value (output).

The proposed fuzzy system was implemented using the Matlab Fuzzy Logic Toolbox 7.0 [23] based on the knowledge gained from the statistical analysis of the investigated database as shown in section 3.

3 Data analysis and Results Discussion

3.1 Knowledge Acquired Through the Statistical Techniques

In the clustering process we used the following variables correlated with partial discharges: reactive power (230kV), temperature (oC) and relative humidity, where it was possible to obtain four clusters in pC: PD <500, 500 $\leq$ PD <800 , 800 $\leq$ PD < 1200 and PD $\geq$ 1200. The analyses were performed using the Statistical software package[23].

Table 1 shows the contingency table for the partial discharges by period. The C test from the correspondence analysis was significant, with p = 0.000.

Table 1. Contingency table of partial discharges at the transformer output

Period	PD<500	500≤PD<800	800≤PD<1200	PD≥1200	Total
LateNight	189	53	38	61	341
Morning	206	51	36	41	334
Afternoon	270	32	11	10	323
Night	214	59	35	30	338
Total	879	195	120	142	1336

The circles in Figure 2 show the highest degrees of associations between the time of day and the group of partial discharges. The residue analysis validated the chi-square analysis (Table 2), and it can be observed that the greatest association occurred in the afternoon with PD < 500 pC, followed by late night with PD > 1200 pC.

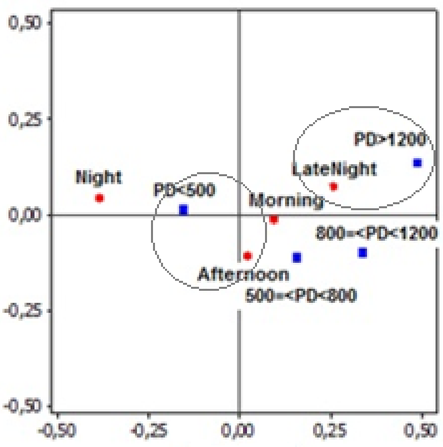


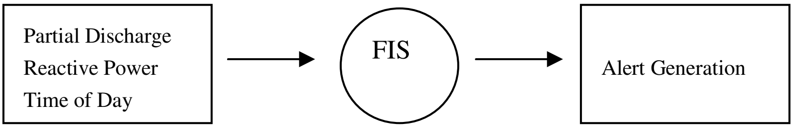
Fig. 2. Graph representing the chi-square residue analysis

Table 2. Analysis of the chi-square residue analysis

Period	PD<500	500≤PD<800	800≤PD<1200	PD≥1200
LateNight	-0.93	0.32	1.10	0.92
Morning	-0.56	1.38	0.84	-0.99
Afternoon	3.94	-2.21	-3.34	-4.15
Night	-2.36	0.46	1.33	4.11

3.2 Fuzzy System

Based on the correspondence analysis and chi-square residue analysis results and information on previous research by other experts in the field[4], it was possible to identify patterns of behavior for 3 variables (Reactive Power 230kV, Time of Day and Partial Discharge), which allowed for the construction of a fuzzy system to generate alerts inspections on the analyzed equipment. The behavior patterns of these variables were translated to the fuzzy approach by means of Figures 3, 6 and 7 respectively.



Reactive Power

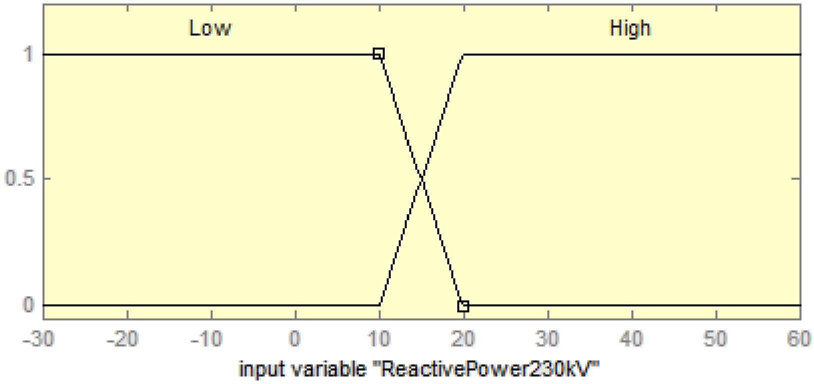


Fig. 3. Reactive Power 230kV

The histogram of Figure 4 implies that the dominance of the reactive power varies in the range of -50 to 60.

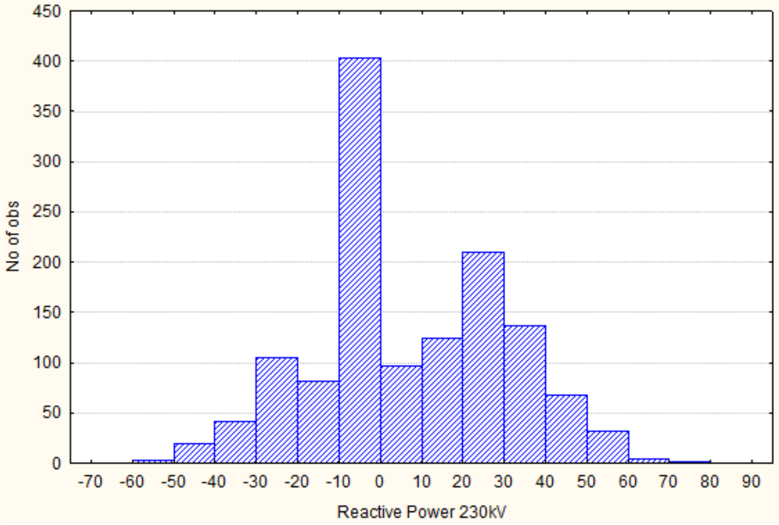


Fig. 4. Histogram of Reactive Power 230 kV

The line in Figure 5, relating to the linear regression, indicates the relationship of increased reactive power with the partial discharge. However, we can observe that there are points indicating high values for low values of discharges reactive power and vice versa.

When the reactive power reaches above 20, the frequency of partial discharges above 500pC intensifies, so when this limit is reached, the high range of the fuzzy set of Figure 1 is considered *high* with maximum relevance.

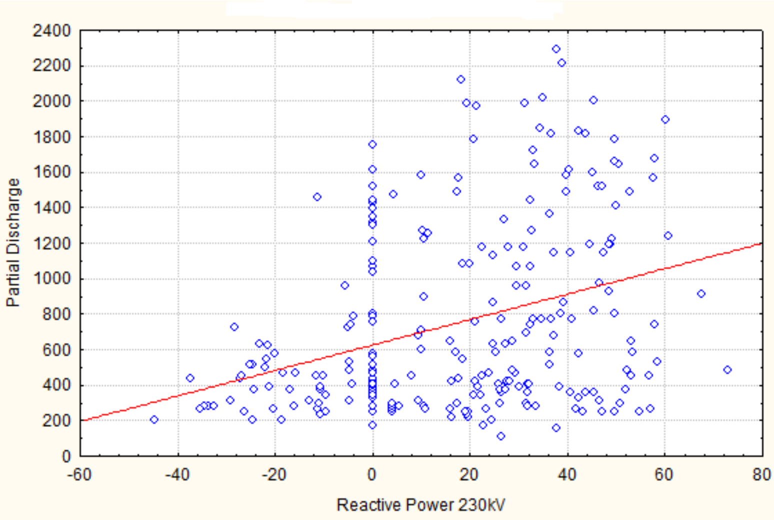


Fig. 5. Scatterplot with Linear Regression between Partial Discharge and Reactive Power 230kV

Time of Day

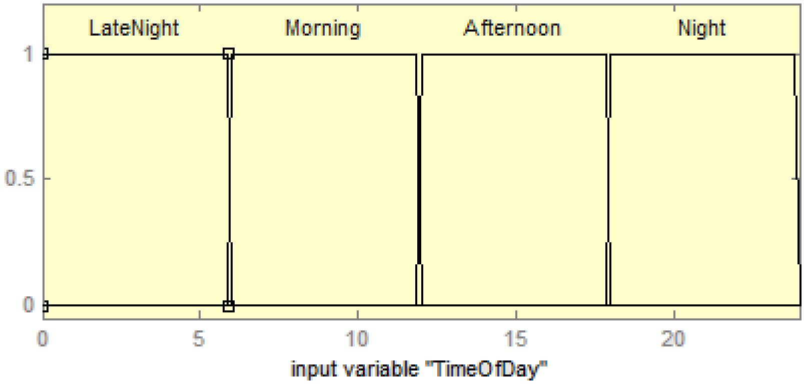


Fig. 6. Time of Day

The variable of Figure 6 has the highest relevance throughout the time of day for all intervals. Because the data are divided into 6 hour intervals, there are only 2 possibilities of values, as follows:

- LateNight: times between 0:00 and 6:00
- Morning: times between 6:00 and 12:00
- Afternoon: times between 12:00 and 18:00
- Night: times between 18:00 and 24:00

Partial Discharge

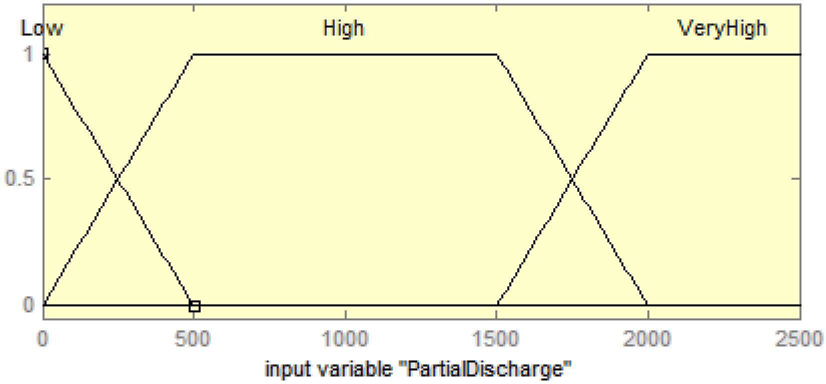


Fig. 7. Partial discharge

According to the ABNT NBR 5356 norm [4], in order to consider a partial discharge safe, it must be below the 500pC represented by the fuzzy set as *low* in the variable of Figure 7.

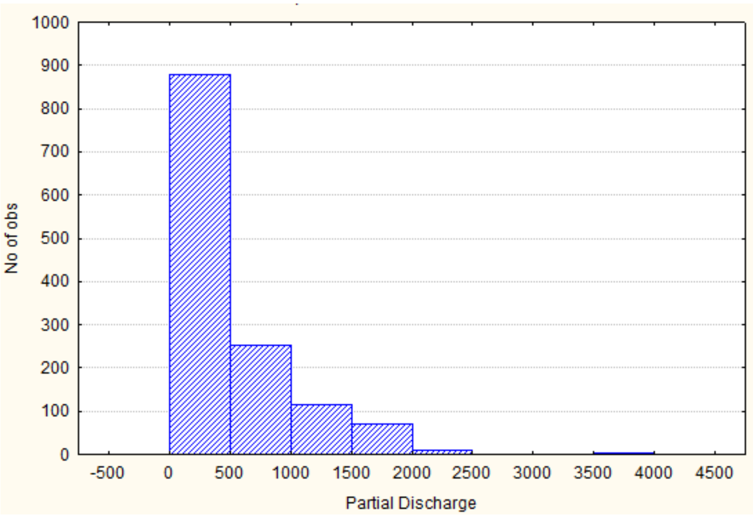


Fig. 8. Histogram of partial discharge

A third fuzzy set, *VeryHigh*, was added to represent unusual discharge values, as shown in the histogram of Figure 8, where in rare occasions discharges occur above 2000pC.

Alert Situations

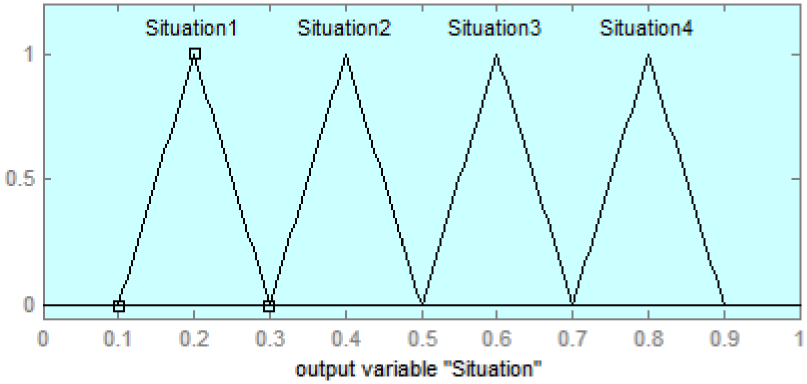


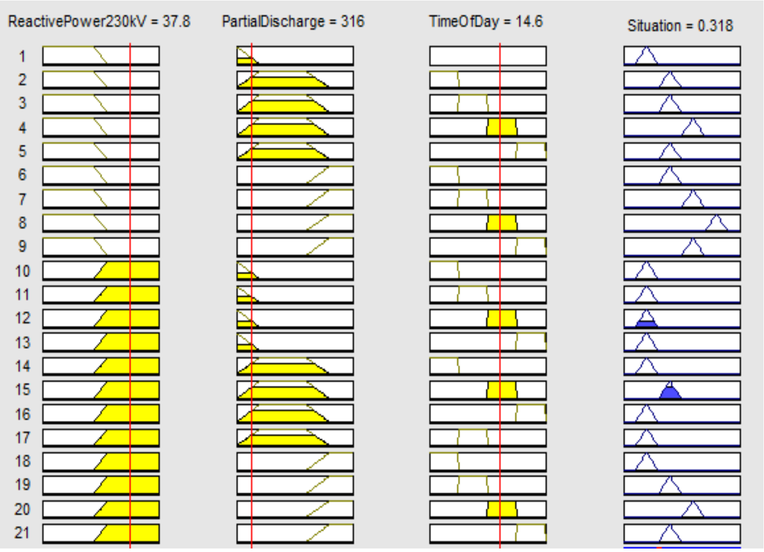
Fig. 9. Alert situations

During the exploratory data analysis, we found some unfamiliar situations not recognized by experts in the field. Using the tacit knowledge of experts from the Eletronorte laboratory, it was possible, from the variables of Reactive Power 230kV, Time of Day and Partial Discharge, to propose a classification of partial discharges grouped into four (4) sets of conditions (Figure 9), which allowed the following Base construct production rules:

Rule base examples used in the present study

1. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *low*) then (*Situation* is *Situation1*);
2. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *LateNight*) then (Situation is *Situation2*);
3. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *Morning*) then (*Situation* is *Situation2*);
4. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *Afternoon*) then (*Situation* is *Situation3*);
5. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *Night*) then (*Situation* is *Situation2*);
6. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *VeryHigh*) an (*TimeofDay* is *LateNight*) then (*Situation* is *Situation2*)
7. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *Morning*) then (*Situation* is *Situation3*)
8. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *Afternoon*) then (*Situation* is *Situation4*)

9. If (*ReactivePower230kV* is *Low*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *Night*) then (*Situation* is *Situation3*)
10. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *Low*) and (*TimeofDay* is *LateNight*) then (*Situation* is *Situation1*)
11. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *Low*) and (*TimeofDay* is *Morning*) then (*Situation* is *Situation1*)
12. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *Low*) and (*TimeofDay* is *Afternoon*) then (*Situation* is *Situation1*)
13. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *Low*) and (*TimeofDay* is *Night*) then (*Situation* is *Situation1*)
14. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *LateNight*) then (*Situation* is *Situation1*)
15. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *Afternoon*) then (*Situation* is *Situation2*)
16. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *Night*) then (*Situation* is *Situation1*)
17. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *High*) and (*TimeofDay* is *Morning*) then (*Situation* is *Situation1*)
18. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *LateNight*) then (*Situation* is *Situation1*)
19. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *Morning*) then (*Situation* is *Situation2*)
20. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *Afternoon*) then (*Situation* is *Situation3*)
21. If (*ReactivePower230kV* is *High*) and (*PartialDischarge* is *VeryHigh*) and (*TimeofDay* is *Night*) then (*Situation* is *Situation2*)



ReactivePower230kV = 37.8

PartialDischarge = 316

TimeOfDay = 14.6

Situation = 0.318

Fig. 10. System results for a situation

Figure 10 shows the result of the situation for an event consisting of the following values: Reactive Power 230KV = 37.8, Partial Discharge = 316 and Time of Day = 14.6. It can be observed that the value of 0.318 for the variable Situation signifies a state transition from situation 1 to situation 2 on the scale of inspection alerts situations.

4 Conclusion

The study results demonstrate that it is possible to construct a proposed approach using fuzzy computational intelligence, to help expand the knowledge of specialist in the fields of control and diagnosis of transformers operational status in Electric Power Systems. A Cluster analysis associated with a Correspondence Analysis allowed to establish groups of discharges and their association with times of day, determining the periods where the occurrence of discharges are higher or lower.

This approach can be a valuable tool for use in solving problems in environments of uncertainty, such as the process of decision-making regarding the maintenance needs of high power transformers. This approach enables the means to diagnose the operating condition of the equipment prior to physical defects that might appear and, thereby, guide the construction plan of preventive maintenance to avoid possible equipment failures. The prediction of preventive inspection alerts supported in this study of partial discharge is an inexpensive monitoring method.

In future works, a comparison of this proposed method results and other known techniques to diagnose transformers, will be made for testing and validation.

References

1. Bartnikas, R.: Partial discharges: Their mechanism, detection and measurement. IEEE Transactions on Dielectrics and Electrical Insulation 9, 763–808 (2002)
2. Montanari, G.C.: Partial discharge measurements: becoming a fundamental tool for quality control and risk assessment of electrical systems? In: IEEE International Symposium on Electrical Insulation, pp. 281–285 (2006)
3. McArthur, S.D.J., Strachan, S.M., Jahn, G.: The design of a multi-agent transformer condition monitoring system. IEEE Transactions on Power Systems 19, 1845–1852 (2004)
4. ABNT. NBR 5356: Transformadores de Potência, parte 1: Generalidades (2007)
5. IEC60270: High-voltage test techniques - Partial discharge measurements (2000)
6. Sahoo, N.C., Salama, M.A., Bartinikas, R.: Trends in Partial Discharge Pattern Classification: A Survey. IEEE Transactions on Dielectrics and Electrical Insulation 12, 248–264 (2005)
7. James, R.E., Phung, B.T.: Development of Computer-based Measurements and their Application to PD Pattern Analysis. IEEE Transactions on Electrical Insulation 2, 838–856 (1995)
8. Diniz, F.C.C.B.: Supressão de Ruído, Detecção e Classificação de Sinais de Descargas Parciais em Transformadores de Potência. In: Engineering Department (COPPE). Universidade Federal do Rio de Janeiro, Rio de Janeiro (2005)

9. Cuenca, W.M.H.: Caracterização dos Sinais de Descargas Parciais em Equipamentos de Alta Tensão a Partir dos Modelos Experimentais. In: Engineering Department (COPPE). Universidade Federal do Rio de Janeiro, Rio de Janeiro (2005)
10. Faier, J.M.: Curvas Principais Aplicadas na Identificação de Descargas Parciais em Equipamentos de Potência. In: Engineering Department (COPPE). Universidade Federal do Rio de Janeiro, Rio de Janeiro (2006)
11. Jeyabalan, J., Usa, S.: Statistical Techniques for Partial-Discharge Location in Transformer Windings. IEEE Transactions on Power Delivery 26, 2064–2065 (2011)
12. James, R.E., Jones, S.L.: Some Aspects of the Statistical Modeling of Partial Discharge Inception Conditions. IEEE Transactions on Electrical Insulation 23, 297–306 (1988)
13. Liao, R.-J., Yang, L.-J., Li, J., Grzybowski, S.: Aging Condition Assessment of Transformer Oil-paper Insulation Model based on Partial Discharge Analysis. IEEE Transactions on Dielectrics and Electrical Insulation 18, 303–311 (2011)
14. Guo, C., Zang, L., Qian, Y., Wang, H., Yao, L., Jiang, X.: Application of adaptive neuro fuzzy inference system to the partial discharge pattern recognition. Intelligent Computing and Intelligent Systems, 729–732 (2009)
15. Fard, M.A., Akbari, A., Shojaee, R., Mirzaei, H.R., Naderi, P.: Partial discharge defects classification using neuro-fuzzy inference system. Solid Dieletrics, 1–4 (2010)
16. Silva, A.C.M., Castro, A.R.G., Miranda, V.: Transformer failure diagnosis by means of fuzzy rules extracted from Konohen Self-Organizing Map. Electrical Power and Energy Systems 43, 1034–1042 (2012)
17. Miranda, V., Castro, A.R.G., Lima, S.: Diagnosing Faults in Power Transformers with Autoassociative Neural Networks and Mean Shift. IEEE Transactions on Power Delivery 27, 1034 (2012)
18. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice Hall, New Jersey (1992)
19. Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E., Tathan, R.L.: Multivariate Data Analysis. Prentice Hall, New Jersey (2005)
20. Turban, E., Aronson, J.E.: Decision Support Systems and Intelligent Systems. Prentice Hall, New Jersey (2001)
21. Wang, L.: A Course in Fuzzy Systems and Control. Prentice Hall International Inc., New Jersey (1997)
22. D'Errico, G.E., Murru, N.: Fuzzy treatment of candidate outliers in measurements. Advances in Fuzzy Systems, 1–6 (2012)
23. MATLAB: MATLAB - The Language of Technical Computing

Smart Meter Data Analysis for Power Theft Detection

Daniel Nikolaev Nikovski[1], Zhenhua Wang[1], Alan Esenther[1], Hongbo Sun[1],
Keisuke Sugiura[2], Toru Muso[2], and Kaoru Tsuru[2]

[1] Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA
[2] Mitsubishi Electric Corporation, 5-1-1, Ofuna, Kamakura, 247-8501, Japan

Abstract. We propose a method for power theft detection based on predictive models for technical losses in electrical distribution networks estimated entirely from data collected by smart meters in smart grids. Although the data sampling rate of smart meters is not sufficiently high to detect power theft with complete certainty, detection is still possible in a statistical decision theory sense, based on statistical models estimated from collected data sets. Even without detailed knowledge of the exact topology of the distribution network, it is possible to estimate a statistical model of the technical losses that allows indirect estimation of the non-technical losses (power theft) with high accuracy.

1 Introduction

The power grids of many countries are currently undergoing radical upgrades, and are increasingly equipped with massive sensing and communication infrastructure that can significantly improve the measurement and control capabilities of the resulting "smart" grids. This infrastructure includes devices such as phasor measurement units (PMUs) and smart power meters that are installed at many locations and/or measure data very frequently, resulting in very high-volume data streams. Using these data streams for various decision problems has opened up new opportunities for the application of data analytical algorithms and techniques.

One such decision problem of critical importance to electrical power utilities is the reliable detection of power theft [1]. It exists in practically all countries, but in some markets, for example Southeast Asia, the amount of theft can even exceed 40% [2-3]. The most common type of power theft occurs when an illegal user draws power directly from the power lines, between the distribution transformer (DT) and any electrical power meter that can measure electricity consumption. In general, the mismatch between the total energy supplied by the distribution transformer and the sum of energy consumed by all legal paying end users (EU) can be detected, and the total amount of losses in the distribution sub-network can be estimated. However, this amount includes both technical losses that are inevitable during the normal operation of the power distribution system, and non-technical losses (theft). Technical losses comprise ohmic losses in the electrical lines due to the resistance of the lines, conversion losses in any intermediate devices, leaks due to imperfect isolation, etc. Because some of these components of the technical losses depend on the amount of power being

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 379–389, 2013.

delivered to customers, and that amount varies significantly throughout the day, week, and year, it is generally difficult to decide what part of the total loss is technical, and what part might be due to theft.

It would be possible to calculate accurately the exact amount of technical losses if all the parameters of the distribution network were known, including its connection topology, order and attachment points of all users, the line resistances between the attachment points, as well as the instantaneous power consumption by every user at any moment in time. In practice, full knowledge of these parameters is not economically feasible - a power utility would normally know which user is served by which distribution transformer from its geographic coverage plan, but the connection order and exact line resistances would not normally be known. In addition, full knowledge of the power consumption by any user at any instant in time would only be possible by installing detailed measurement equipment, such as PMUs, that performs very frequent measurements (multiple times per second). However, such an installation would be prohibitively expensive, its cost far exceeding the cost of power theft. In practice, utilities collect only infrequent, average and/or aggregated measurements, usually over a fairly long period of time - one month for traditional power meters, and 30 to 60 minutes for the new generation of smart meters that have advanced telemetry functions. The most important measurement available to power utilities is the total amount of active power consumed by a user during the measurement period, because this value is the basis on which payment by the customer is determined. Additional variables provided by some meters, for example smart meters conforming to the ANSI C12.19 standard, comprise reactive power consumed by the user (important for billing of some industrial customers), instantaneous voltage and current at the beginning and end of the measurement interval, power quality information, etc.

One popular theft detection method, implemented in most meter data management systems (MDMS), is to estimate the amount of total losses as described above, by performing energy balance between the energy supplied by the DT and the energy consumed by all metered users, and calculate the loss rate as a percentage of the total amount provided. When this loss rate exceeds a specified threshold, e.g. 3%, theft can be suspected and investigated. A disadvantage of this method is that no distinction is made between technical and non-technical losses, so when technical losses are unusually high for perfectly good and legal reasons, for example very uneven power consumption, a power theft even can be detected erroneously.

The advent of smart meters, with their much more frequent sampling intervals, has made it possible to calculate loss rates at a much finer temporal scale, and possibly detect power theft events of much shorter duration that would otherwise get lost in the much larger monthly energy aggregates measured and reported by traditional power meters. However, if power theft is not an irregular, one-time event, but is systematic and follows similar consumption patterns as those of the legal electricity users, the finer temporal scale of analysis would not improve detection significantly, because it would be computing the same loss rate, only more frequently. So, if the same loss rate calculation method is applied on smart meter data, higher accuracy of detection could not be expected for the economically more significant case of long-term systematic power theft. In order to use more productively the much larger data sets produced by smart meters, more advanced detection algorithms are needed. Section 2 proposes one such method, Section 3 describes some experimental results using a detailed network simulator, and Section 4 concludes and proposes direction for further improvement in the accuracy and reliability of the method.

2 Power Theft Detection Based on a Technical Loss Model

We consider the problem of estimating the non-technical losses (NTL) in a branch of a distribution network consisting of a distribution transformer (DT) connected to a sub-station by means of a feeder, and a number of users connected to the secondary side of the DT in some manner (Fig. 1). The total amount of losses L_k in such a network over a particular time interval k can easily be estimated by performing the energy balance between the energy $E_{DT,k}$ supplied by the DT during this time interval and the sum of the energy E_k^i consumed by each user i, as measured by each smart meter:

$$L_k = E_{DT,k} - \sum_{i=1}^{n} E_{i,k} \quad . \tag{1}$$

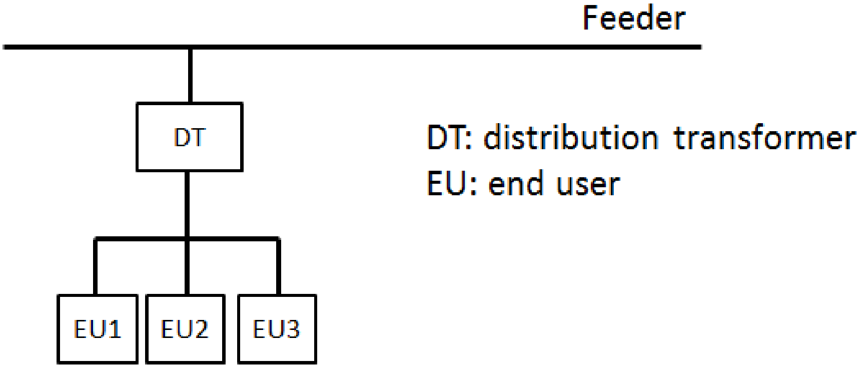


Fig. 1. Diagram of a distribution system. Power theft occurs when one of the end users (EU) is attached to the distribution transformer and draws power from it, but its consumption is not measured and paid for.

If we can estimate the amount $\hat{L}_\mathrm{k}^\mathrm{TL}$ of technical losses (TL) during the same interval, we can indirectly compute the amount $\hat{O}_k$ of NTL as $\hat{O}_k = L_\mathrm{k} - \hat{L}_\mathrm{k}^\mathrm{TL}$, and since NTL are usually attributed to power theft, a decision of whether to investigate can be based on this estimate. That is, our approach is to reduce the problem of estimating NTL to that of estimating TL.

One of the biggest problems in estimating the amount of TL $\hat{L}_\mathrm{k}^\mathrm{TL}$ is that, in general, the connection pattern between the end electricity users and the distribution transformer is not known. This connection pattern includes the topology and length of the power lines between the DT and EU. The most typical connection pattern is by means of a feeder connecting the secondary side of the transformer to individual users attached at various points along the feeder. The exact location of these points is not known, and the resistances of the line between these points are not known, either.

In the absence of detailed information about the actual circuit of the branch of the distribution system, we make the simplifying assumption that it has a specific topology and connectivity pattern, shown in Fig. 2, represented as a one-line diagram [6]. We assume that each user (1101 to 1170) is connected to the secondary side of the transformer (the bus 1100) by means of an individual line. (When a user is connected to multiple phases of the distribution system, we treat each phase as a separate and independent user.) Because energy balance and loss modeling is performing independently for each phase in a multi-phase system, below we describe the model for a single phase only.

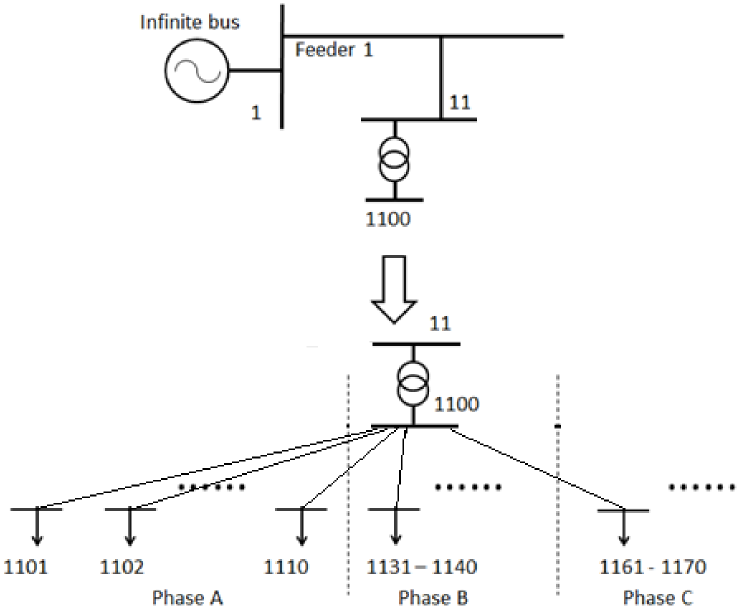


Fig. 2. Approximating circuit of a distribution system, represented as a one-line diagram. After the distribution transformer, all users are attached to the same bus 1100, through independent lines of varying resistance, one per user.

We define the following variables for the simplified circuit:

R_i The actual resistance of the line to user i

$\hat{R}_i$ The estimated resistance of the line to user i

$I_{i,k} = I_i(t_k)$ The measured instantaneous current of branch i at the end of time interval k

$L_{i,k}$ Actual technical loss of branch i during time interval k

$\hat{L}_{i,k}$ Estimated technical loss of branch i during time interval k

L_k Total loss during time interval k for all branches (users), obtained by means of power balance between the DT and all legal users

L_k^{TL} Technical loss during time interval k for all branches (users). When there is no theft, $L_k^{TL} = L_k$

l_0 Non-ohmic technical loss (time independent)

$\hat{O}_k$ The estimated non-technical loss (NTL) during time interval k

For smart meters, the measurement time interval is typically equal to 30 minutes, and for traditional meters, it could be equal to one month or longer.

The ohmic losses during measurement period k due to the resistance of the transmission line to user *i* are

$$L_{i,k} = \int_{t_{k-1}}^{t_k} I_i(t)^2 R_i dt \tag{2}$$

In practice, as noted above, we will know neither the actual resistance R_i of branch *i*, nor the instantaneous current $I_i(t)$ at all instants between times t_{k-1} and t_k. For this reason, we will make the additional simplifying assumption that the relation between current magnitude and time is piecewise linear:

$$\hat{I}_i(t) = s_{i,k}t + I_i(t_{k-1}), \tag{3}$$

where:

t time, $t_{k-1} < t < t_k$

$s_{i,k}$ slope, $s_{i,k} = \frac{I_i(t_k) - I_i(t_{k-1})}{t_k - t_{k-1}}$

We rewrite equation (2) as

$$\hat{L}_{i,k} = \frac{1}{s_{i,k}} \int_{I_i(t_{k-1})}^{I_i(t_k)} \hat{I}_i(t)^2 R_i dI_i(t) = \frac{R_i}{3s_{i,k}} [I_i(t_k)^3 - I_i(t_{k-1})^3] \tag{4}$$

The total loss then is:

$$\hat{L}_k = \sum_{i=1}^{n} \frac{I_i(t_k)^3 - I_i(t_{k-1})^3}{3s_{i,k}} R_i + l_0 \ , \tag{5}$$

where:

n The number of smart meters

The estimates of the branch resistances $\hat{R}_i$ can then be obtained by means of the least squares method, for example using Moore–Penrose pseudoinverse:

$$\hat{R} = (H^T H)^{-1} H^T L \ , \tag{6}$$

where:

$$H = \begin{bmatrix} \frac{I_1(t_2)^3 - I_1(t_1)^3}{3s_{1,2}} & \frac{I_2(t_2)^3 - I_2(t_1)^3}{3s_{2,2}} & \dots & \frac{I_n(t_2)^3 - I_n(t_1)^3}{3s_{9,2}} & 1 \\ \frac{I_1(t_3)^3 - I_1(t_2)^3}{3s_{1,3}} & \frac{I_2(t_3)^3 - I_2(t_2)^3}{3s_{2,3}} & \dots & \frac{I_n(t_3)^3 - I_n(t_2)^3}{3s_{9,3}} & 1 \\ \dots & \dots & \dots & \dots & 1 \\ \frac{I_1(t_m)^3 - I_1(t_{m-1})^3}{3s_{1,m}} & \frac{I_2(t_m)^3 - I_2(t_{m-1})^3}{3s_{2,m}} & \dots & \frac{I_n(t_m)^3 - I_n(t_{m-1})^3}{3s_{n,m}} & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} L_2 \\ L_3 \\ \dots \\ L_m \end{bmatrix}$$

$$\hat{R} = \begin{bmatrix} \hat{R}_1 \\ \hat{R}_2 \\ \dots \\ \hat{R}_n \\ l_0 \end{bmatrix}$$

m the number of measurement periods

The free term l_0 represents the non-ohmic losses, that is, the losses that are not caused by and are proportional to line resistances.

In order to compute the least-squares (LS) estimate of the resistances, the system should be over-constrained, that is, the number of measurement periods *m* should be greater than the number of smart meters *n*, and the matrix *H* should have full rank (*n*+*1*). This requirement is usually easy to satisfy.

Once the parameter vector $\hat{R}$ has been obtained, we can use it to compute the non-technical losses $\hat{O}_k$ for any future period *k* as

$$\hat{O}_k = L_k - \sum_{i=1}^{n} \frac{I_i(t_k)^3 - I_i(t_{k-1})^3}{3s} \hat{R}_i - l_0$$

3 Experimental Set-up

In order to verify the proposed algorithm, we conducted an experimental study in simulation, where a detailed simulator was used to calculate the state of a typical branch of a distribution network under typical loads and fairly frequently (every 10 seconds), and the resulting losses and power consumption measurements were accumulated over the much longer intervals (30 minutes) typical for the current generation of smart meters and automatic metering infrastructure. This procedure simulates the measurement process of a set of typical smart meters, and the aggregated data computed in this way was provided to the power theft analysis algorithm described above.

The test branch of the distribution system contained one user group of 30 users, such that 10 users were attached to each of the three phases. Without loss of generality, all consumption was assumed to be single-phase, and the analysis was performed on one phase only (phase A). One of the 10 users attached to phase A was assumed to be stealing power, resulting in power theft on the order of 10% of total consumption.

The time span of captured user data was 6 days (144 hours). Power theft occurred only during the last 2 days (48 hours). The first 2 days (48 hours) of data was used for estimating a predictive model for technical losses, as described above. Then, the last 4 days (96 hours) were used to verify the accuracy of prediction of technical (and, respectively, non-technical) losses. Of these 96 hours, the first half (48 hours) had no theft, and the last half (48 hours) had theft.

In order to compute the state of the network branch (represented by all voltages, currents, phase angles, and active and reactive power consumed at each node), power flow calculation was executed every 10 seconds for the entire period of 144 hours. There were a total of 51,840 time periods for which power flow was executed.

The loading conditions for the network were specified by means of a time-varying demand profile for each user. Since actual demand profiles recorded by actual smart meters were not available, we generated them from a reasonable statistical model that assumed that the demand profile for each user had two components: a seasonal component that was the same for every user, and a random component that was different for every user. This is a reasonable model under the assumption that all users are of the same type (that is, all are commercial or all are residential), and their consumption patterns are similar, because they are driven largely by the same external factors (for example, by the air temperature in the same neighborhood that determines the demand for air conditioning services).

The seasonal component represents the average demand profile over an entire week. For our experiments, we used the actual total demand profile for the entire United Kingdom for six consecutive days in June 2012 [4]. Fig. 3 shows this seasonal profile. It is very smooth, because it is the sum of the demands of all consumers in an entire country (the UK).

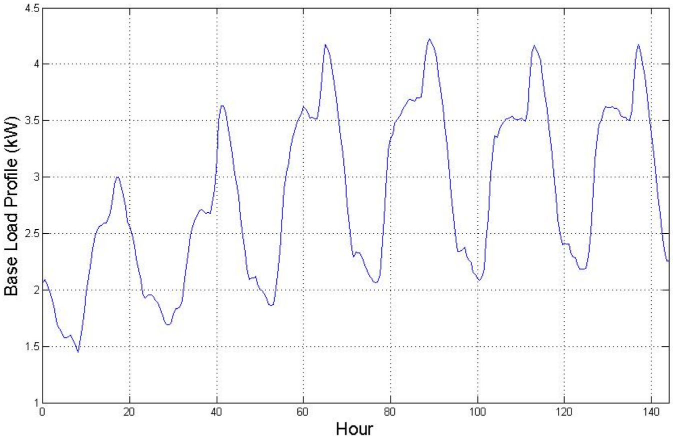


Fig. 3. Base load profile (P_{base})

The load profile for an individual user in our simulation was generated by adding a random component coming from the autoregressive (AR) process given by equation (7) to the seasonal base profile. Fig. 4 shows an example of the load profile for one user. In contrast to the seasonal component, individual load profiles are much noisier.

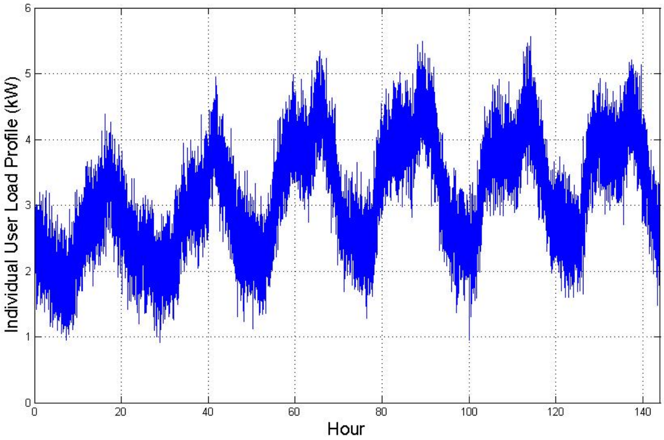


Fig. 4. Individual user load profile

$$P_i(t_k) = P_{base}(t_k) + 0.8 \cdot P_i(t_{k-1}) + 0.2 \cdot \mathcal{N}(0,1), \tag{7}$$

where:

$P_i(t)$	The load of user i at time t
$P_{base}(t)$	The base load at time t
$\mathcal{N}(0{,}1)$	Normal distribution with $\mu = 0$ and $\sigma^2 = 1$.

Here, 0.8 is the autoregressive coefficient of the AR process, and 0.2 is the standard deviation of the white noise that is driving the process.

By using this stochastic process for user demand, we are ensuring that the users attached to the same transformer have similar, but not identical demands.

4 Experimental Results

4.1 Resistance Estimates

The predictive model for technical losses was estimated from the data for the first 48 hours (96 data points). The resulting estimates for the branch resistances are shown in Fig. 5. The agreement is reasonably good, and discrepancies are due to the relatively slow sampling rate of smart meters, the quadratic nature of losses, and the necessity to approximate the profile of the current during the sampling period (in our case, by a piece-wise linear curve).

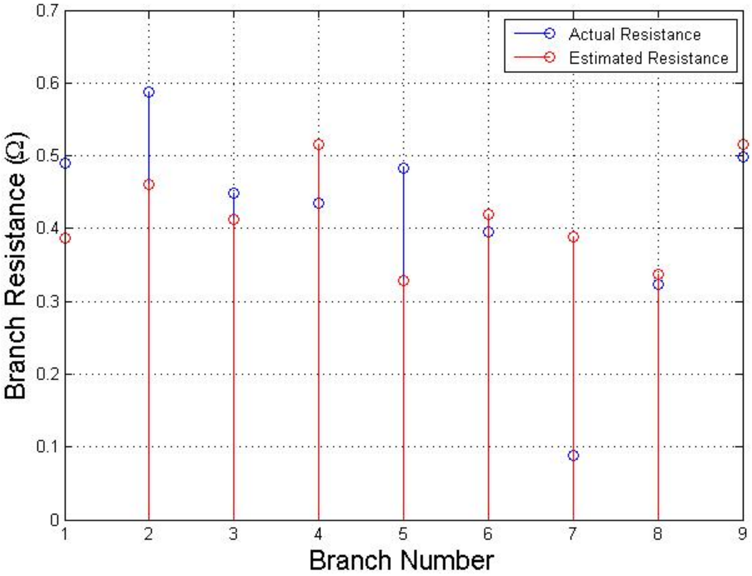


Fig. 5. Estimated and actual values for the nine branch resistances

4.2 Non-technical Loss Estimates

For every 30-minute measurement period during all three intervals of 2 days (48 hours, or 96 data points) each, we calculated the expected technical losses and subtracted them from the measured total losses to arrive at an estimate for the NTL during that period. Fig. 6 shows histograms of the NTL estimates for the three periods. The first histogram shows in red the NTL estimates from the first 48 hours (no theft). Since this dataset was used to estimate the line resistances, the shown values are in fact equal to the residuals from the least squares (LS) estimation. The implicit assumption behind the LS computation is that the residuals come from a normal (Gaussian) distribution with mean zero, and the histogram confirms that. This histogram also allows us to compute the expected variation of the NTL estimates under no-theft conditions, which we can use for determining confidence intervals and detection thresholds.

The second histogram shows in blue the NTL estimates from the second 48 hours (still no theft, but the line resistances used in producing these estimates were the ones obtained from the first data set). This histogram also shows the variation of the NTL estimates that can be expected normally, without theft, and is in agreement with the first histogram.

The third histogram shows in green the NTL estimates from the last 48 hours, when there is power theft. Visibly, the NTL values are larger than in the second case, when there was no theft. The two histograms (blue and green) do not overlap, so it is possible to completely separate the two cases, resulting in 100% accuracy of detection for this level of theft (10%). For lower level of theft, though, the two histograms might overlap, in which case an optimal separation threshold must be determined.

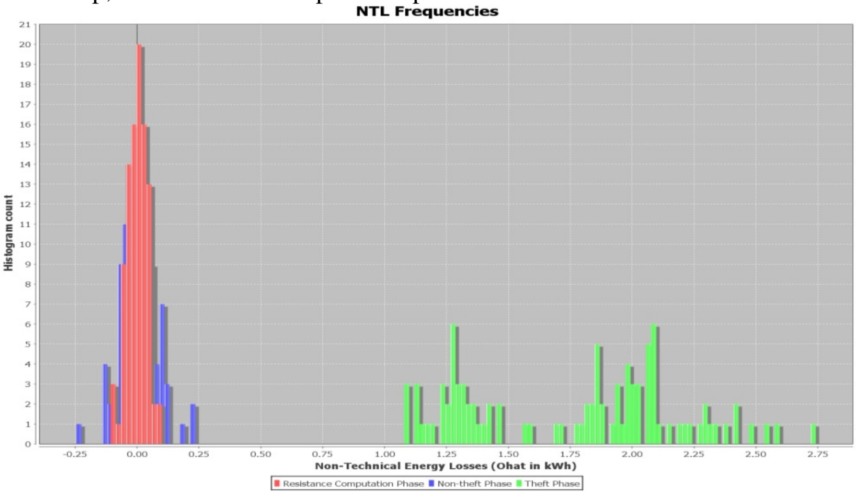


Fig. 6. Histograms for NTL estimates during three testing intervals (phases). In red, NTL estimates are shown for the interval from which the predictive model for technical losses was constructed. In blue, NTL estimates are shown for the second interval during which no theft was present. In green, NTL estimates are shown for the last interval during which theft did occur.

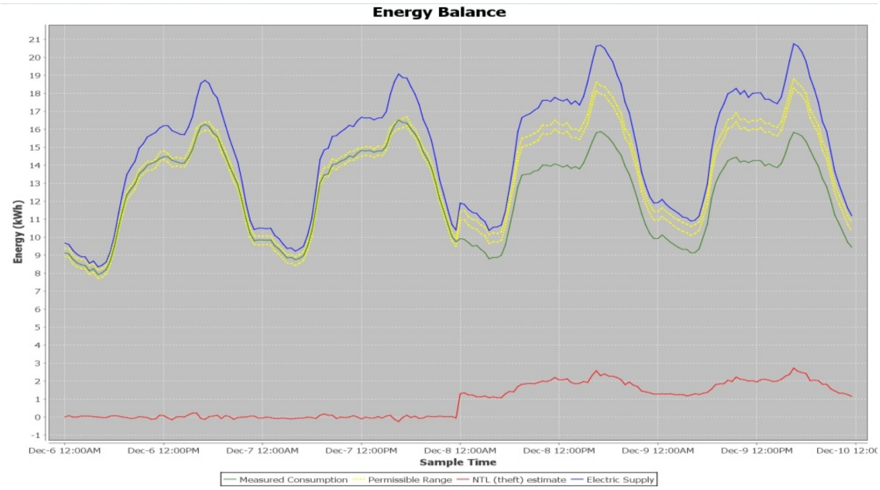


Fig. 7. A 95% confidence interval for expected total consumption (yellow lines) is computed from the measured total supply by the branch DT (blue line) and the predictive model for technical losses. When the measured total consumption (green line) goes outside of the confidence interval, theft can be suspected. The estimated amount of theft is also shown (red line).

The estimation method applied only to the last two intervals (two days of no theft followed by two days of theft) is shown in Fig. 7, as would be seen by power utility employees during actual operation. The blue line shows the total amount of energy supplied to the group of users by the branch DT, and the green line shows the sum of the reported consumption amounts for all smart meters in the group. The two yellow lines represent a 95% confidence interval for the expected total consumption derived from the technical loss model, if no theft was happening. So, when the measured consumption (green line) is outside of the confidence interval (yellow lines), power theft can be identified. The estimated amount of theft is also shown by the red curve.

5 Conclusion

We have described a method for power theft detection based on a predictive model for technical losses in distribution networks equipped with smart meters. The predictive model is constructed entirely from data collected by smart meters. Since there are several significant sources of error and noise in the measurement process, such as infrequent measurements and unknown topology of the distribution circuit, our method relies on a statistical estimation procedure to fit a good model to the data. Experimental results in simulation showed that the resulting predictive model still allows for excellent separation between cases of theft and no theft, for power theft amounts on the order of 10% of power consumption (1 illegal unmetered user out of 10). In future work, we will further investigate the accuracy of the method for smaller amounts of theft, as well as expand the model to include other external variable factors, such as environmental temperature, rain, etc. We will also adapt it to the much more difficult case when not all users in a distribution network branch are equipped with smart meters, and some of them use the traditional kind of meters that provide power consumption readings aggregated over much longer periods (one month or more). We also plan to address other types of power theft, for example theft after the meter by a third party [3,5], again using a data analytical approach.

References

1. Depuru, S., Wang, L., Devabhaktuni, V.: Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft. Energy Policy 39(2), 1007–1015 (2011)
2. Nagi, J., Yap, K.S., Nagi, F., Tiong, S.K., Koh, S.P., Ahmed, S.K.: NTL detection of electricity theft and abnormalities for large power consumers in TNB Malaysia. In: Proceedings of 2010 IEEE Student Conference on Research and Development (SCOReD 2010), Putrajaya, Malaysia, December 13-14 (2010)
3. ECI Telecom Ltd., Fighting Electricity Theft with Advanced Metering Infrastructure (March 2011) http://www.ecitele.com
4. National Grid UK,
   http://www.nationalgrid.com/uk/Electricity/Data/Demand+Data/
5. Nagi, J., Mohammad, A., Yap, K., Tiong, S., Ahmed, S.: Non-technical loss analysis for detection of electricity theft using support vector machines. In: Proc. 2nd IEEE Int. Power and Energy Conf., pp. 907–912 (2008)
6. McAvinew, T., Mulley, R.: Control System Documentation, ISA, p. 165 (2004)

Discovering Frequent Itemsets on Uncertain Data: A Systematic Review

Juliano Varella de Carvalho and Duncan Dubugras Ruiz

Computing Science Graduate Program – Faculty of Informatics
Pontifical Catholic University of RS - PUCRS
Porto Alegre, RS – Brazil
juliano.carvalho@acad.pucrs.br, duncan.ruiz@pucrs.br

Abstract. In this paper, we describe the development of a systematic review about the topic "Discovering Frequent Itemsets on Uncertain Data". To the best of our knowledge, this work seems to be the first systematic review addressing the topic. We show the whole process executed and its findings. Initially we define a rigorous protocol to lead the process. In the first phase, we create a systematic mapping of the area. In addition, from the complete reading of each article, a panorama of this area is presented. We reveal the search engines that most publicize this topic and which publishing types, authors and research institutions are involved in these papers. Moreover we identify the algorithms and the classes of these algorithms most compared over the years, how are made these comparisons, as well as their availabilities. Therefore this systematic review becomes a rich material for understanding this knowledge area.

Keywords: Systematic Review, Systematic Mapping, Uncertain Data, Frequent Itemsets, Frequent Patterns, Probabilistic Databases.

1 Introduction

One of the well-known and widely used descriptive tasks in the Data Mining area is the "Extraction of Association Rules" [3], or as called by Han et al [16], "Mining of Frequent Patterns". This task consists basically in sweeping a large amount of previously processed data to discover multidimensional correlations on them.

According to Agrawal [3], discovering frequent patterns is a process of two phases. The first one is responsible for finding all the frequent itemsets (those that satisfy a pre-define threshold). The second aims to generate the association rules among the items, into the itemsets discovered in the first phase. Han et al [16] say that the first step is more computationally expensive than the second.

Nowadays, extraction of association rules is also needed on contexts with likelihood in the data [2]. This is due to several techniques of data gathering. Besides, current applications also have increased the data with uncertainty, caused by imprecise measurements, outdated sources or sampling errors [10].

In [1], Aggarwal et al affirm and prove that classical algorithms as Apriori [3], AprioriTid [4] and FP-Growth [15] are not adapted to keep the probabilistic characteristics

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 390–404, 2013.

of the data. Thus, new methods have been developed to capture and work under these circumstances. Aggarwal et al [2] claim that studies about this theme have extended known algorithms to deal with uncertain data, from the pruning data and the creating of different computational structures to store the probabilities.

In order to contribute to the understanding of the subject area, this article discusses the results consolidated from a systematic review about the topic "Discovering Frequent Itemsets on Uncertain Data". Using a rigorous method, based on the ideas and protocol suggested by Kitchenham [21] and Biochini et al [8], this research discusses perspectives and directions found in the selected works.

This article has five sections. After this introduction, the second section explains the systematic review, its concepts and the protocol used. In the third section, it is presented the Systematic Mapping produced by applying the protocol indicated. The fourth section examines and explains some findings of the systematic review and the last section presents our achievements and limitations.

2 Systematic Review

A systematic review is a kind of secondary study used as initial step of a research. It maps the knowledge about a topic and discovers initiatives already undertaken. Thereby, it is possible to look for techniques, ideas, algorithms and strategies that exploit the studied topic. Consequently, researchers receive more support to guide their scientific investigation processes, avoiding efforts in wrong directions and, thus, helping them to plan new researches.

In [21], a systematic review is characterized as a long investigation applied on a large set of bibliographies, of a particular theme, with a previous and organized planning. This method of research follows a methodological sequence with clearly defined steps according to a protocol. The creation of this protocol occurs in the beginning of the process, by focused and well-structured questions.

2.1 The Protocol

The protocol developed in this systematic review has many items, and due to the lack of space, we summarized our observations, considering the most important items.

The Research Question

According to [21] the main activity during the construction of the protocol is to formulate the research questions. This work has three questions and they need to be answered at the end of the systematic review:

(i) Which are the algorithms, algorithm classes and frameworks most studied in the context of "Discovering Frequent Itemsets on Uncertain Data"?

(ii) How comparisons between algorithms and frameworks on uncertain datasets were made?

(iii) Can the algorithms and datasets found be used to develop comparative studies with new approaches?

Keywords

The keywords guide the creation of the search string that will be used on the web search engines. The keywords were organized in four groups and they were chosen from authors' background, bibliographic searches ad-hoc and discussions with other expert researchers in the areas of Systematic Review and Data Mining.

The keywords for each group are defined as following: Essential Terms: Itemset, Item, Pattern. Data Mining Techniques: Association Analysis, Frequent Itemsets Mining, Frequent Itemsets, Frequent Items, Frequent Patterns. Algorithm Classes/Algorithms: Generate and Test Algorithms, Hyper-Structure Algorithms, Pattern Growth Algorithms, Apriori, UApriori, FP-Growth, UFP-Growth, H-Mine, UF-Growth, UH-Mine, UEclat. Data Sources: Uncertain Data, Probabilistic Data, Uncertain Database, Probabilistic Database.

Sources Selection

Sources Identification

This section selects the sources where the primary studies will be searched [8]. The protocol subdivides this moment in some items. Regarding to local search, we selected the search engines IEEExplore, ACM Digital Library, Scopus and Elsevier InterScience. These virtual libraries were chosen because they are important references of research in Computer Science area. Based on keywords defined previously, the search string below was generated to be submitted in the elected search engines:

("Itemset" OR "Item" OR "Pattern") AND ("Frequent" OR “Analysis Association”) AND (“Generate and Test” OR “Hyper-Structure” OR “Pattern Growth” OR “Apriori” OR “UApriori” OR “FP-Growth” OR “UFP-Growth” OR “H-Mine” OR "UF-Growth" OR "UH-Mine" OR "UEclat") AND ("Uncertain" OR “Uncertain Data” OR “Probabilistic Data” OR “Uncertain Database” OR “Probabilistic Database")

The algorithms listed above were added to search string in order of reduce the scope of the research. Moreover, most papers of this area perform comparisons among some of these algorithms. However, if at least one algorithm name cited does not appear in the title or abstract of a paper, it will not be selected.

Control Article

The paper of Aggarwal [1] is referenced by various articles that work with uncertain data and the discovery of frequent patterns. Therefore, the results of the searches must show this article. The search string, which has been shown previously, recovered the control article in the search engines of the ACM Digital Library and Scopus. However, the Elsevier InterScience and IEEExplore engines did not show this paper. After some investigation we figured out that none of these two databases record that paper.

Inclusion and Exclusion Criteria Definition

After elected the sources, it is necessary to describe the criteria for studies selection and evaluation. Typically search engines recover a lot of articles. As a consequence, it is important to define the choice criteria of the returned studies. In this systematic review it was observed four criteria:

(a) the article must approach at least one algorithm or algorithm class of those listed by the keywords;
(b) the algorithms/frameworks must run on uncertain data;
(c) the papers must work with *frequent itemset patterns on traditional data model*;
(d) the publish year of article must be equal or greater than 2005.

The criterion (c) was used because discovering itemset patterns on complex structures is a wide research area and it would increase too much the amount of articles returned. The criterion (d) was added because previous ad-hoc searches did not list articles with date less than 2005.

Procedures for Studies Selection

In the beginning, the search string was submitted to the four search engines selected. All articles displayed were collected and cataloged. The selection of the studies was divided in two phases.

In the first phase, the title and abstract of all articles retrieved were read. Each article was characterized with the following items: title, authors, abstract, year, publication type, publishing vehicle name, and search engine. For each article, it was also answered the criteria of inclusion and exclusion (a), (b), (d), defined in previous subsection. The papers selected to next phase answered yes for the criterion (d) and answered yes for any of the criteria (a) or (b).

The second phase refines the systematic review. At this point, all articles that satisfied first-phase criteria were completely read. The criterion (c) was used after the complete reading of articles. A list of issues was created to allow a better comprehension of each paper and raised many important discussions, detailed in section 4.

Planning Evaluation

This step of the systematic review is essential to ensure its consistency and to capture perceptions about its feasibility. With this goal, the protocol was presented and discussed with the supervisor of a research group in business intelligence. The protocol was also showed to a Software Engineering researcher, expert in conducting systematic reviews. Afterwards, the protocol defined was discussed with two PhD students, whose do research on data mining field.

From these discussions, new ideas were raised and the absence of some items in the protocol was identified. The search string was rewritten until the one listed in the subsection "Sources Identification". Therefore, the systematic review became more consistent and precise.

Execution of the search string

The string was executed in two different ways on the four engines: (1) singular words only (2) plural words. Table 1 illustrates the number of articles found for each search engine. We can see that some engines were susceptible to variation between singular and plural.

Table 1. Number of articles extracted from search engines

	Scopus	IEEExplore	ACM	Elsevier
singular	41	2	60	103
plural	42	2	62	103

3 Systematic Mapping

In the first phase of the systematic review all titles and abstracts were read and only those articles that respect the criteria (a), (b), and (d), detailed in the subsection "Procedures for Studies Selection", were selected. From the 209 articles, just 36 (17.22%) [1, 5, 6, 7, 9, 11, 12, 13, 14, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 44, 45, 47, 48, 49] obeyed these criteria. Based on this, it was executed the mapping about discovering of frequent patterns on uncertain data.

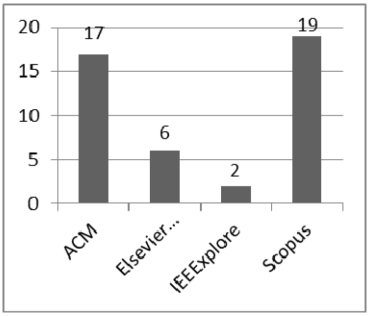


Fig. 1. Articles retrieved by search engine

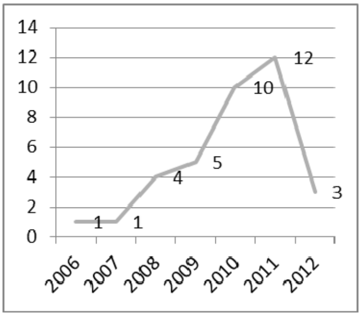


Fig. 2. Number of articles by year

3.1 Search Engine

Fig. 1 shows a chart distribution of the 36 articles, segmenting them by search engine. The Scopus database presented the greatest amount of papers, 43.18%, followed by the ACM database that returned 38.63%. These two virtual libraries have concentrated 81.81% of the 36 articles selected. It was noted that 6 articles were present in both engines: ACM and Scopus. Besides, 2 articles were listed for Scopus and IEEExplore.

3.2 Year

It was also evaluated the evolution of the area in the past seven years. Analyzing the Fig. 2, it is possible to observe the growing interest in the area. Since 2008 many articles have been published. In 2010 and 2011, this growth was very evident.

3.3 Publishing Type

The articles of the area were accepted in 25 conferences, 10 journals and one workshop. The theme had better acceptance in conferences and from the 25 articles found in conferences, 17 (68%) of them are concentrated in 5 conferences: ACM SAC, PAKDD, SIGKDD, IDEAS and CIKM, as stated in the Table 2. The remaining articles are distributed in 8 distinct conferences.

Table 2. Amount of articles by conference

Conference	ACM SAC	PAKDD	SIGKDD	IDEAS	CIKM
Articles	4	4	4	3	2

The 10 articles found in journals were detected in 9 different vehicles of publishing. Just the journal "Expert Systems with Applications" recovered 2 articles about the theme. The only article published in workshop was written to the workshop on knowledge discovery of uncertain data, held in Paris, 2009.

3.4 Authors

This mapping also permits to discover which authors publish more. 69 dissimilar authors were found. Only 12 authors have more than one publication about the theme, showing the great heterogeneity in the authorship of the articles. Indeed, two authors stood out, "Leung, Carson Kain-Sang" and "Brajczuk, Dale A.", both from University of Manitoba, with 8 and 5 articles, respectively. The authors "Gao, Hong", "Li, Jianzhong", "Zou, Zhaonian" and "Muzammal, Muhammad" also appeared with 3 articles each one.

In order to understand the relation between researchers, a graph was generated exhibiting the Social Network Analysis (SNA) [17], composed by authors (graph not presented due to space limitations). The statistical tool R was used to build this SNA. [43]. From the graph, we perceived some common relations among the authors. "Brajczuk, Dale A." and "Leung, Carson Kain-Sang" are the authors that publish more together, with 5 joined articles written.

4 Findings on the Area

After the complete reading of the 36 articles, it was necessary to include and eliminate some articles of the discussion. The famous article of "Chui et al" [12] was not listed in the 36 articles returned. The article did not appear in any search engine used in this systematic review, because, as stated in section Sources Selection, the article does not cite any algorithm of the search string in its title and abstract. However, as 22 of the total (61.1%) cited it, evidencing its relevance to this area, we decided to include it in this phase. We did not find any other frequently cited paper by the 36 selected ones.

It was also observed that 5 articles [11, 18, 22, 36, 45] do not work with uncertain data, although they have been selected. Still, 12 articles were removed because they are not in conformity with the criterion (c), detailed in subsection "Inclusion and Exclusion Criteria Definition". Explaining the exclusion of these 12 papers: from their complete reading, it is possible to classify the algorithms, techniques and frameworks found on four different approaches:

- Frequent itemset patterns: in this classical approach, the datasets are composed by traditional transactions, whose items (or transaction in some cases) are associated with existential probabilities, defining a model of uncertain data. Using this uncertain data model (with some variations), twenty articles were found [1, 5, 6, 9, 12, 13, 14, 23, 24, 25, 26, 27, 29, 31, 32, 39, 40, 41, 42, 44].
- Frequent sequential pattern: Sequential Pattern Mining (SPM) is an adaptation of the classical approach. It consists basically in finding frequent sequences of specific events with a temporal component associated. Five articles [7, 19, 33, 34, 35] work in this context.
- Frequent itemset pattern on streaming data: according to Carson Leung, *et al* [28], nowadays there is a large volume of applications that generate data streaming. Examples of these generators are environmental sensors. However, because of the limitations of these sensors, data streaming can contain uncertainty. In this systematic review, two articles approach this context, [28, 30].
- Frequent structured pattern: another approach was detected in five articles [20, 37, 47, 48, 49]. They work with discovering frequent patterns on graphs with uncertain associated.

The protocol of the systematic review allows (criterion c) the inclusion of articles that work only with traditional data models, excluding those with complex structures. Thereby, the 12 articles that work with the last three approaches presented above were eliminated. Thus, "20 articles" remained in the systematic review.

Aiming to raise discussions to answer the research questions formulated, other quality criteria were created in the protocol to lead this phase of the systematic review. The criteria showed in Table 3, were analyzed for each one of the 20 articles selected.

All the 20 articles are classified as basic research because they propose algorithms and frameworks, producing new knowledge about the area. The studies are exploratory because they examine deeply the theme, and it has gained space, in the last years, in conferences and journals, according to Fig. 2. The research method used for all articles is the benchmarking. They make the evaluation of both CPU performance and memory trade-off. This method is essential to the maturity of the area, because it creates a consensus about the existing problems and which approaches are more appropriated to solve them.

Table 3. Quality criteria used in the systematic review

What type of Study is used in the paper?	Are used datasets available?	Is the dataset type real or synthetic?
Which are the algorithms classes cited in the paper?	Are the developed algorithms available?	How are the synthetic datasets generated?
Data type: with uncertainty or not?	What is the dataset size?	Does the article detail some pruning technique?
Does the article introduce a new algorithm? What is the algorithm's name?	Are there other graphical comparatives? Which ones?	Is there charts comparing support and dataset size variations?
How the data uncertainty is represented? Which structure type is used to model this representation?	Does the paper make comparisons among algorithms? Which ones?	Is there charts comparing CPU and memory costs?

4.1 Representation of the Uncertain Data

From the 20 studies, 17 (85%) work on datasets composed of transactions, which items are associated with existential probabilities. Table 4, extracted and adapted from [14], shows an example of this database type. It consists of two transactions $T = \{T_1, T_2\}$ and each transaction $t_j \in T$, is constituted of four items $i_k \in I$, where $I = \{i_1, i_2, i_3, i_4\}$.

Table 4. Data Model with uncertain associated

	i_1 (Insomnia)	i_2 (Depression)	i_3 (Hypertension)	i_4 (Obesity)
T_1	0.5	0.2	0.25	0.4
T_2	0.9	1.0	0.2	0.5

The existential probability of an item i_k is expressed by $p(i_k, T_j) \in (0,1]$. The existential probability of the item i_1 in T_1 is equal to 0.5, represented by $T_{1,1} = <i_1, 0.5>$. Semantically, this means that $p(i_1, T_1)$ has 50% chance of suffering Insomnia.

There are 3 exceptions that represent uncertain data of different ways. In [44] the probabilities are associated to each tuple (transaction). In this case, the likelihood of a transaction t_j is represented by $P(t_j) \in (0,1]$.

The article [29] organizes data in a different way, in a vertical format. Each tuple is represented by an item i_k, with a collection of related transactions, called *tidlist*, and its associated probabilities. Consider the *tidlist* of an item i_k as *tidlist* (i_k): $\{t_1{:}0.9, t_2{:}0.8, t_3{:}0.2\}$. The item i_k appears in the transaction t_1 with 90% of probability and is also present in the transactions t_2 and t_3 with 80% and 20% of certainty, respectively.

The last exception is the article [32]. It names the data as univariate uncertain data. Each item i_k, in a transaction t_j, has the likelihood represented by a basic quantitative interval. This representation can be applied in some cases. For example, a sensor of low sensibility, used to record the amount of atmospheric pollution particles on the air, at 5 am, will store a quantitative interval, instead of a precise value.

4.2 The Algorithms and Its Classes

The articles selected develop and discuss many different algorithms and frameworks. Liu [32] categorizes the approaches that work on uncertain data in three classes: Apriori-based, FP-growth-based and H-mine-based. On the other hand, Aggarwal [1] divides the algorithms in two classes: Candidate Generate-and-Test and Pattern Growth.

Grouping the two classifications it is possible to say that the algorithms belonging to Candidate Generate-and-Test class are based on the traditional algorithm Apriori [4]: they are Apriori-based. Otherwise the Pattern Growth class is composed by the algorithms FP-growth-based and H-mine-based.

To [1], these two algorithm types have as main difference the data structure used to save the data in memory. FP-growth-based algorithms adopt a tree-based structure [15] and H-mine-based algorithms work with a hyper-linked array, developed originally in [38].

Table 5. Classification of the algorithms

Class	Algorithms and frameworks
Apriori-based	P-Apriori [14], uCBA [39], U-Apriori [1, 6, 12, 13], UCP-Apriori [13], IMBP [41], PFCIM [42], p-FPS and TODIS [40], Apriori with Poisson binomial distribution [44], PFIM [5].
FP-growth-based	U2P-Miner [32], UFP-Growth [1, 6, 9], UF-Growth [23], U-FPS [24], U-FPS [26], uCFS [25], uCFS2 [27], CUFP-mine [31].
H-mine-based	UH-Mine [1], P-Hmine [6].
Eclat	U-Eclat [9], UV-Eclat [29]

Table 5 organizes the algorithms pointed out by articles in their respective classes. The majority of them were classified in the three classes cited previously. Just two algorithms work with another approach: based on the Eclat algorithm [46].

4.3 Availability of the Algorithms

Four [1[1], 40[2], 41[3], 42[4]] from 20 articles offer their source code in a repository. Clearly, this fact demonstrates the lack of habit from the scientific community to make available their ideas and approaches to be used by other researchers. Thus, it is

[1] http://dbgroup.cs.tsinghua.edu.cn/liyan/u mining.tar.gz

[2] http://www.cs.hku.hk/~lwsun/codes/kdd10/

[3] http://code.google.com/p/imbp-programe-dataset/downloads/detail?name=IMBP and datasets.rar&can=2&q=

[4] http://www.erichpeterson.com/wp-content/uploads/2009/04/Archive.zip

necessary to implement most of the algorithms to compare them. Besides of the time to do this task, locally-created solutions may have different behavior comparing to the original implementations, making difficult a fair comparison.

4.4 Dataset Types

All articles extract their analyses and discussions from tests on real and synthetic datasets. 14 articles [1, 6, 9, 14, 23, 24, 25, 27, 29, 32, 40, 41, 42, 44] work with both dataset types, 4 papers [5, 12, 13, 26] use just synthetic datasets and 2 papers [31, 39] use only real datasets.

Generation of Synthetic Datasets

From the 18 articles that use synthetic datasets, just 2 [41, 42] do not use the IBM Synthetic Data Generator, initially developed by Agrawal and Srikant [4]. Instead, they use synthetic datasets available at Frequent Itemset Mining Dataset Repository (FIMI - http://fimi.cs.helsinki.fi/data/). However, IBM Synthetic Data Generator was also used to generate the synthetic datasets in FIMI.

The articles produce the data, basically, in two steps. The first one generates the datasets without uncertainty, using just IBM Generator and the second step assigns likelihood to data. This data generator creates automatic and configurable datasets, with transactions of users, simulating a market basket data. It allows the configuration of three variables: (a) average number of items by transaction (T); (b) average size of frequent itemsets (I) and (c) amount of transactions in the dataset (D).

Thereby, several articles identify their synthetic datasets by abbreviations like T20I6D100K. In this example, each transaction has an average of 20 items, whose average number of frequent items (itemsets) is equal to 6 and the dataset is constituted by 100 thousand transactions.

The second step is responsible by introduce uncertainty for each item (or transaction) of the generated dataset. In the article of Chui [12], for example, some items are associated with high probabilities and others with low probabilities. These items use a distribution with mean and standard deviation established previously.

Feng Gao and Chengrong Wu [14] created two datasets setting T20I10D100K, besides generating probabilities using a normal distribution with mean equal to 0.2 and 0.8. The standard deviation in this case was equal to 0.01 to both datasets. The work of Liang Wang *et al* [44] also tests two synthetic datasets, one with probabilities in the items and other with likelihood in the tuples (transactions), using the configuration T10I4D100K. Its average number of items per transaction and the average size of frequent itemsets are lower than the work of Feng Gao and Chengrong Wu [14]. Lian Wang also uses different mean and standard deviation to generate probabilities: 0.5 and 0.125, respectively.

Charu C. Agrawal, *et al* [1] applies its tests on two synthetic datasets, with different configurations (T40I10D100K and T25I15D320K) too. In this article the probabilities also are generated from a normal distribution, with the mean and standard deviation varying randomly in the range [0.87, 0.99] and [1/21, 1/12], respectively.

Analyzing only the articles cited, it was visible the large diversity of configurations applied on the synthetic datasets used in the tests. This absence of homogeneity interferes in the comparison of the results between the articles.

Real Datasets

From 16 articles that work with real datasets, 15 apply their experiments on well-known repositories: FIMI and UC Irvine Machine Learning Repository (UCI - http://archive.ics.uci.edu/ml/). Only the article [32] works with other repositories. The 15 articles use the repositories in accordance with Table 6.

Table 6. Articles and repositories used

Repositories	Amount of articles	References
FIMI	9	[1, 6, 9, 14, 31, 40, 41, 42, 44]
UCI	2	[29, 39]
Both	4	[23, 24, 25, 27]

Although UCI and FIMI repositories have many datasets, the articles often use a short set of them. The datasets most frequent in the papers, remembering that in some cases one article uses more than one dataset, are *Accident*, *Chess*, *Mushroom*, *Kosarak* and *Retail*.

4.5 Comparisons between Algorithms

Table 7 shows the algorithms proposed in each article and the ones that they were compared with. It also displays the comparison type performed. This table was sorted by publishing year, in order to give a chronological view of comparisons.

The column "Compare with" shows the greater interest of the researchers to compare their solutions with the algorithms U-Apriori, UF-Growth, UH-Mine and UFP-Growth. The algorithm U-Apriori, introduced by [12], is present in 9 comparisons of different articles. The other three algorithms UF-Growth, UH-Mine and UFP-Growth are compared in 6, 3 and 3 different papers, respectively. Just three articles [12, 14, 42] do not make any comparison with other solutions.

Most articles use many charts to analyze parameters. According to column "Comparison type" the parameters analyzed are various. From 20 articles, 13 use charts to show the execution time of the algorithm versus support (CPU cost vs. support). In addition, seven articles compare the CPU cost versus dataset size. Only the article [29] does not exhibit comparative charts.

Table 7. Articles selected

Year [Ref]	New solution	Compare with	Comparison type
2007 [12]	U-Apriori, LGS-Trimming.	The both solutions created.	CPU cost vs. % items with low existential probabilities; CPU cost vs. support; CPU cost vs. iteration; Number of dataset scan vs. iteration.
2008 [13]	UCP-Apriori.	U-Apriori, Framework LGS-Trimming.	% of candidates pruned vs. fraction of dataset scanned; CPU cost vs. iteration CPU cost vs. support; CPU cost vs. % items with low existential probabilities.

Table 7. *(Continued)*

Year [Ref]	New solution	Compare with	Comparison type
2008 [23]	UF-Growth.	U-Apriori.	CPU cost vs. support; CPU cost vs. dataset size; CPU cost vs. existential probability; Reduction in tree size vs. support.
2009 [1]	U-Apriori, UFP-Growth, UH-mine.	The algorithms itself, besides UCP-Apriori, UFP-Growth.	CPU cost vs. support; Memory usage vs. support; CPU cost vs. Number of Transactions; Memory usage vs. Number of Transactions.
2009 [24]	U-FPS (with UF-Tree).	UF-Growth, FPS.	CPU cost vs. percentage of items selected; CPU cost vs. support; CPU cost vs. dataset size.
2009 [5]	PFIM.	Five approaches of the framework itself.	CPU cost vs. dataset size; CPU cost vs. dataset density; CPU cost vs. support vs. dataset density.
2009 [25]	uCFS	U-Apriori, UF-Growth, CAP, FIC, DCF.	CPU cost vs. constraint selectivity; CPU cost vs. existential probability; CPU cost vs. support; CPU cost vs. dataset size.
2010 [9]	U-Eclat and UFP-Growth.	U-Eclat, UCP-Apriori, UH-Mine.	CPU cost vs. support; Error vs. support; Precision vs. support; Recall vs. support.
2010 [39]	uCBA.	CBA.	Accuracy vs. coverThreshold
2010 [27]	uCFS$_2$.	U-Apriori, UFP-Growth, CAP, uCFS.	CPU cost vs. selectivity (% of frequent sets selected); Number of constraints checks vs. selectivity; Number of extensions vs. selectivity.
2010 [26]	U-FPS	DCF, CAP, FIC, U-Apriori, UF-Growth, U-FPS.	CPU cost vs. selectivity.
2010 [40]	p-Apriori (bottom-up), TODIS (top-down).	p-Apriori DP, p-Apriori DC, p-Apriori DP-n, p-Apriori DC-n.	CPU cost vs. dataset size; CPU cost vs. support; CPU cost vs. confidence.
2010 [44]	Two algorithms Apriori-based using binomial distribution of Poisson.	Apriori-based using dynamic programming.	CPU cost vs. support; Number of PFIs vs. support; CPU cost vs. probability; CPU cost vs. dataset size; CPU cost vs. probability distribution; Dataset scanned vs. support; CPU cost vs. top-k.
2011 [14]	P-Apriori.	The P-Apriori itself.	CPU cost vs. support; CPU cost vs. confidence; CPU cost vs. top-k; Memory usage vs. support; Memory usage vs. confidence; Memory usage vs. top-k.
2011 [29]	UV-Eclat.	U-Apriori, UF-Growth, UH-Mine.	It don't show comparative graphs.
2011 [6]	P-Hmine.	U-Apriori, P-MaxClique, UFP-Growth.	CPU cost vs. support; Memory usage vs. support.
2011 [42]	PFCIM.	The PFCIM itself.	CPU cost vs. support; CPU cost vs. frequentness probabilities.
2012 [32]	U2P-Miner.	Extended Apriori, H-miner and extended Depth-first backtracking.	CPU cost saving vs. support; CPU cost vs. dataset size; CPU cost vs. support.
2012 [41]	IMBP (based in the MBP).	MBP.	CPU cost vs. support; Recall vs. support; Number of candidates generated vs. support.
2012 [31]	CUFP-Mine.	UF-Growth.	CPU cost vs. support; Number of nodes vs. support; CPU cost vs. decimal range; Number of nodes vs. number k of decimals; Number of nodes vs. number k of decimals.

5 Discussion and Conclusion

The systematic review answered the questions (i), (ii) and (iii), listed in “The Research Question” section.

(i) Apriori-based and Pattern-Growth are the most studied classes in the context of uncertainty data. Regarding to algorithms, those that received more attention from scientific community are the U-Apriori and UFP-Growth.

(ii) Most articles concentrate their analysis focusing on the performance of the proposed algorithms, often making comparisons using memory consumption and CPU cost.

(iii) Real datasets used can be comfortably retrieved from FIMI and UCI repositories. However, the synthetics datasets, although generated by IBM Synthetic Data Generator in 88.88% of the papers, have great diversity in the configuration parameters, making difficult performance comparisons between the algorithms.

5.1 Limitations and Future Work

This systematic review has two limitations. The first one is the small number of search engines used. Probably, the use of other engines, such as Google Scholar, Wiley and Springer, would enrich the work, but the time necessary to do other comparisons would be much longer.

Rigorously, the whole process of systematic review was carried out by only one of the authors. However, frequent interactions between the authors helped to check all tasks, solve doubts and plan next steps. In fact, this method allowed a better homogeneity of the discussions in the article, but recommendations to the application of systematic review strongly state the whole process should be performed by at least two researchers.

The most articles work with the definition of *mining frequent itemsets over uncertain data* using *expected support-based itemsets* semantic approach [1][12][13]. Another approach implemented in the articles is based in *probabilistic frequent itemsets* [5][44]. Studies that compare both definitions were not found and suggest future works. Even as the absence of articles that shows the results over uniform datasets also suggest future investigations, because they will allow comparisons and conclusions most realistic.

References

1. Aggarwal, C., et al.: Frequent Pattern Mining With Uncertain Data. In: 15th ACM SIGKDD, Paris (2009)
2. Aggarwal, C.: Managing and Mining Uncertain Data. Springer, USA (2009)
3. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: ACM SIGKDD (1993)
4. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: VLDB (1994)

5. Bernecker, T., et al.: Probabilistic frequent itemset mining in uncertain databases. In: 15th ACM SIGKDD (2009)
6. Bhadoria, R.S., Kumar, R., Dixit, M.: Analysis on probabilistic and binary datasets through frequent itemset mining. In: WICT 2011 (2011)
7. Bhatt, C.: Kankanhalli M. Probabilistic temporal multimedia data mining. ACM Transactions on Intelligent Systems and Technology (2011)
8. Biolchini, J., et al.: Systematic Review in Software Engineering. COPPE/UFRJ Technical Report RT-ES 679/05, Rio de Janeiro (May 2005)
9. Calders, T., Garboni, C., Goethals, B.: Efficient pattern mining of uncertain data with sampling. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS, vol. 6118, pp. 480–487. Springer, Heidelberg (2010)
10. Chau, M., Cheng, R., Kao, B.: Uncertain Data Mining: A New Research Direction. In: Workshop on the Sciences of the Artificial, Hualien, Taiwan, December 7-8 (2005)
11. Chen, Y., Weng, C.: Mining association rules from imprecise ordinal data. Fuzzy Sets and Systems (2008)
12. Chui, C.-K., Kao, B., Hung, E.: Mining Frequent Itemsets from Uncertain Data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
13. Chui, C.-K., Kao, B.: A decremental approach for mining frequent itemsets from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 64–75. Springer, Heidelberg (2008)
14. Gao, F., Wu, C.: Mining frequent itemset from uncertain data. In: ICECE 2011 (2011)
15. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. SIGMOD (2000)
16. Han, J., Kamber, M., Pei, J.: Data Mining Concepts and Tecniques. Morgan Kaufmann (2011)
17. Hanneman, R.A., Riddle, M.: Introduction to social network methods. Univ Calif. Riverside (2005), http://faculty.ucr.edu/~hanneman/
18. Herawan, T., Deris, M.: A soft set approach for association rules mining. Knowledge-Based Systems (2011)
19. Kadri, O., Ezeife, C.I.: Mining uncertain web log sequences with access history probabilities. In: ACM SAC (2011)
20. Khan, A., Yan, X., Wu, K.L.: Towards proximity pattern mining in large graphs. In: ACM SIGMOD (2010)
21. Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Keele Univ. EBSE Tech. Rep. EBSE-2007-01, UK (2007)
22. Lee, Y., Hong, T., Wang, T.: Multi-level fuzzy mining with multiple minimum supports. Expert Systems with Applications (2008)
23. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
24. Leung, C., Brajcsuk, D.A.: Efficient algorithms for mining constrained frequent patterns from uncertain data. In: 1st ACM SIGKDD Workshop on Knowledge Discovery from Uncertain Data (2009)
25. Leung, C., Brajcsuk, D.A.: Mining uncertain data for constrained frequent sets. In: IDEAS (2009)
26. Leung, C., Hao, B., Brajcsuk, D.A.: Mining uncertain data for frequent itemsets that satisfy aggregate constraints. In: ACM SAC (2010)

27. Leung, C., Brajcsuk, D.A.: uCFS2: an enhanced system that mines uncertain data for constrained frequent sets. In: IDEAS (2010)
28. Leung, C., Jiang, F., Hayduk, Y.: A landmark-model based system for mining frequent patterns from uncertain data streams. In: 15th IDEAS (2011)
29. Leung, C., Sun, L.: Equivalence class transformation based mining of frequent itemsets from uncertain data. In: ACM SAC (2011)
30. Leung, C., Jiang, F.: Frequent itemset mining of uncertain data streams using the damped window model. In: ACM SAC (2011)
31. Lin, C., Hong, T.: A new mining approach for uncertain databases using CUFP trees. Expert Systems with Applications (2012)
32. Liu, Y.: Mining frequent patterns from univariate uncertain data. Data and Knowledge Engineering (2012)
33. Muzammal, M., Raman, R.: On probabilistic models for uncertain sequential pattern mining. In: Cao, L., Feng, Y., Zhong, J. (eds.) ADMA 2010, Part I. LNCS, vol. 6440, pp. 60–72. Springer, Heidelberg (2010)
34. Muzammal, M., Raman, R.: Mining sequential patterns from probabilistic databases. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 210–221. Springer, Heidelberg (2011)
35. Muzammal, M.: Mining sequential patterns from probabilistic databases by pattern-growth. In: 28th British National Conference on Databases (2011)
36. Özyer, T., Alhajj, R., Barker, K.: Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening. Network and Computer Applications (2007)
37. Papapetrou, O., Ioannou, E., Skoutas, D.: Efficient discovery of frequent subgraph patterns in uncertain graph databases. In: 14th EDBT (2011)
38. Pei, J., et al.: H-mine: hyper-structure mining of frequent patterns in large databases. In: ICDM (2001)
39. Qin, X., Zhang, Y., Li, X., Wang, Y.: Associative classifier for uncertain data. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 692–703. Springer, Heidelberg (2010)
40. Sun, L., et al.: Mining uncertain data with probabilistic guarantees. In: ACM SIGKDD (2010)
41. Sun, X., Lim, L., Wang, S.: An approximation algorithm of mining frequent itemsets from uncertain dataset. Intl. Journal of Advancements in Computing Technology (2012)
42. Tang, P., Peterson, E.A.: Mining probabilistic frequent closed itemsets in uncertain databases. In: 49th Annual Southeast Regional Conference (2011)
43. The R Project for Statistical Computing, `http://www.r-project.org/` (accessed on October 8, 2012)
44. Wang, L., et al.: Accelerating probabilistic frequent itemset mining: a model-based approach. In: 19th ACM CIKM (2010)
45. Yin, P., Li, S.: Content-based image retrieval using association rule mining with soft relevance feedback. Visual Communication and Image Representation (2006)
46. Zaki, M., et al.: New algorithms for fast discovery of association rules. In: ACM SIGKDD (1997)
47. Zou, Z., et al.: Frequent subgraph pattern mining on uncertain graph data. In: CIKM (2009)
48. Zou, Z., Gao, H., Li, J.: Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In: ACM SIGKDD (2010)
49. Zou, Z., et al.: Mining frequent subgraph patterns from uncertain graph data. IEEE Transactions on Knowledge and Data Engineering (2010)

Mining Groups of Common Interest: Discovering Topical Communities with Network Flows

Liyun Li[1] and Nasir Memon[2]

[1] Linkedin Corporation,
2029 Stierling Ct, Mountain View, 94043
liyli@linkedin.com

[2] Polytechnic Institute of New York University
6 Metrotech Center, Brooklyn, NY, 11201
memon@nyu.edu

Abstract. This paper tackles the problem of detecting topical communities from within an organization by mining readily available network access pattern information. A Bayesian generative process is used to model the behavior of user's network access pattern and thereby her consumption of online content. The idea is that users within same topical interest group tend to share similar online access patterns. By leveraging this pattern, along with side information of domain-names and keywords within the accessed websites, one is able to model these observations under the framework of a mixed membership statistical model. Hence the access patterns of users-to-websites, as measured at the edge of an organization's network boundary, can be decomposed into constituent topical communities without any human effort in selecting specific features. Experimental results on real-world network flow trace demonstrate that the proposed method can effectively detect topically meaningful community structures. Besides better detection accuracy of communities compared with other community detection methods, the proposed method can detect interesting but non-evident hidden communities which cannot readily be detected by other known methods.

Keywords: Topical Community Detection, Generative Model, LDA, Network Flow.

1 Introduction

With the emergence of social media, the role of web users has shifted from passive navigators to active content creators, moderators and consumers. Given the rich source of data and tools provided by Web 2.0, users have the ability to actively create and share content while collaborating and communicating online. Consequently, there are now many rich sources of heterogeneous data created by millions of users participating in numerous online social interactions, not only accessing but creating, sharing, and annotating content they are interested in. Furthermore, the millions of users who share, communicate, and interact with each other online come from diverse demographical, geographical, and even

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 405–420, 2013.

topical interest groups which may be implicitly reflected in the content that they access and other online behavior patterns they demonstrate.

Analyzing user interaction data and exploiting both evident and non-evident (often hidden) communities that are created is important[19] since the communities a user belongs to can critically influence the content he/she consumes. And the aggregate community behavior will probably drive the trends and events on a website or even the entire Web [13]. Moreover, the discovery of either explicit or implicit user communities on the Web can be beneficial as many web or enterprise based applications could specifically tailor and leverage such community information[13]. Applications such as recommendation[18], personalization[11], content distribution[12], and marketing[13], can become more efficient, targeted and focused. In addition, information such as access performance, quality of service and user satisfaction could be improved if information on users' community relationships is known[23].

The problem of decomposing networks into cohesive clusters, mostly known as community detection, has attracted increasing attention and research in recent years. While different techniques of community detection have been widely used in many scientific disciplines and in many different ways (such as search engine optimization, targeted advertising, and personalized recommendations [24]), most research has focused on detecting either social communities or common groups using the interaction graph of users [23]. Hence these approaches, while focusing on the user-to-user interaction graph, remain limited to the aspects of people's behavior on specific websites (such as a certain social network website or a shopping website) and therefore do not fully leverage user's day-to-day network access behavior across multiple websites and their broader semantical topic interests. For example, decomposing all the users accessing a shopping website cannot fully exhibit the actual users' professional topical interest, which might be reflected in their activities while visiting other websites.

In this paper we are interested in the topic-based community detection problem from users' web access behavior across multiple websites. This can in general be difficult as such information can be hard to obtain. However, we observe that for users within an institution, their web access patterns can be readily gleaned from simple and readily available network flow data. By looking into the aggregated network access patterns observable from network flow data, we aim to discover the hidden topical-driven communities within the organization. Here "the topical communities" we want to discover could be groups of users either from similar demographical or ethical groups, or formed by sharing similar interests and visiting external websites of similar topical content, as per the extracted text information from the external websites visited.

By leveraging the network access patterns as well as scarce content information, we show that users within an organization could be categorized into both evident demographical and non-evident topical clusters. Understanding the topical interests within an organization could be useful in many ways. For an enterprise, if one can understand which types of professional topics and industry trends are ongoing, he/she can then predict interest trends, grasp the most popular industry innovations and take opportunities to make the business more successful. As shown in[11], companies are very interested in leveraging the on-going topical communities in context of productivity and security. Understanding the hidden topics within a company is critical and beneficial in many aspects such as expertise and knowledge search, social proximity and collaboration, social recommendations and even cyber-security. For non-profit driven institutes such as colleges and universities, a better understanding of the topical communities will facilitate the school to be able to provide better educational resources as well as more specific talent discovery and matching.

In the rest of this paper, we report our experience and solutions for discovering and analyzing existing topical communities in organizations such as enterprise or institutions. Given the network access trace of users within an organization, we use Latent Dirichlet Allocation (LDA), a Bayesian topical model, to predict the topical communities that each user belongs to.

More specifically, the contributions made in this paper are as follows:

1. We tackle the problem of mining semantically meaningful topic communities from within an organization, and formulated the topical community detection problem as a probabilistic model for clustering users in a graph.
2. Using only the observed who-visited-what network flows, along with limited content-based keyword frequency information extracted from either the website itself or an external source, we present a methodology to use an LDA-based approach to dynamically cluster users into possibly overlapping communities, each of which represents a certain topical interest shared among a group of users.
3. Experimental results on a real-world network flow trace from an institute shows that our proposed method can effectively detect topically meaningful communities including both demographical communities and non-evident communities sharing similar interests in certain semantic topic, which cannot readily be detected by other known methods.

The rest of this paper is organized as follows. In Section 2 we discuss related work on community detection. Then we describe our approach in details at Section 3 and present the experimental results in Section 4. In Section 5, we conclude and discuss limitations and future work.

2 Related Work

Detecting community structures from complex networks has been studied in different contexts such as Social Network[6][7][14][16][2][3], Web Graph[19], and Biology[15]. Most of these approaches could be categorized as either modularity-driven[15], where the goal is to optimize certain objective function given the graph structure, or statistical inference based where a Bayesian Generative Model is applied [20][21]. Working under the definition of a community which should have better internal connectivities than external connectivities, the objective function to optimize is often related to the cut of a graph, and the problem under this scenario is proven to be NP-hard[26]. Therefore, different heuristics[24] are employed to solve the problem. Meanwhile, there are also approaches using Bayesian Generative Models to tackle the community detection problem[20][21]. Then the question becomes what posterior settings of members belonging to each community best explains the membership data-observed. For detailed comparison of the community detection algorithms, we point to the survey and comparison in [24].

Studying the community detection problem from an enterprise angle for either talent discover or industry revenue/trend prediction has been proposed by Lin in [13][11][12], where the behavior of users within an enterprise is studied and modeled. Users' behaviors on different social websites as well as content information is utilized in the work[13] to bring predictive value for the enterprise. Community detection has also attracted interest from the social network area. In [17], the authors proposed an LDA approach to model user interaction behavior communities. In [9][10], researchers demonstrated information leakage could happen for certain user groups. More specifically, by adopting a similar LDA-based approach, they are able to infer demographical information based on user 'likes' on social network sites such as Facebook. In [8][1], both content and user interaction data are utilized to provide semantically improved clusterings.

3 Topic Modeling on Web-Access Data from NetFlow

In this section we describe our community mining approach. We first describe how we build a graph from network flow data. Then we describe the intuition how we can apply the Latent Dirichlet Allocation(LDA) algorithm in our setting. Finally we formulate our approach and present the detailed algorithm used.

3.1 User-Web Access Graph

Network flow data, specifically the industry standard NetFlow, is the main source of information we use for community detection. When the edge router within an enterprise network is NetFlow enabled, statistics of communication sessions between two hosts (one inside the network and the other outside) are collected in the form of a NetFlow record. From each NetFlow record we only use the source and destination addresses as well as the port number and protocol associated with the flow. Given this information we first extract all the web accesses made by only considering flows which are associated with web access ports (such as port 80, 8080, 8081). For each such flow, the internal IP address represents an internal user while the external IP represents a website which the user visited. Given that domain name information is public, we query the DNS servers along with other public Domain name services (such as Google DNS) to get the real-time domain name mapping of the external website for each record and store this information.

What we now have is a bi-partite graph which has two type of nodes. On one side are the internal IP addresses (or users) and the other side is the external website domain names, and there is an edge between an internal user that accessed a particular website. We call this graph the User-Web Access Graph. We make assumption here that the mapping of User-Identify to IP is relative stable in a short period of time given prior work on the dynamics of IP addresses[25].

Annotating the User-Web Access Graph with Content Information. The bipartite nature of the User-Web Access Graph described above makes it hard to capture social interactions between users, i.e., the user-to-user social graph which has often been used for user segmentation and profiling[17]. However, the bipartite graph represents more complete and multi-faceted activities of users, as it is not restricted to specific websites. This richer source of activities provides us with users' website access patterns. In this section we show how we can supplement these access patterns with content-based semantic information. The resulting dataset reflects topic-driven user behaviors, where the patterns of how users visit different websites and what content they consume are largely dependent on such explicitly expressed or implicitly self-formed communities (which we refer to as "topical communities").

To capture semantic information for each website in the User-Web Access Graph we extract text content from these external websites by performing the following procedures. Firstly, we actually query the website against external repositories of information such as Wikipedia and mine the descriptive text from that source. By performing *tf-idf* based processing of the frequencies of the words in the extracted text, we store the frequencies of the remaining words in the text, which we address as descriptive "keywords" depicting the characteristics of the website's content. So for example, when a user visited *techcrunch.com*, keywords such as "technology" and "start-up" will be extracted.

In many cases, such Wikipedia like descriptions may not be available. We then actually establish a connection to the external website and mine the text using existing web-text mining tools. In this way, for each website, we will have keywords information describing the website. For globally popular websites such as *Amazon.com* or *facebook.com*, the mined keywords may not be descriptive enough to annotate the content preference of users. Therefore we discard these keywords from such general-purpose websites and only store the domain name related keywords. However, if a users visits websites such as *techcrunch.com* or a specific websites only for people speaking certain languages, the extracted keywords will then be informative in representing users' characteristics, either in a demographical or topical fashion.

We now have the dataset we run our algorithm consists of User-Web Access graph, along with frequencies of keywords annotating each external website.

3.2 The Intuition of Applying LDA for Clustering

Given the bipartite nature of the fully observed User-Web access graph and very limited content information extracted from the websites, we propose to apply LDA as a topic modeling tool to solve the community detection problem. In the context of our problem, the LDA model formalizes the intuition that users visit certain websites for a “topical purpose”. The idea is that, if we were to treat users and websites as nodes in a graph and place an edge between two entities if they are found to communicate, then densely connected regions (nodes and edges) of the graph are likely to belong to the same topical interest. The rationale for this is that there are likely to be many more links between members of the same topical community in the web-visiting graph, than between two different topical communities.

Imagine that each web-access record between an internal host and an external host was produced by an imaginary two-step process (see Fig 1): each internal host first picks, at random, a topical community to participate in one out of K possible communities, and then given the chosen community, an external-website that belongs to that community is picked, also at random. This two step process is repeated for each web-access network record, with the random draws made independent of other draws. Each of the two random draws are made according to specific (as yet unknown) distributions - the first, a distribution over communities, and the second, over external websites. These distributions can be treated as tunable parameters of a model. Using Bayes’ rule to answer the question: “What setting of the parameters best explains our observed web-access activities?”, gives us the distributions, from which we can infer a grouping of hosts into communities. In addition to clustering the users, external websites are also clustered into different topical communities together with the users. Then with content keywords extracted from these websites, we are able to label and explain the topical communities in our results.

3.3 Applying LDA to Mine Topical Communities

The LDA model has been applied to community discovery in social networks [17][22]. We describe it here in our problem’s setting. Consider a dataset D, consisting of $\{internalUser, ExternalWebsite\}$ pairs in a given time interval, where internal users are represented by IP address and external websites are identified as domain names. If we think of D as a graph G where vertices represent hosts, and undirected edges represent flows between pairs of hosts, then G is a bipartite graph. Each user in D can be a member of one or more topical communities. We assume there are K topical communities in all.

Detailed description of terminology is presented in Table 1. We first define two families of multinomial distributions:

θ: For each user(identified as an internal IP) i, consider a K-dimensional multinomial probability distribution θ_i, where $i = 1, \ldots, N_{int}$, where N_{int} is the number of internal users in $\mathcal{D}$. The n^{th} element of θ_i represents the extent to which internal user i belongs to the n^{th} topical community.

β: For each topical community C_j, $j = 1, \ldots, K$, consider a multinomial probability distribution $\beta_j, j = 1 \ldots K$ over all the external websites in $\mathcal{D}$. The dimensionality of each β_j is N_{ext}, the number of external websites that users have visited. The m^{th} element of β_j represents the extent to which the m^{th} external website participates in the j^{th} topic community.

Table 1. Summary of notation

Symbol	Meaning
$\mathcal{D}$	Dataset of {Internal User-IP, External Website } pairs
$\mathcal{D}_i$	Set of all flows involving internal users i
N_{int}	Number of internal users
N_{ext}	Number of external websites
K	Approximate number of topical communities
θ_i	Multinomial of dimension K
β_j	Multinomial of dimension N_{ext}
$C_{i,n}$	Community of the n^{th} network access record of internal user i
$H_{i,n}$	External website involved in i^{th} internal user's n^{th} flow
α	Dirichlet hyperparameter for producing prior θ_is
η	Dirichlet hyperparameter for producing prior β_js
γ	Fraction of users that are internal

We first collect all the website accessing records in D involving internal user i into a datasetD_i. Let $|D_i|$ denote the number of accessing records in D_i. We ignore the order in which these accessing network flows occur. The behavior of each internal user i in D_i can be described via a probabilistic generative process as follows (see Fig1 for an illustration of the generative process. The numbers within parentheses in Fig1 correspond to the numbering of the 3 steps below as well as in Fig2).

1. A Dirichlet distribution (a distribution over distributions) is sampled to randomly pick a multinomial for each of the θ_i and β_j distributions. In Figure 2, α and η are parameters of the two separate Dirichlet distributions. We use $\alpha = 0.3$ and $\eta = 0.01$ based on suggestions in [5].
2. For each website access record in $\mathcal{D}_i$, first randomly pick one of the K communities, by sampling the multinomial θ_i. Let the chosen community be denoted by $C_{i,n}$.
3. Then given $C_{i,n} = k$ was picked, ($k \in \{1, \ldots, K\}$), pick an external website by sampling from β_k. Let the external website picked be denoted by $H_{i,n}$.

The last two steps above are repeated for each of the $|D_i|$ flows in D_i, and the set of three steps above is repeated for each internal user in D. Given prior distributions θ_i, $i = 1, ..., N_{int}$ and β_j, $j = 1, ..., K$, and this probabilistic generative process (i.e., how observed data D is linked to a given set of prior distributions θ_i and β_j), our objective is to infer posteriors $\theta_i, \beta_j|\mathcal{D}$. Exact computation of the posteriors on θ_i and β_j is intractable. We used collapsed Gibbs sampling as our approximate inference algorithm as it is scalable and could be paralleled on a cluster[4].

As a result, the Gibbs sampling based LDA algorithm provides us with a matrix of size $N_{ext} \times K$ in which the $(i,j)^{th}$ element is the estimate of the number of access-records that external websites i has been involved in while participating in community j. The columns of these matrices can be normalized

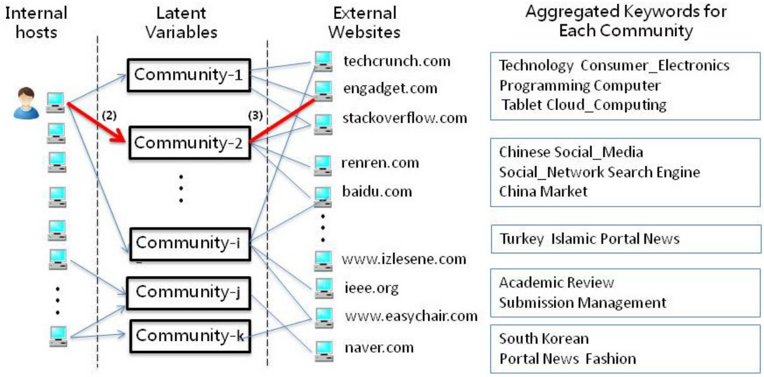

Internal hosts
Latent Variables
External Websites
Aggregated Keywords for Each Community
Community-1
Community-2
Community-i
Community-j
Community-k
(2)
(3)
techcrunch.com
engadget.com
stackoverflow.com
renren.com
baidu.com
www.izlesene.com
ieee.org
www.easychair.com
naver.com
Technology Consumer_Electronics Programming Computer Tablet Cloud_Computing
Chinese Social_Media Social_Network Search Engine China Market
Turkey Islamic Portal News
Academic Review Submission Management
South Korean Portal News Fashion

Fig. 1. The probabilistic generative process that explains the dataset of (User, Website) pairs. The numbers in parenthesis correspond to the steps in Section 3.2 and Figure 2.

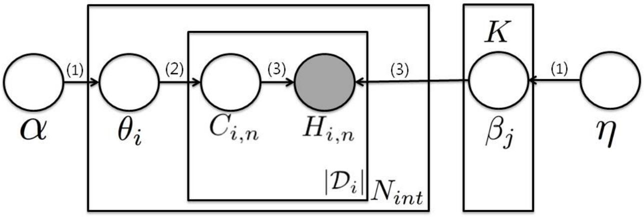


Fig. 2. The probabilistic graphical model for LDA. Only the shaded nodes are observed

to sum up to 1 to yield the posterior distributions $\beta_i, i = 1, \ldots, K$. Since internal users are connected to external websites by web-access flows, and we have assigned external hosts to communities, we can now infer a second matrix of size $N_{int} \times K$ containing the estimated number of flows for each internal user in each community. The rows of this matrix can be normalized to sum up to 1, to yield the posterior distributions $\theta_i, i = 1, \ldots, N_{int}$. By examining into these estimated posterior probabilities, we are able to put the internal users as well as external websites into possible overlapping clusters by simple thresholding on the posterior probability values.

Given the discovered communities of our algorithm, the remaining problem is how to label and interpret these communities that we have generated. We use the mined frequencies of keywords (as described in Section 3.1) associated with the websites a cluster of users have visited, to interpret our discovered communities. These keywords are filtered to remove stop words without semantic significance. By labeling each user community with a collection of keywords, the keywords and their associated frequencies constitute a description of the "topic" in this community. By annotating the discovered communities with these mined text keywords, we have decomposed the user-to-website access record into interesting descriptive clusters. Each community will have a "description", which is essentially a distribution over the keywords. These keywords somehow represent the popular topic(s) within the discovered community, and enables us to explain and interpret the topics for the community. As a simple example, if we see a community whose keywords are mostly related to a country, then we have probably discovered the demographical community from that country.

By running our algorithm periodically over a time-window (12 Hours in our setting), we are able to perform topical community discovery by clustering all the users into topical communities characterized by a unique distribution of keywords. Looking into the keywords of each community, we are able to infer semantical meaning of the topics we have generated. Such topics may include information such as demographics, ethnicity, occupations, professional interests and skills, and popular topics on the internet. More interestingly, we can not only build a knowledge base of users' topical interests, but also monitor the dynamics and developments of each topical community. In addition, new trending and unseen topics could also be exhibited by running our approach and examining the changing frequencies of keywords as well as new keywords.

4 Experimental Results

We now present experimental results showing that our approach can accurately identify topical communities. In addition, we examine and present the characteristics of several expected communities as well as some unexpected ones.

4.1 Evaluation Metric and Labeled Dataset from DHCP Log

Given our unique problem setting and viewpoint, obtaining a labeled dataset is difficult for several reasons. First, getting the complete topical interests from users is difficult due to privacy concerns. In addition, each user has his/her own unique topical interests and a reasonably large dataset has to be obtained to cover a large portion of users in order to perform accurate evaluation. Rather than manually obtaining a small labeled dataset by individual user survey, we explored the dhcp log associated with our network access traffic. Dhcp log includes information on which IP addresses have been assigned to each device. A device, which could either be a PC, laptop, tablet or even mobile phone, is identified by either its MAC address or a host name. In our case, we found that many devices actually have meaningful host names associated with them. In many cases, the host name included information of the users' actual name or information of the office location of the computer. By connecting this information with our roster database, we were able to accurately identify 262 users' identities. With this information, we constructed a labeled dataset which describes the demographical and ethnic data of these users, as shown in Table2. This dataset then was used as a benchmark for evaluating our approach for measuring performance of identification of ethnic communities. It should be noted that the experiments were done with procedures approved by the university IRB and proper procedures to protect the privacy of individuals were put in place.

4.2 Performance of the Mined Topical Communities

To evaluate the accuracy of our approach, we ran our clustering algorithm and compared with the ground truth labeled community dataset for the period of time where the corresponding dhcp log was recorded. To pick an appropriate number of communities, we adapted the notion of perplexity measure as suggested in [5]. Note that the precise number of communities is difficult to compute and remains an open problem. However, in our scenario, a good approximation which enables us to capture most active communities will suffice. In our problem, the number K was tuned to be 40.

Two metrics, precision and recall, were used to measure performance. Let C denote a labeled demographical community and $\hat{C}$ be its detected version. In order to pick up the "detected" version of C, we went through every discovered community and chose the one with most "similarity" with respect to C . More specifically, for each community, we compute its Jaccard Similarity with the labeled demographical community, and select the one with highest Jaccard Similarity as the "detected" corresponding one. Precision and recall were then computed from C and $\hat{C}$. Precision was defined as the fraction of the members in $\hat{C}$ that actually come from the underlying ground-truth community C. Recall was defined as the fraction of members in C that appear in the detected community $\hat{C}$. Formally:$precision_C(\hat{C}) = \frac{|C \cap \hat{C}|}{|\hat{C}|}$; $recall_C(\hat{C}) = \frac{|C \cap \hat{C}|}{|C|}$

Table 2. Labeled Ground Truth: Demographical Topical Communities

Community Information	Community Size	Total Number of Web Access Flows
India(IN)	48	6594
China(CN)	132	10270
Germany(DE)	132	2787
Russia(RU)	29	1069
Italy(IT)	13	385
Poland(PL)	12	1161
Turkey(TR)	9	3601
Japan(JP)	7	104
Total	262	23971

To compare we also implemented the K-Means clustering algorithm[24], where the distance between two internal users is measured by the Jaccard Similarity between the sets of the external websites these two users have visited. Experimental results on precisions and recall are shown in the bar graph Fig 3, where the height of the bars indicates accuracies and recalls. The results are ordered by the sizes of the ground-truth communities.

From the graph, it can be seen that the LDA-approach performs consistently and significantly better than the K-Means algorithm. More specifically, the algorithm tends to have better performance given more active communities. For

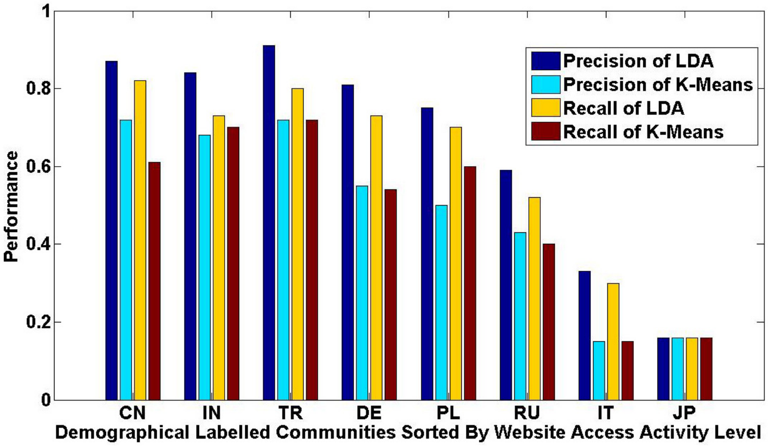


Fig. 3. Detection precisions and accuracies on labeled demographical communities, compared with K-Means clustering algorithm

example, the Russian demographical topic community has a larger size in terms of number of users. However, the LDA algorithm shows better detection for the Turkish community where the number of captured web-access flow records is larger. Also, even though the algorithm is able to detect most communities, there are certain small communities (such as the Japanese and Italian community), which are not very well detected. This is not surprising given that LDA is a Bayesian algorithm where its performance gets better with more data (essentially more stable statistics). Given the smaller size of these communities and lower activity, LDA will not be able to detect these ones given the inherent "resolution" of LDA's application in our problem.

To have a sense of how active a community has to be in order to be detected, we conducted experiments on a relatively large community (the Chinese community). By intentionally deleting some web-access flows randomly, we repeated our experiments to see how the portion of remaining flows will affect detection performance. The graph in Fig4 shows how the detection performance changes with respect to the remaining number of web-access records. While precision is

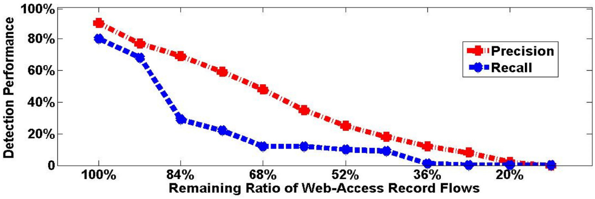


Fig. 4. Performance of LDA communities with Changing Level of Website Access Flow Records Captured

slowing decreasing, recall is more effected by the activities of a community as it goes lower more quickly with smaller activitiy level of the community.

4.3 Expected and Unexpected Communities

Demographical communities, while being measurable against ground truth, are perhaps less interesting and valuable to an enterprise as opposed to other types of subject matter based communities. In addition to comparing with the ground truth communities to evaluate the exact performance of our algorithm, it is also interesting to study the "expected" and "unexpected" communities that were discovered, especially the topic driven ones. Most of these communities are either identifiable by demographically featuring keywords as shown in the domain names, or driven by topical content. We address these as the "expected" communities as we expect our algorithm to be able to discover existing and on-going topics within our institute.

Among the 40 communities discovered, around 50% of them (22 communities) were highly associated with demographical keywords such as a specific country. The remaining communities were more driven by specific keywords such as "linux" and "hack". Note that the results were conducted based on datasets collected from an engineering-focused college. More characteristics of the detected communities are shown in Table 3. Given that our NetFlow data comes from a university, where a significant amount of students are international students, the demographical communities are mainly driven by nationalities and languages of the visited external websites. More interestingly, for the content-based communities, there are many topical communities which we can interpret from the associated keywords. Due to space limit, we present as example, four communities of "Consumer-Electronics", "Hacker-Security", "Deals-Online-Sales", and "Academia-Education-Admission", along with their most frequent keywords in Fig 5. From the keywords associated with these communities, we can infer the topics driving the common interests of the users within these communities.

Table 3. Detected communities and their statistics

Categories	**Total Number of Communities**	**Total Number of Users**	**Example Communities**
Demographical	22	1227	Chinese Indian Korean
Content Based	15	538	Consumer-Electronics Hacker-Security Deal-Sales-Shopping Academia-Education-Admission

"Consumer-Electronics"	"Hackers-Security"	"Deals-Online-Shopping"	"Academia-Education"
ANDROID	HACK	DEAL	EDUCATION
IPHONE	STORM	CREDIT	ADMISSION
MACBOOK	ROOTKIT	DISCOUNT	COLLEGE
CONSUMER	SECURITY	SALE	UNIVERSITY
ACCESSORY	VULERABLE	MONEY	STUDENT
LAPTOP	INJECTION	REBATE	TEACHER
COMPUTER	SPAM	SHOPPING	CONFERENCE
DEAL	VIRUS	AMAZON	COMMITTEE
ULTRABOOK	TROJAN	MYHABIT	SCHOLARSHIP
BLUETOOTH	BOT	EBAY	RANKING
ENGADGET	ZOMBIE	BUY	SALARY
MEMORY	OVERFLOW	ONLINE	JOUNRNAL
LENOVO	BUG	ADVERTISEMENT	PUBLICATION

Fig. 5. Examples of 4 content topical communities discovered: each column represents a topical community. The colored text is the name we give to the community, followed by the most frequent keywords from the websites associated with these users accesses.

In addition to these foreseeable demographical or common topical communities, we also present case studies of two correlated special communities. Both of them have unique keywords distributions making themselves very self-explanatory. These communities we address as "unexpected".

Two Special Communities: The "Apple-Steve-Jobs-Death" Topic and the "Sales-Deals-Buy-Iphone" Topic. By coincidence, the dates when we built the labeled datasets is very close to the major news of the death of Steve Jobs. Interesting, two related topics, one featuring keywords such as "apple, steve-jobs, death", and the other featuring keywords as "sales, deals, iphone", both get discovered in our system. The degree distributions, along with the featuring keywords are shown in Fig6. The associated keywords somehow tell the stories. There is a popular topic about the death of Steve Jobs while there is another overlapping community discussing buying iphone and Android phones. Interestingly, the degree distribution for the first one is very concentrated, meaning that the death of Steve-Jobs is the only dominant topic in this community. However for the "Buy-Iphone" topic, there are two peaks. Looking into the graph structure of the community, it actually has "iphone" and "android" as two clustering centers, thereby making two peaks in the degree distribution.

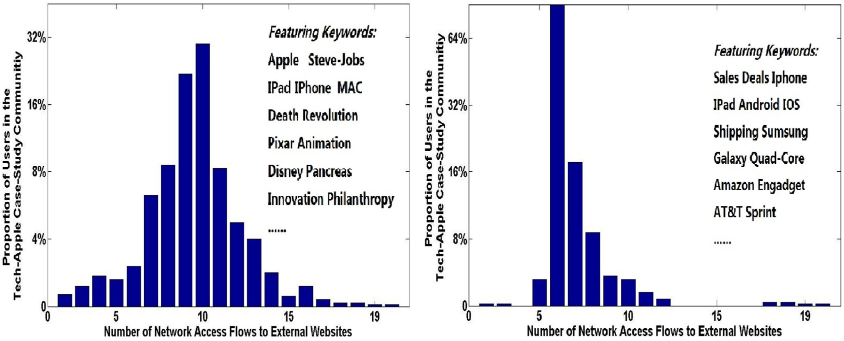


Fig. 6. Degree distribution for the case-studies: (1)Apple-Steve-Jobs community along with featuring keywords (2) The Iphone-Android Sales Community with featuring keywords sorted. Both keywords are sorted by their mined frequencies.

5 Conclusion, Limitations and Future Work

In this paper, we proposed to use Latent Dirichlet Allocation (LDA) as an innovative approach to solve the problem of topical community detection within an enterprise like setting, where we observe the full user-to-website network traffic graph with limited content information. By utilizing the analogy of LDA document-topic-word to user-topic-websites, along with extracted keywords from crawling the external websites, we show that our LDA method can successfully discover many interesting and active topical communities with better precision and recalls than other method(s). The discovered topical communities are either formed by demographical similarities of users, or common interest based on similar content.

Given the Bayesian nature of our approach, our performance is expected to get better with more observed data. However, there is a limit to the length of the time-window on which we conduct our solution. As we used internal IP address to identify an internal user, the IP-Churn [25] problem places a limit on the time window we can employ while still making statistical sense. Once the user's IP changes, the collected flow data is no longer consistent in the sense of indicating internal users. This could be partially solved given the access to DHCP log which records the allocation of IP addresses to physical devices. However, it is not common to assume that the DHCP log is always available. In fact, DHCP log is more privacy sensitive than Netflow data. To overcome this limitation, possible approaches may leverage Hierarchical Dirichlet Process (HDP)[22], where the IP-Churn is also captured by the generative model. We leave this as future work.

References

1. Changuel, S., Labroche, N.: Content Independent Metadata Production as a Machine Learning Problem. In: Perner, P. (ed.) MLDM 2012. LNCS, vol. 7376, pp. 306–320. Springer, Heidelberg (2012)
2. Si, J., Li, Q., Qian, T., Deng, X.: Discovering K Web User Groups with Specific Aspect Interests. In: Perner, P. (ed.) MLDM 2012. LNCS, vol. 7376, pp. 321–335. Springer, Heidelberg (2012)
3. Alshukri, A., Coenen, F., Zito, M.: Incremental Web-Site Boundary Detection Using Random Walks. In: Perner, P. (ed.) MLDM 2011. LNCS, vol. 6871, pp. 414–427. Springer, Heidelberg (2011)
4. Liu, Z., Zhang, Y., Chang, E.Y., Sun, M.: PLDA+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing. ACM Transactions on Intelligent Systems and Technology, Special Issue on Large Scale Machine Learning (2011)
5. Blei, D., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent dirichlet allocation. Journal of Machine Learning Research 3 (2003)
6. Liu, X., He, Q., Tian, Y., Lee, W., McPherson, J., Han, J.: Event-based Social Networks: Linking the Online and Offline Social Worlds. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD, Beijing (2012)
7. Gupta, M., Gao, J., Sun, Y., Han, J.: Integrating Community Matching and Outlier Detection for Mining Evolutionary Community Outliers. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD, Beijing (2012)
8. Sun, Y., Han, J., Yan, X., Yu, P., Yu, X.: Integrating Meta-Path Selection with User Guided Object Clustering in Heterogeneous Information Networks. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD, Beijing (2012)
9. Chaabane, A., Acs, G., Kaafar, M.A.: You are what you like! Information leakage through users' Interests. In: NDSS Symposium, San Diego (2012)
10. Krishnamurthy, B., Wills, C.E.: On the leakage of personally identifiable information via online social networks. In: Proceedings of the 2nd ACM Workshop on Online Social Networks, WOSN 2009, pp. 7–12 (2009)
11. Lin, C.Y., Wu, L., Wen, Z., Tong, H., Griffiths-Fisher, V., Shi, L., Lubensky, D.: Social Network Analysis in Enterprise. Proceedings of the IEEE 100(9), 2759–2776
12. Wen, Z., Topkara, M., Cao, L., Lin, C.Y., Lai, J.: How Multimedia in Enterprise Social Networks Matters to People's Performance. In: IEEE International Conference on Multimedia and Expo Workshops, ICMEW (2012)
13. Jin, Y., Lin, C.Y., Matsuo, Y., Ishizuka, M.: Mining Longitudinal Network for Predicting Company Value. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence, IJCAI 2011, vol. 3, pp. 2268–2273 (2011)
14. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70, 066111 (2004)
15. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821–7826 (2002)
16. Guimera, R., Sales-Pardo, M., Amaral, L.A.N.: Modularity from Uctuations in Random Graphs and Complex Networks. Phys. Rev. E 70 (2004)
17. Zhang, H., Qiu, B., Giles, C.L., Foley, H.C., Yen, J.: An LDA-based Community Structure Discovery Approach for Large-Scale Social Networks. In: IEEE International Conference on Intelligence and Security Informatics (2007)

18. Krestel, R., Fankhauser, P., Nejdl, W.: Latent dirichlet allocation for tag recommendation. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys 2009 (2009)
19. Sachan, M., Contractor, D., Faruquie, T.A., Subramaniam, L.V.: Using Content and Interactions for Discovering Communities in Social Networks. In: World Wide Web Conference, WWW 2012, pp. 331–340 (2012)
20. Henderson, K., Eliassi-Rad, T., Papadimitriou, S., Faloutsos, C.: Hcdf: A Hybrid Community Discovery Framework. In: Proceedings of the SIAM International Conference on Data, ICDM (2010)
21. Zhang, H., Li, W., Wang, X., Giles, C.L., Foley, H.C., Yen, J.: Hsn-pam: Finding Hierarchical Probabilistic Groups from Large-scale Networks. In: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, ICDMW (2007)
22. Zhang, H., Giles, C.L., Foley, H.C., Yen, J.: Probabilistic community discovery using hierarchical latent gaussian mixture model. In: Proceedings of the 22nd National Conference on Artificial Intelligence, AAAI 2007, vol. 1, pp. 663–668 (2007)
23. Vakali, A., Kafetsios, K.: Emotion aware clustering analysis as a tool for Web 2.0 communities detection: Implications for curriculum development. In: World Wide Web Conference, WWW 2012 (2012)
24. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: World Wide Web Conference, WWW 2010 (2010)
25. Xie, X., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., Wobber, T.: How dynamic are IP addresses. In: Proceedings of SIGCOMM (2007)
26. Leighton, T., Rao, S.: An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In: FOCS 1988, pp. 422–431 (1988)

Optimal Time Segments for Stress Detection

Nandita Sharma and Tom Gedeon

Research School of Computer Science
Australian National University, Canberra, Australia
{Nandita.Sharma,Tom.Gedeon}@anu.edu.au

Abstract. Some response signals being modeled for humans over some time segments may not be relevant for analysis and modeling. These signals could contribute to reducing the quality of patterns captured by models, inefficient processing and may impose huge demands on storage resources. This work proposes an approach to search for relevant time segments from human response signals particularly, physiological and physical signals to recognize stress. The paper proposes an approach to determine time segments that were critical to differentiate the types of text based on stress. A support vector machine (SVM) was used to classify the different types of text based on the features of the response signals. A SVM and genetic algorithm (GA) hybrid approach is developed to determine optimal time segments for stress detection (OTSSD). As well as optimizing time segments, the GA also dealt with hundreds of stress features that may have included redundant and irrelevant features. Optimal time segments for stress in reading were successfully found and the GA and SVM hybrid classifier showed an improvement in stress recognition when optimized features from the critical time segments were used.

Keywords: time segments, genetic algorithms, support vector machines, physiological signals, physical signals, stress modeling.

1 Introduction

Human response signals can be used to objectively determine the state of a person in terms of how they feel and react to stimuli at certain points in time or over certain time periods. The main types of response signals to determine the *states* of a human are physiological (e.g. heart rate [1-3] and galvanic skin response [4, 5]) and physical (e.g. facial expression [6-9]) signals. To determine the state of a person under some given circumstance, these signals are captured over some time period to include the data for analyzing the state of the human. However, the signals over the total time period may not be necessary for processing to extract patterns for detecting the state and it may be possible that a smaller segment of the total time period may contain the data sufficient for analysis and modeling. Analysis of the smaller segment could result in significant improvement in efficiency because the effect of irrelevant or redundant data on the analysis will be reduced.

Several computational techniques already reported in literature can be used to select relevant and optimal time segments, but they are not useful for cases when

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 421–433, 2013.

relationship between the characteristics of signals and their features characterizing the human state that needs to be identified are not known. Visual inspection could be used to support selection of a more optimal time segment as reported in literature [10]. The visual analysis could be difficult and prone to errors. Previous research in literature has developed techniques to detect events from time series data and modeled the problems statistically e.g. change point detection problem using maximum likelihood methods [11]. These methods have been applied to traffic data [11] and electrical brain signals (EEG) [12]. For a problem like stress recognition defined in this paper, multiple response signals need sourcing and analysis. This poses a major technical challenge as little information is available on how the various signal combinations can be decoded and how the features affect stress recognition. Feature extraction methods including clustering analysis provide amenable methods to extract and use optimal patterns from time segments. These time segments have to be optimally selected though. Some human intervention may still be required in the process.

Stress is the body's reaction or response to the imbalance caused between demands and resources available to a person. Stress is seen as a natural alarm, resistance and exhaustion [13] system for the body to prepare for a fight or flight response to protect the body from threats and changes. When experienced for longer periods of time and left unmanaged, stress has been recognized as a growing concern in our age adversely impacting society due to its potential to cause chronic illnesses (e.g. cardiovascular diseases, diabetes and some forms of cancer) and high economic costs (especially in developed countries [14, 15]). Benefits of stress research range from improving personal operations, through increasing work productivity to benefitting society - motivating interest, making it a socially beneficial area of research and posing technical challenges in Computer Science. Various computational methods have been used to objectively classify stress to differentiate conditions causing stress from other conditions. The methods developed have used simplistic models formed from techniques like Bayesian networks [16], decision trees [17] and support vector machines [18]. These models have been built from a relatively smaller set of stress features than the sets used in the models in this paper. Further, this work contributes to stress research in the understanding of when individuals respond to stress during a typical activity like reading using a computational approach.

An experiment was conducted where an experiment participant read *stressed* and *non-stressed* text while their physiological and physical response signals were captured. Text was displayed to a participant for a specific time period and of this time period possibly a portion is needed to determine the stress state of participants. It may be that participants showed stress in their response signals while they were reading or they may have showed stress after they finished reading and digested what they read. This paper proposes approaches for selecting the critical time segments during the reading time.

The human body's physiological and physical response signals obtained from non-invasive methods that reflect reactions of individuals to stressful situations have been used to interpret stress levels. Physiological signals include the galvanic skin response (GSR), electrocardiogram (ECG) and blood pressure (BP). Unlike physiological signals, we define physical signals as a time-varying characteristic where changes can be seen by humans without the need for equipment and tools that need to be attached to individuals to detect general fluctuations. However, sophisticated equipment and sensors using vision technologies are still needed to obtain physical signals at sampling rates sufficient for capturing stress patterns. Examples of physical signals are eye fixation and pupil dilation signals. GSR, ECG, BP, eye fixation and pupil dilation signals are the primary stress response signals used in this paper.

The aim of this paper is to determine whether time segments for stress exist that are more relevant to recognize stress in reading. We coin the term *stress episode* to mean a time segment that gives stress patterns in stress response signals required for stress recognition and critical to classify stress. As a consequence of this definition, without stress episodes, two different types of stimuli cannot be differentiated based on stress provided that one stimulus causes stress and the other stimulus does not cause stress. A computational approach is proposed to obtain stress episodes based on a support vector machine (SVM) and a genetic algorithm (GA). The approach also deals with redundant and irrelevant features from stress response signals for stress classification.

The paper presents the reading experiment that was implemented to collect physiological and physical data from which stress features were derived. Then it proposes a hybrid method to search for stress episodes in the features using a GA and a SVM. The GA was implemented to optimize time segments encoded in various forms. The optimization was based on the features and stress classification quality obtained from a SVM. Results of the optimization of the time segments encoded in various forms are presented and discussed. A summary of this work and suggestions for future work are provided as conclusion.

2 Data Collection: Reading Experiment

A reading experiment was done to collect various physiological and physical data from individuals while they read text. Thirty-five undergraduate Computer Science students, compromising 25 males and 10 females, over the age range of 18 to 24 years old were recruited as experiment participants. Each participant had to understand the requirements of the experiment from written experiment instructions with the guidance of the experiment instructor before they filled in the experiment consent form. Afterwards, physiological stress sensors were attached to the participant and physical stress sensors were calibrated. The instructor notified the participant to start reading, which triggered a sequence of text excerpts. After finishing the reading, participants had to do an assessment. To summarize, the process of the experiment for an experiment participant was:

1. Study experiment requirements
2. Provide consent
3. Calibrate sensors
4. Read text
5. Answer survey questions related to the reading

Every participant had their physiological and physical measurements taken over the twelve minutes reading time period. During the reading period, a participant read the experiment instruction text and then read the main text, which was made up of *stressed* and *non-stressed* types of text validated by participants. The instruction text was in the style of the main text for participants to get trial runs of the reading tasks. Stressed text had stressful content in the direction towards distress (e.g. an excerpt about war victims and an excerpt about a ghost in a bedroom), whereas the non-stressed text had content that created an illusion of meditation or soothing environments (e.g. an excerpt about a drive through a scenic terrain and an excerpt about a flower festival) and was not stressful or at least was relatively less stressful compared with the text labeled as stressed. Results from the experiment survey validated the text classes. This is a common method used in literature to validate stress classes for tasks [19]. Participants found the text that were labeled stressed stressful and text labeled non-stressed as not stressful with a statistical significance of $p < 0.001$ according to the T-test.

Each type of text had the same number of text excerpts and each text excerpt was displayed on a computer monitor for participants to read. For consistency, each text excerpt had approximately 120 words. It was displayed on a 1050 x 1680 pixel Dell monitor, displayed for 60 seconds and positioned at the same location of the computer screen for each participant. Each line of the text had 70 characters including spaces.

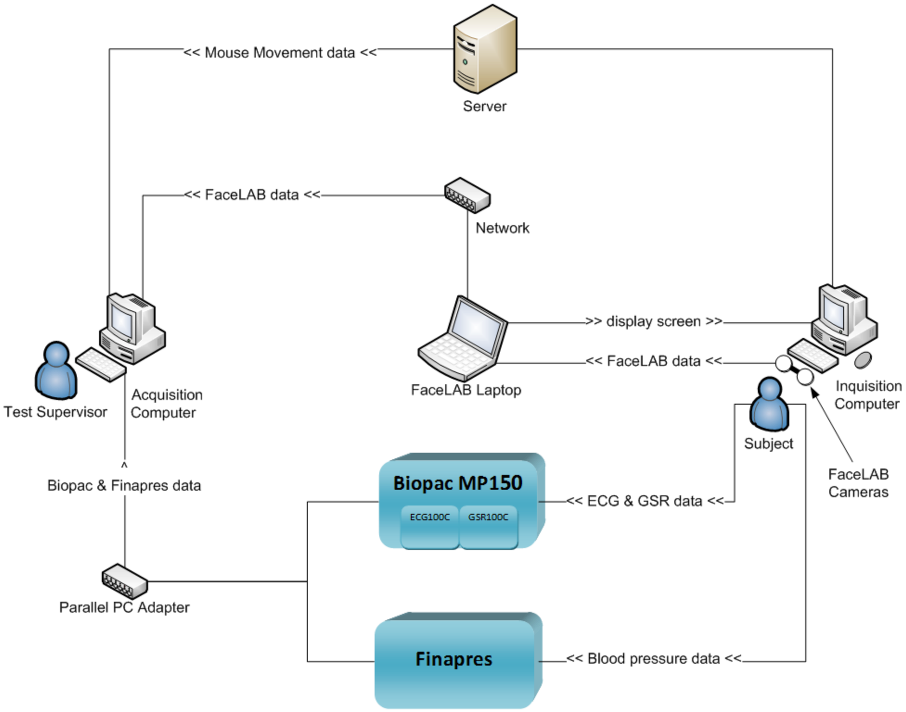


Fig. 1. Equipment setup for the reading experiment

Feature values were derived from physiological and physical signals. Biopac ECG100C, Biopac GSR100C and Finapres Finger Cuff systems were used to take ECG, GSR and blood pressure recordings at a sampling rate of 1000 Hz. Eye gaze and pupil dilation signals were obtained using Seeing Machines FaceLAB system with a pair of infrared cameras at 60 Hz. The equipment setup for the reading experiment is shown in Fig. 1. Other stress symptom signals were derived from primary signals such as, heart rate variability, which was calculated from consecutive ECG peaks and another popular signal used for stress detection [20, 21]. Statistics (e.g. mean and standard deviation) were calculated for the signal measurements for each 5 second interval during the stressed and non-stressed reading. Measures such as the number of peaks for periodic signals, the distance an eye covered, the number of forward and backward tracking fixations, and the proportion of the time the eye fixated on different regions of the computer screen over 5 second intervals were also obtained. Regions of the computer screen are shown in Fig. 2. The statistic and measure values formed the stress feature set. There were 215 features in total. Feature vectors for each participant were normalized by the participant's baseline before they were provided to the classifier.

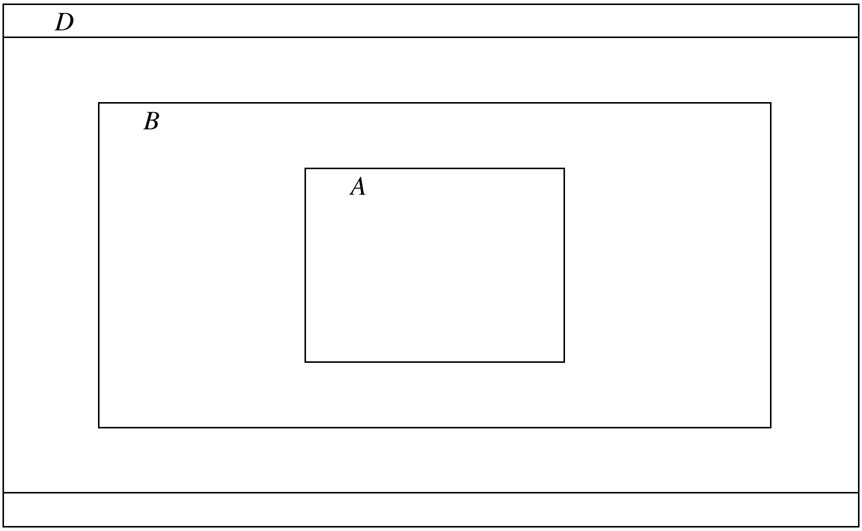


Fig. 2. Bounding rectangles define the different regions of the computer screen to determine a subset of eye gaze features. The bounding rectangles show regions A, B, C and D. Region A contains the text area where text was displayed to a participant. Region B is the region without A and regions C and D are defined similarly. Region D had the application menu and the toolbar. Note that the diagram is not drawn to scale.

A participant was shown a segment of text for a set period of time for reading. According to the reading rate for an average person and considering the number of words per line and character size, sufficient time was given to participants to read the text [22]. This was validated by the participants' eye movement data patterns and survey responses. In addition, results from the survey responses showed that participants easily understood the text. However, the reading time provided may have been longer than necessary to show stress. Using stress response signals over the total reading time may bury the critical time segments that have stress symptoms. As a result, the data over the irrelevant time segments may be irrelevant and could outweigh the data that has stress symptoms. This may have a negative impact on stress classification.

3 A Genetic Algorithm for Feature and Time Segment Selection

GAs have been widely used for optimization problems. A GA is a global search method and it has been successfully used for feature selection from physiological signals for human state classification [23]. This work uses a GA for feature selection from various stress response signals and proposes a GA based approach to find stress episodes in the reading data.

A GA is based on the concept of natural evolution. It evolves a population of candidate solutions, represented by *chromosomes*, in search for better quality chromosomes. The approach applies *crossover* and *mutation* operations to the chromosomes to achieve diversity in the population and reduce risk of the search to reach a local optimal population. After each iteration during the search, the GA *selects* chromosomes, mostly made up of better quality solutions, for the population in the next iteration to direct the search to more favorable chromosomes. In this work, the quality for a chromosome is based on the accuracy, sensitivity, specificity and the F-score values of the stress classifications using the information represented by the chromosome.

A chromosome for the problem for stress recognition in reading either represented time segments as candidate stress episodes, a subset of features from the set of all stress features or a combination of both. Conventionally, a chromosome has encoding for one type of a solution e.g. features [23]. In the case for this work, the goal was to find stress episodes and features over the stress episodes that produced better stress recognition results. Hybrid chromosomes have been developed and used in literature [24, 25] but to the best of our knowledge, it has not been used for this kind of work particularly, for encoding signal and time segment data.

Each chromosome was a binary string of fixed length. For the chromosome that encoded features, which we denote as C_F, the chromosome contained information for all the features in the space of all stress features. The index of the chromosome represented a feature and its value indicated whether the feature was used in the classification to define the corresponding model. The length of the chromosome was equal to the number of stress features. On the other hand, there were three different ways in which time segments were encoded. A time segment chromosome had one of the following definitions:

- C_{TS}. An index of the chromosome represented a time segment of the total reading time. The value for the index represented whether the feature values over the corresponding time segment was used in the classification.
- C_{OTS}. A chromosome that had overlapping time segments where an index of the chromosome represented a time segment of the total reading time. Similar to C-TS, the value for the index represented whether the feature values over the time segment was used in the classification.

- C_{TTS}. A chromosome that had the start and end times of a time segment where the first half of the chromosome had the start time of the time segment and the second half had the end time. Stress feature values over the time segment used in the classification.

Another type of chromosome was developed that had encodings for both features and time segments. The classification model was defined by the feature values during the time segments depicted in the time segment encoding component of the chromosome. There were three variations for this type of chromosome and they differed on the time segment encoding component:

- $C_{F\text{-}TS}$. A composite structure with encodings for features and time segments of the total reading time. An index of time segment encoding component of the chromosome represented a time segment in the total reading time and its value represented whether the feature values in the feature encoding component was used in the classification.
- $C_{F\text{-}OTS}$. A composite structure with encodings for features and overlapping time segments of the total reading time. An index of time segment encoding component of the chromosome represented a time segment in the total reading time and its value represented whether the feature values in the feature encoding component was used in the classification.
- $C_{F\text{-}TTS}$. A composite structure with encodings for features and start and end times of a time segment. The definition for time segment encoding component was based on $\mathrm{C_{TTS}}$. Feature values depicted by the feature encoding component over the time segment was used in the classification.

4 A Genetic Algorithm and Support Vector Machine Hybrid Stress Classifier

A stress recognition approach consisting of a hybrid of a GA and SVM was used to determine OTSSD. The GA searched for stress episodes based on the quality of the stress classifications produced by the SVM given subsets of stress features and time segments over the reading time.

SVMs have been widely used in literature for classification problems based on physiological data [26]. A SVM constructs a maximum margin separator, to separate data into classes. It transforms the data to a higher dimensional space where it constructs a linear separating hyperplane. The hyperplane is a decision boundary with the largest possible distance to example points. This helps to generalize classifications well and have been claimed to be resistant to overfitting the data. However, the classification performance of a SVM is subject to the features provided as input. The GA provided a set of features over GA selected time segments to the SVM to determine the quality of stress classifications and evolved the set of features and time segments in search for stress episodes.

To obtain stress classification results from a SVM, the stress reading data set was divided up into 3 subsets – training, validation and test sets – where 50% of the data samples were used for training the SVM model and the rest of the data set was divided up equally to validate and test the model. MATLAB was used to implement and test the models.

The parameters for the GA implemented were set as provided in Table 1.

Table 1. Parameter settings for GAs that used to find optimal time segments and feature selection

GA Parameter	Value/Setting
population size	100
number of generations	2000
crossover rate	0.8
mutation rate	0.01
crossover type	scattered crossover
mutation type	uniform mutation
selection type	stochastic uniform selection

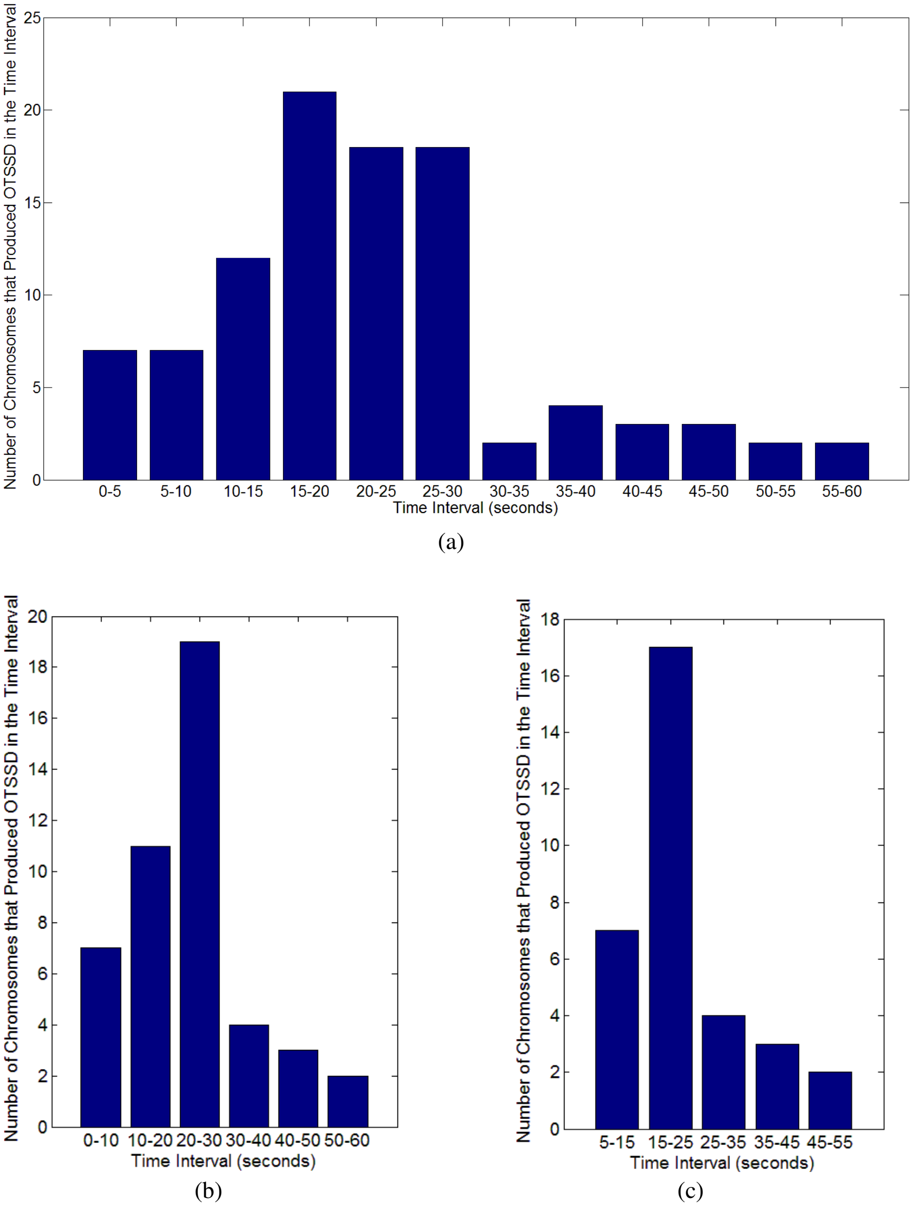
5 Results and Discussion

The GA-SVM hybrid approach was implemented using each of the different chromosomes to search for stress episodes in reading. GA-SVMs with C_{TS}, C_{OTS}, $C_{F\text{-}TS}$ and $C_{F\text{-}OTS}$ were implemented with different time segment intervals. The interval sizes were chosen based on the time segment size for feature extraction, which was 5 seconds and detailed earlier in section 2. The GA provided multiple time segments at the end of the GA search from which the longest time segment that was defined by the multiple consecutive time segment intervals was chosen as a stress episode. Results of the GA and SVM approach using the different types of chromosomes to search for OTSSD are presented in Table 2. The interval length for a time segment in each of the chromosomes C_{TS}, C_{OTS}, $C_{F\text{-}TS}$ and $C_{F\text{-}OTS}$ is depicted by *i* seconds in C_{TS_i}, C_{OTS_i}, $C_{F\text{-}TS\_i}$ and $C_{F\text{-}OTS\_i}$ respectively.

Participants of the reading experiment were given 60 seconds to read a text excerpt. Results in Table 2 show that stress episodes in the reading data appeared within a smaller segment of the total reading time in particular, they appeared around 15-30 seconds. During this time interval, the SVM produced the highest stress recognition rates. Fig. 3 shows the number OTSSD searches that produced stress episodes during various time intervals of the total reading time. Each OTSSD search used one type of chromosome listed in Table 2. The distribution of the number of OTSSD searches show that most of the searches produced stress episodes between 15-30 seconds.

Table 2. Summary of the results obtained for different types of chromosomes used to find stress episodes in the reading data set. The table shows stress episodes obtained from the different types of chromosomes and their performances in stress recognition based on 10-fold cross-validation are provided.

Chromosome	Stress episode interval (seconds)	Recognition rate	F-score
C_{TS}			
C_{TS_60}	0-60	0.67	0.67
C_{TS_30}	0-30	0.75	0.72
C_{TS_20}	0-20	0.85	0.86
C_{TS_15}	15-30	0.89	0.88
C_{TS_10}	10-20	0.90	0.89
C_{TS_5}	15-30	0.92	0.90
C_{OTS}			
C_{OTS_30}	0-30	0.75	0.72
C_{OTS_20}	10-30	0.85	0.86
C_{OTS_15}	15-30	0.89	0.88
C_{OTS_10}	15-30	0.89	0.90
C_{OTS_5}	15-30	0.92	0.90
C_{TTS}			
C_{TTS}	15-30	0.92	0.90
$C_{F\text{-}TS}$			
$C_{F\text{-}TS\_60}$	0-60	0.76	0.74
$C_{F\text{-}TS\_30}$	0-30	0.88	0.90
$C_{F\text{-}TS\_20}$	0-20	0.93	0.91
$C_{F\text{-}TS\_15}$	15-30	0.99	0.98
$C_{F\text{-}TS\_10}$	10-20	0.93	0.91
$C_{F\text{-}TS\_5}$	15-30	**0.99**	**0.98**
$C_{F\text{-}OTS}$			
$C_{F\text{-}OTS\_30}$	20-50	0.80	0.82
$C_{F\text{-}OTS\_20}$	20-40	0.87	0.86
$C_{F\text{-}OTS\_15}$	15-30	0.99	0.98
$C_{F\text{-}OTS\_10}$	10-20	0.93	0.91
$C_{F\text{-}OTS\_5}$	15-30	**0.99**	**0.98**
$C_{F\text{-}TTS}$			
$C_{F\text{-}TTS}$	15-30	**0.99**	**0.98**



(a)

(b)

(c)

Fig. 3. Distribution for the stress episodes produced by OTSSD searches where each search used one type of chromosome in Table 2 over the reading time. Frequency of the OTSSD searches that produced stress episodes within various time intervals: (a) 5 seconds (b) 10 seconds starting from 0 seconds and (c) 10 seconds starting from 5 second.

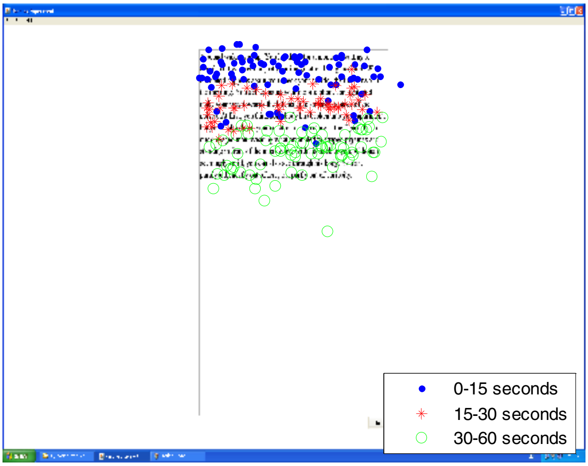


Fig. 4. Average eye fixation coordinates of experiment participants reading a text excerpt over different time intervals of the total reading time

Searches for OTSSD that optimized features produced better stress recognition rates than searches with chromosomes that only optimized time segments. The searches that optimized features had chromosomes that had a feature encoding component. The average stress recognition rate for searches that optimized features was 0.92 whereas the average stress recognition rate searches without feature optimization was 0.85. Without feature optimization, redundant features and features irrelevant to SVM based stress recognition may have existed. These features were reduced with the use of a GA in search for OTSSD, which produced a better stress recognition rate. The best recognition rate was 99% and it was produced by the OTSSD search that optimized features and found a stress episode between the 15-30 reading time interval.

OTSSD searches with chromosomes with time segment intervals that evolved the start and end times of the time segment interval and time segments with relatively short time intervals, in particular 5 seconds, found stress episodes with higher stress recognition rates. Using chromosomes C_{TTS} and $C_{F\text{-}TTS}$ in the search that evolved the start and end times for candidate stress episodes provided more flexibility in the search for OTSSD because the search space was less constrained. Chromosomes with time segments that had 5 second intervals also had a similar characteristic. On the other hand, the design for other chromosomes made it difficult or impossible for the OTSSD search to find stress episodes with higher stress recognition rates.

To relate the time interval for stress episodes found by the OTSSD searches to an average reading excerpt in the experiment, average eye fixations of participants in relation to the reading display during the reading time are presented in Fig. 4. In accordance to the stress episode for reading, participants approximately read a third of the text before the start of the stress episode. They read over half of the text in total by the end of the stress episode. Investigation of the content of a text excerpt with division of text by content and its relationship with stress was beyond the scope for this work. Future research could investigate what divisions of a text excerpt caused stress episodes.

6 Conclusion and Future Work

A computational approach for optimal time segments for stress detection (OTSSD) was developed to determine optimal time segments, or stress episodes, with optimized stress features to obtain necessary stress data and improve stress recognition in reading. A GA and SVM based hybrid technique was used where the GA searched for stress episodes based on the classification results produced by the SVM. The GA was extended to include stress feature optimization as well. Stress episodes were successfully found and results showed an improvement in SVM based stress recognition when a GA was used to select appropriate features and relevant time segments for stress during the reading period. The GA and SVM hybrid proposed could be extended to search for stress episodes or other types of critical time segments for other types of signal based classification for efficient storage, processing and analysis. There may be a possibility that certain feature signals correspond to stress levels more strongly than other feature signals during a particular time segment. To consider such a case and possibly different latency times for stress symptoms from the different response signals, future work could investigate developing classification models with time segment selection that are adaptable to features individually. Further, future research could extend statistical methods for event detection reported in literature for the multi-signal reading stress data.

References

[1] Vu, T.H.N., Park, N., Lee, Y.K., Lee, Y., Lee, J.Y., Ryu, K.H.: Online discovery of Heart Rate Variability patterns in mobile healthcare services. Journal of Systems and Software 83, 1930–1940 (2010)

[2] Vu, T.H.N., Lee, J.W., Lee, Y., Kim, H., Ryu, K.H.: An HRV Patterns Discovering Neural Network for Mobile Healthcare Services, pp. 92–97 (2008)

[3] Kana, M., Jiina, M., Holík, J.: Estimation of Sympathetic and Parasympathetic Level during Orthostatic Stress Using Artificial Neural Networks. Recent Advances in Mechatronics, 431–436 (2010)

[4] Shi, Y., Ruiz, N., Taib, R., Choi, E., Chen, F.: Galvanic skin response (GSR) as an index of cognitive load. In: Extended Abstracts on Human Factors in Computing Systems, CHI 2007, San Jose, CA, USA, pp. 2651–2656 (2007)

[5] Bundele, M.M., Banerjee, R.: Detection of fatigue of vehicular driver using skin conductance and oximetry pulse: a neural network approach. In: Proceedings of the 11th International Conference on Information Integration and Web-Based Applications and Services, Kuala Lumpur, Malaysia, pp. 739–744 (2009)

[6] Jabon, M., Bailenson, J., Pontikakis, E., Takayama, L., Nass, C.: Facial Expression Analysis for Predicting Unsafe Driving Behavior. IEEE Pervasive Computing 10, 84–95 (2010)

[7] Lerner, J.S., Dahl, R.E., Hariri, A.R., Taylor, S.E.: Facial expressions of emotion reveal neuroendocrine and cardiovascular stress responses. Biological Psychiatry 61, 253–260 (2007)

[8] Kurose, H., Ito, K., Nishida, S.: A Method for Selecting Facial Expression based on Emotions and Its Application for Text Reading. In: IEEE International Conference on Systems, Man and Cybernetics (SMC 2006), Taipei, pp. 4028–4033 (2006)

[9] Pantic, M., Bartlett, M.S.: Machine analysis of facial expressions. In: Face Recognition, pp. 377–416 (2007)

[10] Bakker, J., Pechenizkiy, M., Sidorova, N.: What's your current stress level? detection of stress patterns from gsr sensor data. In: International Conference on Data Mining Workshops (ICDMW), Vancouver, BC, pp. 573–580 (2011)

[11] Guralnik, V., Srivastava, J.: Event detection from time series data. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 33–42 (1999)

[12] Cecotti, H., Phlypo, R., Rivet, B., Congedo, M., Maby, E., Mattout, J.: Impact of the time segment analysis for P300 detection with spatial filtering. In: 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), pp. 1–5 (2010)

[13] Hoffman-Goetz, L., Pedersen, B.K.: Exercise and the immune system: a model of the stress response? Immunology Today 15, 382–387 (1994)

[14] The-American-Institute-of-Stress (March 5, 2012) America's No. 1 Health Problem - Why is there more stress today? `http://www.stress.org/americas.htm`

[15] Lifeline-Australia (2009), Stress Costs Taxpayer $300K Every Day, `http://www.lifeline.org.au` (March 15, 2013)

[16] Liao, W., Zhang, W., Zhu, Z., Ji, Q.: A real-time human stress monitoring system using dynamic bayesian network. In: Computer Vision and Pattern Recognition - Workshops, p. 70 (2005)

[17] Zhai, J., Barreto, A.: Stress recognition using non-invasive technology. In: Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference FLAIRS, pp. 395–400 (2006)

[18] Dou, Q.: An SVM ranking approach to stress assignment. University of Alberta (2009)

[19] Hill, J.D., Boyle, L.N.: Driver stress as influenced by driving maneuvers and roadway conditions. Transportation Research Part F: Traffic Psychology and Behaviour 10, 177–186 (2007)

[20] Ferreira, P., Sanches, P., Höök, K., Jaensson, T.: License to chill!: how to empower users to cope with stress. In: Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, pp. 123–132 (2008)

[21] Dishman, R.K., Nakamura, Y., Garcia, M.E., Thompson, R.W., Dunn, A.L., Blair, S.N.: Heart rate variability, trait anxiety, and perceived stress among physically fit men and women. International Journal of Psychophysiology 37, 121–133 (2000)

[22] Ziefle, M.: Effects of display resolution on visual performance. Human Factors: The Journal of the Human Factors and Ergonomics Society 40, 554–568 (1998)

[23] Hosseini, S.A., Khalilzadeh, M.A.: Emotional stress recognition system using EEG and psychophysiological signals: Using new labelling process of EEG signals in emotional stress state. In: International Conference of Biomedical Engineering and Computer Science (ICBECS), pp. 1–6 (2010)

[24] Tamminedi, T., Ganapathy, P., Zhang, L., Yadegar, J.: Classifier fusion framework using genetic algorithms. In: International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), Toronto, ON, pp. 2224–2228 (2011)

[25] Huang, H., Yang, X., Hao, Z., Wu, C., Liang, Y., Zhao, X.: Hybrid chromosome genetic algorithm for generalized traveling salesman problems. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 137–140. Springer, Heidelberg (2005)

[26] Cheng, B.: Emotion recognition from physiological signals using support vector machine. In: Wu, Y. (ed.) Software Engineering and Knowledge Engineering. AISC, vol. 114, pp. 49–52. Springer, Heidelberg (2012)

Personalized Expert-Based Recommender System: Training C-SVM for Personalized Expert Identification

Yeounoh Chung, Hye-Wuk Jung, Jaekwang Kim, and Jee-Hyong Lee

Information and Intelligence Systems Laboratory, Sungkyunkwan University, Suwon, Korea
yeounohster@gmail.com, {wukj,linux,john}@skku.edu

Abstract. In order to improve the performance of the existing recommendation algorithms, previous researches on expert-based recommender systems have exploited the knowledge of experts. However, the previous expert-based recommender systems are limited in that the same experts are suggested for all users. In this paper, we study personalized expert identification problem, assuming each user needs different kinds and levels of expert help. We demonstrate the feasibility of personalized expert-based recommendation; we present and analyze an SVM framework for finding personalized experts.

Keywords: Expert-based Recommender System, Collaborative Filtering (CF), Support Vector Machine (SVM), Personalized Expert.

1 Introduction

With the success of many e-commerce services (e.g. Amazon, Netflix, Last.fm), recommender systems have gained much interests and popularity over years; there have been much research efforts to build better recommender systems and algorithms.

A typical recommender system builds up user profiles based on users' past records; and the system provides personalized recommendations based on either collaboration of users or similarity of items; many variations of the above approach have been introduced to enhance performance of the existing recommendation algorithms.

One of the interesting ideas suggested in recent studies is to use expert knowledge for generating recommendations. Real users often trust more reliable and knowledgeable experts in making decisions to buy products. Following this observation, recent studies on expert-based recommender systems have exploited the knowledge of the real-world experts [1] or that of the experts identified among users from a user group [2]. Although the suggested approaches showed some promising results, those approaches are still limited in that the same experts are suggested for all users.

Users have different needs and expectations. Hence, it is more intuitive to suggest different groups of experts to different users. In this paper, continuing our earlier work on personalized expert-based recommender system [8], we study a problem of identifying a personalized expert group for each user, within the same user group; preferences of personalized experts of a user are used to generate recommendations for the user.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 434–441, 2013.

To demonstrate the idea, we translate the personal expert identification problem into an SVM optimization problem; we present a framework for using Support Vector Machine (SVM, C-SVM) to find varying expert groups for users. However, use of SVM requires generating training data with proper expert labels; in the context of real-world applications, it is much preferable to label data without any user feedback. Here we approximate the optimal solution or personalized expert groups using a random search based method and label training data using the approximated solution.

As noted in the previous studies, evaluating each user's recommendation performance gain to identify optimal experts is too cost intensive. To reduce the computational complexity, we assume that the personal experts of a user are somewhat similar in preference to the user. Rather than searching for the optimal experts across the entire user group, we reduce the search space to a handful of the most similar users.

It is shown in an experiment that the personalized expert-based recommender system outperforms the traditional nearest neighbor algorithm. Furthermore, the proposed SVM approach can identify personalized experts, even with a class imbalanced training dataset.

The rest of this paper is organized as follows. In Section 2, we discuss related works in expert-based recommender systems and recommender systems using SVM. In Section 3, we describe the personal expert identification problem, as well as the two approaches to solve the problems: random search algorithm and SVM framework for personalized expert identification. In Section 4, we discuss our experimental results. Finally, we conclude in Section 5.

2 Related Work

2.1 Expert-Based Recommender Systems

Previous works on expert-based recommendation systems have exploited the knowledge of the common experts. Amatrianin et al. [1] use explicit expert data (from a movie review website) to model preferences of a much larger user group; they show that one can utilize expert knowledge to generate recommendations for users, yielding comparable results to traditional collaborative filtering algorithms.

Sang et al. [2] suggest a different approach to exploit expert knowledge in recommender systems. They propose three different expert measures and identify expert users using a latent variable model; the opinions of expert users are used for generating recommendations for other users.

Both approaches use the same experts to generate recommendations for all other users. However, it is more intuitive to consult personalized experts to generate recommendations for each user. In our initial work on expert-based recommender system [8], we suggest a personalized expert-based recommender system, assuming each user needs different kinds and levels of expert help

2.2 Recommender Systems Using SVM

Standard SVM classifier has not been very successful in recommender systems due to the sparse problem, in which there are insufficient transactional data for training [6].

Xia et al. [7] proposed a heuristic method to fill in the missing transactional data for training a SVM classifier for recommendable items. In their proposed recommender system, a SVM classifier is built for each user, treating every user as a different problem. The computational cost for generating and updating different SVM models for every other user makes the approach impractical for real-world applications.

Similarly, Xu et al. [3] suggested building a personalized SVM model to recommend TV programs for each user. In addition to the computational cost for building an SVM model for every other user, their approach uses user feedbacks to train SVM models. Such feedback information is often not available in real-world applications.

In this work, we train a single global SVM classifier to identify a personalized expert group for each and every user, instead of directly generating recommendation lists; next, recommendations are generated by a collaborative filtering algorithm based on the expert groups, without explicit user feedback. The proposed approach is computationally more favorable than previous SVM-based recommender systems as it builds a single SVM model for all users, instead of a single user.

3 Personalized Expert Identification

3.1 Problem Statement

A typical recommendation system builds up user profiles based on users' past records; and the system provide personalized recommendations based on either collaboration of users or similarity of items. Given a user set, $\mathrm{U} = \{u_i\}$, an item set, $\mathrm{I} = \{i_j\}$, and the associated ratings, denoted as $r_{u_i i_j}$, we want to identify a personalized expert group V_u for each user. In this problem, we define experts as users, whose preferences are more valuable in generating more accurate recommendations.

Personalized expert identification problem can be formalized as follows:

$$\underset{f_{\vec{w}}}{argmax} \sum_i \tau\left(V_{f_{\vec{w}}(u_i)}, V_{u_i}^*\right) \tag{1}$$

$V_{u_i}^* = \{v_{ij}^*\}$ is an optimal personalized expert set for u_i. The personalized expert group for u_i, $V_{f_{\vec{w}}(u_i)}$ is identified using a function $f_{\vec{w}}$ learned by an SVM classifier; $\tau\left(V_{f_{\vec{w}}(u_i)}, V_{u_i}^*\right)$ measures the similarity between the two sets; We find $f_{\vec{w}}$ that gives near optimal personalized experts for all users.

3.2 Random Search Algorithm for Personalized Experts

Evaluating each user's recommendation performance gain to identify optimal experts is cost intensive. To reduce the computational complexity, we assume that the personal experts of a user are somewhat similar in preference to the user. Rather than searching for the optimal experts across the entire user group, we reduce the search space to a handful of the most similar users.

Within the reduced search space, $\mathrm{U}_i = \{u_j\}$ for u_i, we randomly choose a fixed number of users, $V_{u_i}^{'}$, to recommend items to u_i using nearest neighbor collaborative filtering algorithm; this random search procedure is repeated many times; $V_{u_i}^{'}$ with the best prediction accuracy is taken to be an approximated solution for $V_{u_i}^{*}$.

Approximating optimal personalized experts for every user is still very expensive. In the following section, we present an SVM framework for finding personalized experts. Here we label training data using the approximation results, but the expensive random search based method is only used in training phase; the single generated SVM model can be used for all users, classifying a personalized expert group for each user.

3.3 SVM for Personalized Experts

We translate the personalized expert identification problem into a SVM optimization problem:

$$min_{\overrightarrow{W}} \frac{1}{2} \overrightarrow{W} \cdot \overrightarrow{W} + C \textstyle\sum_i \sum_j \varepsilon_{ij} \tag{2}$$

$$Subject\ to\ y_{ij}\left(\overrightarrow{W} \cdot \overrightarrow{X_{ij}} + b\right) \geq 1 - \varepsilon_{ij},$$

$$for\ all\ \left\{\left(\overrightarrow{X_{ij}},\ y_{ij}\right)\right\}\ and\ \varepsilon_{ij} \geq 0\ for\ all\ i, j$$

Here we introduce expertise $\overrightarrow{X_{ij}}$, which indicates how much u_i views u_j as an expert. Given an optimal expert group for u_i, we get y_{ij} with corresponding $\overrightarrow{X_{ij}}$; we train SVM to learn personalized expert identification function $f_{\overrightarrow{W}}$.

The above formulation can be further generalized to a C-SVM optimization problem as follows:

$$min_{\overrightarrow{W}} \frac{1}{2} \overrightarrow{W} \cdot \overrightarrow{W} + C_p \textstyle\sum_i \sum_j \varepsilon_{ij} + C_n \sum_i \sum_j \varepsilon_{ij} \tag{3}$$

$$Subject\ to\ y_{ij}\left(\overrightarrow{W} \cdot \overrightarrow{X_{ij}} + b\right) \geq 1 - \varepsilon_{ij},$$

$$for\ all\ \left\{\left(\overrightarrow{X_{ij}},\ y_{ij}\right)\right\}\ and\ \varepsilon_{ij} \geq 0\ for\ all\ i, j$$

C_p and C_n control the tradeoff between training errors and margin maximization for positive and negative examples, respectively. By tuning the cost factor, C_p/C_n, one can more effectively learn from class imbalanced data sets.

In addition to the above SVM formulations, we also suggest using a threshold value for SVM output in testing. In this way, only testing examples with larger margin (output value larger than the threshold) are chosen to be personalized experts for a user. In this work, we use three times standard deviation of $\left(\overrightarrow{W} \cdot \overrightarrow{X_{ij}} + b\right)$ as the threshold value for u_i.

3.4 Expertise

In plain English, expertise $\overrightarrow{X_{\imath\jmath}}$ indicates how much u_i views u_j as an expert. We measures expertise between every pair of users, with respect to four expertise measures. First, we measure similarity of u_i and u_j by adjusted cosine similarity:

$$\frac{\sum_{m\in I(u_i)\cap I(u_j)}(R_{u_i m}-\overline{R_{u_\imath m}})(R_{u_j m}-\overline{R_{u_\jmath m}})}{\sqrt{\sum_{m\in I(u_i)\cap I(u_j)}(R_{u_i m}-\overline{R_{u_\imath m}})^2 \sum_{m\in I(u_i)\cap I(u_j)}(R_{u_j m}-\overline{R_{\jmath m}})^2}} \tag{4}$$

Second, we measure how much often u_j accesses the service than u_i. We take the difference between the item set accessed by u_i and the item set accessed by u_j, and then we normalize the difference by the size of the whole item set. Note that this measure can be negative if u_i is more heavy user than u_j:

$$\frac{|I(u_j)|-|I(u_i)|}{|I|} \tag{5}$$

Third, we measure how many new items u_j has rated. If $u_i$has not rated item i_j, then i_j is a new item to u_i. Note that if u_j has no new items, then the relative complement of $I(u_j)$ in $I(u_i)$ would be an empty set:

$$\frac{|I(u_j)-I(u_i)|}{|I|} \tag{6}$$

Lastly, we measure average rating difference between u_i and u_j:

$$\frac{\overline{R_{u_\jmath}}-\overline{R_{u_\jmath}}}{\max \quad \bar{R}-\min \quad \bar{R}} \tag{7}$$

Furthermore, in constructing the mapping between features of u_i and u_j, we only use users' preference, $< u_i, i_j, r_{u_i i_j} >$, and no explicit user feedback; those four measures are used as features for training SVM models.

4 Experimental Results

4.1 Dataset

Due to the computational challenge in dealing with large-scale data, we used a subset of Netflix challenge dataset for the evaluation of our system. The subset data contains 10,000 users and 1,000 movies, while the original dataset contains 480,189 users and 17,770 movies. In addition, we only use users' preference information, $< u_i, i_j, r_{u_i i_j} >$. The dataset is randomly divided into a training data of 9,000 users and a testing data of 1,000 users.

4.2 k-Nearest Neighborhood

We use k-Nearest Neighbor (kNN) algorithm as a reference. The lowest Root Mean Squared Error (RMSE) is 0.7773 when the nearest neighborhood size is 17. As it is shown in Figure 1, taking more opinions (larger neighborhood) does not guarantee more accurate rating prediction.

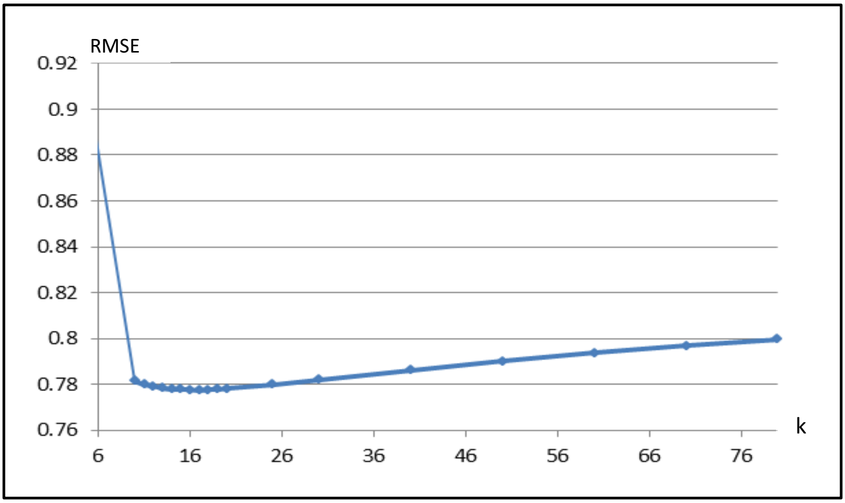


Fig. 1. kNN recommendation accuracy (RMSE) for different neighborhood size (k)

4.3 Personalized Expert

In personalized expert-based recommender system evaluation, we first fix the number of (optimal) personalized experts to the best size of nearest neighborhood, $k = 17$, to show that opinions of personalized experts are more valuable in recommending items. Random search based method described in Section 3.2 is used to identify the optimal personalized experts for each user.

For every user in the dataset, we search 17 personalized experts using a random search algorithm. Due to the computational complexity, we reduce the expert search space to 17×3 most similar users. This reflects our assumption that personalized experts of a user are similar to the user with some degrees. Furthermore, three times the size of the expert group is reasonably small enough (note, $\binom{51}{17}$ is still a huge number).

The search for the optimal expert group is repeated over 3,000 times for each user; the testing prediction error (RMSE) is 0.0596, which is 92% improvement over the best result of kNN. The recommendation accuracy is surprisingly high, and this near perfect accuracy is mainly due to our definition of expert: the algorithm simply searches for neighborhoods that give the best recommendation accuracy for each user. The significant improvement over the reference result (kNN) indicates that there exist personalized expert groups, which can be exploited for more accurate recommendations.

The computational cost of running the algorithm motivates us for a model-based approach; we use the optimal personalized expert groups to label training data for SVM.

4.4 SVM for Personalized Expert

We train SVM and C-SVM to identify any number of personalized experts for each user. Using SVM classifier models, we can much more efficiently identify personalized expert groups for users; we do not fix the size of the personalized experts as in the experiments using random search algorithm. This allows the number of personalized experts vary for each user, utilizing just the right amount of expert knowledge.

The trained SVM model identifies personalized experts for every other user in the testing data; the recommendation accuracy for personalized expert-based recommender system using SVM and C-SVM are shown in Table 1.

Table 1. Personalized expert-based recommender system accuray (error) measures

	kNN	PE-RS	PE-SVM	PE-CSVM	PE-SVM	PE-CSVM
RMSE	0.7773	0.0596	1.1550	1.2742	0.9968	0.9638
k	17	17	17	17	214	971

Table 2. Testing results of personalized expert-based recommender system using C-SVM

C_p/C_n	2	3	4	5
Accuracy	77.63	73.65	70.31	66.24
Precision	69.93	65.16	61.75	58.14
Recall	90.00	91.67	92.98	94.69
RMSE	0.9659	0.9638	0.9659	0.9682
k	765	971	1251	9998

Personalized expert-based recommender systems using SVM and C-SVM are not as good as the reference (kNN). However, if SVM classifiers can identify the optimal personalized experts more accurately, then PE-SVM and PE-CSVM would outperform kNN as the random search-based approach (PE-RS) does.

We suspect that the poor performances of the SVM classifiers are mainly due to the class imbalanced training dataset. Furthermore, as shown in Table 2, forcing C-SVM classifier to handle positive examples with care ($C_p > C_n$) makes the classifier to classify most users as experts, with high recall and low precision values.

The computational efficiency of SVM makes our personalized expert-based recommender system more feasible, compared to the case of using random search algorithm. In the future, we plan on studying ways to improve the classification performance of the SVM-based models in this context.

5 Conclusion

In this work, we study a personalized expert identification problem in the context of personalized expert-based recommender system.

Our experimental results show that there exist personalized expert groups, which can be exploited for generating accurate recommendations. Furthermore, we present and analyze SVM/C-SVM framework for personalized expert identification; with a proper handling of class imbalance dataset, SVM-based models can be computationally efficient alternatives to the random search algorithm.

In the near future, we will investigate the meaning of personal expertise and its feature space. We look to devise better features for personalized expertise. Semi-supervised learning and probabilistic approaches are other possibilities we need to examine for personalized expert identification.

Acknowledgements. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No.2012-008062) and IT R&D program of MKE/KEIT. [KI001810041244 , SmartTV 2.0 Software Platform]

References

1. Amatrian, X., Lathia, N., Pujol, J.M., Kwak, H., Oliver, N.: The Wisdom of The Few: A Collaborative Filtering Approach Based On Expert Opinions From The Web. In: 32nd International Conference ACM Special Interest Group on Information Retrieval, New York, pp. 532–539 (2009)
2. Song, S., Lee, S., Park, S., Lee, S.G.: Determining User Expertise For Improving Recommendation Performance. In: 6th International Conference on Ubiquitous Information and Communication 2012, New York (2012)
3. Xu, J.A., Araki, K.: A SVM-based Personal Recommendation System for TV Programs. In: 12th International Conference on Multi Media Modeling (2006)
4. Min, S.-H., Han, I.: Recommender Systems Using Support Vector Machines. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 387–393. Springer, Heidelberg (2005)
5. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
6. Zhang, T., Iyengar, V.: Recommender Systems Using Linear Classifiers. Journal of Machine Learning Research 2, 313–334 (2002)
7. Xia, Z., Dong, Y., Xing, G.: Support Vector Machines For Collaborative Filtering. In: 44th Annual Southeast Regional Conference, Melbourne, pp. 169–174 (2006)
8. Chung, Y., Lee, S.W., Lee, J.H.: Personalized Expert-based Recommender System. In: Korean Institute of Intelligent Systems Fall Conference 2012, Seoul, pp. 147–148 (2012)

A Comparative Study on Mobile Visual Recognition

Elisavet Chatzilari[1,2], Georgios Liaros[1,3],
Spiros Nikolopoulos[1], and Yiannis Kompatsiaris[1]

[1] Information Technologies Institute,
Centre for Research and Technology Hellas, Thessaloniki, Greece
[2] Centre for Vision, Speech and Signal Processing University of Surrey Guildford, UK
[3] Dept. of Informatics, Ionian University, 49100, Kerkyra, Greece
{ehatzi,geoliaros,nikolopo,ikom}@iti.gr

Abstract. In this work we perform an extensive comparative study of approaches for mobile visual recognition by simultaneously evaluating the performance and the computational cost of state-of-the-art key-point detection, feature extraction and encoding algorithms. Every step is independently tested so that its contribution to the final computational cost can be measured. The widely used OpenCV library is utilized for the implementation of the algorithms, while the evaluation is performed on the PASCAL VOC 2007 dataset, a challenging real world dataset crawled from the web. Our study identifies the algorithmic configurations that manage to optimally balance performance and computational cost, and provide a viable solution for real time mobile visual recognition.

Keywords: image classification, feature extraction, mobile visual recognition, OpenCV.

1 Introduction

In the 90s, the second generation (2G) cellular technology limited the functionalities of mobile phones to the very basics, i.e. making calls and sending text messages (SMS). Rapidly, mobile networks and devices began to evolve to higher speed networks (GPRS and WAP) and smaller devices with new functionalities (MMS and emails). In the last decade, the third generation (3G) was launched which in turn gave its place to 4G in 2009. In parallel with the developments in network capabilities, mobile devices evolved to smartphones, typically equipped with more processing power, rich sensors like GPS receivers and quality cameras. This fact opened up a whole new range of possibilities and radically new services like mobile visual recognition, a field of intense research for the last years. Mobile visual recognition refers to the process of taking a photo with the mobile phone camera and, after processing its visual content, presenting relevant information back to the user. A typical example is when a photo of a landmark is processed to return textual information (e.g. a wikipedia article) or related images (e.g. different views of the same landmark).

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 442–457, 2013.

The basic motivation for using mobile image recognition is the ability of visual content to transfer rich semantic content that is either too complicated or too ambiguous to be expressed with words. Indeed, if the user is not sure how to describe something with words it may be easier to search with a picture. Moreover, it is also possible to use image recognition services with embedded optical character recognition (OCR) capabilities for translating foreign billboards and road signs. Finally, the ability of mobile image recognition to turn the world around us into semantic links through a mobile phone camera, e.g. in the form of augmented reality metatags pointing to news, websites, or special offers (e.g. Layar, Wikitude) is what makes this service way more attractive than text or voice-based search that are more demanding from the perspective of user input.

Although research in visual recognition has made great progress during the last decade achieving relatively high performance figures [11],[24], the situation is different when aiming for practical applications where real-time processing is a critical factor. According to [4], 40% of the users will abandon a mobile application if they have to wait more than three seconds, and if that time rises to 10 seconds, 60% will not use it for a second time. For a visual recognition application, this formulates the typical trade off between the accuracy of the utilized algorithms and their computational cost, which is even more critical in the mobile environment where the resources are limited. In this context, the majority of existing mobile visual recognition applications employ a client/server architecture (Fig. 1). Cell phones act as clients that capture the object of interest and send queries to the server where the actual processing part takes place (i.e. analyze the image, identify its content, retrieve relevant information from the data pool and send it back to the client). However, with the fast paced evolution of mobile phones and the increased processing power, it becomes more and more interesting to investigate whether the entire processing load can be allocated solely on the smartphone, and in this way avoid the shortcomings of the client/server architecture like bandwidth limitations, data transfer latencies and service dependence on the existence of WiFi or 3G network availability. Moreover, another appealing property of this architecture is its potential to scale to an unlimited number of users since the algorithms run entirely on the phone and information is stored locally, removing the possibility of overloading the centralized server by numerous requests.

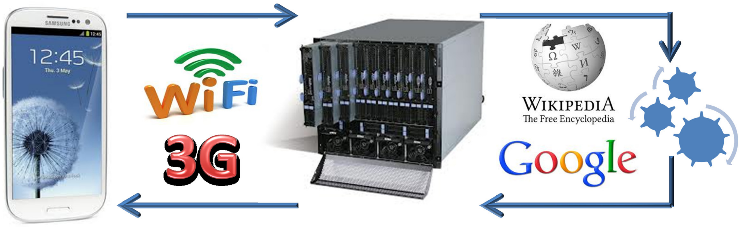


Fig. 1. Client/server architecture

In this work, we attempt to balance the computational cost and the performance of a visual recognition system so as to select a configuration that is able to run in an acceptable time frame and, at the same time, can provide satisfactory results for the specific application. We only consider the case where the entire process runs on the smartphone, and additionally look for a solution that can be executed close to real time. The goal is to identify a configuration that performs at less than one second per image in order to be appealing to the end user. In addition, considering the space and memory limitations of a mobile device environment, the typical content-based image retrieval scheme (i.e. find relevant images with the query image captured by the phone) would not be able to scale to large databases. For this reason, the objective of the proposed system falls under a classification scheme (i.e. analyze a captured image and provide keywords that describe its content), that only requires the storage of a machine learning model in the local database for each concept. Possible applications of such a scheme include the automatic annotation of mobile photo collections for keyword-based retrieval, personalized photo sharing and object recognition and annotation in real time for relevant information provision (e.g. detecting a bus and combining location information, the routes timetables can be presented).

We select one of the most popular image recognition pipelines that consists mainly of two parts, the image representation and the machine learning-classification component. For the classification component, being a field of high research interest for many decades now, many advances have been achieved, resulting in very effective and efficient algorithms [32]. It usually consists of the training step, which is the computationally expensive one, and the testing step. In a mobile recognition setting, the training step is done offline and only the testing step is performed online. Considering that the testing step can be rather fast even for very expensive algorithms, we have decided to employ Support Vector Machines (SVMs) [8], which is a state-of-the-art algorithm that can be very fast at the testing step for linear and additive kernels, while demonstrating exceptional generalization ability. On the other hand, the image representation algorithms will need to be executed online for every captured image and thus need to be carefully selected in order to achieve maximum performance at minimum cost. Towards this direction, we compare different algorithmic configurations consisting of key-point detection, feature extraction and feature encoding, so as to identify the ones that combine speed and performance. Our contribution is on thoroughly studying the existing state-of-the-art algorithms for visual recognition in a mobile environment and providing the necessary information for deciding how to optimally balance the aforementioned trade off based on the requirements of the application at hand.

The rest of this manuscript is organized as follows. In Section 2, related works are presented. In Section 3, an overview of the evaluated system architecture is described, while the two components of the system, image representation and classification, are analytically discussed in the next two Sections 4 and 5, respectively. The experimental results are shown in Section 6, while Section 7 concludes our work and discusses our plans for future work.

2 Related Work

Visual recognition has been a field of intense research interest for the past decades. This has increased even more with the recent changes in software, hardware and network technologies, i.e. digital cameras, Web 2.0 applications, smartphones, computers with high processing power, etc. This technological evolution has resulted into huge amount of multimedia content and has brought up the need for automatic visual recognition. The Query-by-example is the first appearance of image retrieval attempts that are based on visual content. ALIPR (Automatic Photo Tagging and Visual Image Search) [17] is one of the first attempts by researchers to incorporate visual content in search engines. In a similar direction, the authors of [25] present a framework for efficient large scale object retrieval that can potentially be applied on web-scale databases of billions images. Nowadays, Google search engine has employed an image retrieval service where the user can drag an image to the search box and it searches for similar images and sites with respect to its visual content in the Web.

On a mobile environment visual-based image search is even more useful and practical as it is way easier to capture an image to trigger the search than to type keywords. In general, in the visual recognition area the leap from academic research to commercial products has been much shorter and many companies have already incorporated visual recognition in mobile applications. For example, the service offered by Kooaba (www.kooaba.com) receives a snapped image as query and displays related information, further links and available files, applied to wine lists, printed catalogues, etc. oMoby (www.omoby.com) offers a shopping service that helps users find information about products by snapping a photo, such as links to retailers offering product information, reviews, prices, and more. Point & Find (pointandfind.nokia.com) is a service offered by Nokia that uses visual search technology to let users find more information about the surrounding objects, places, etc., in real time. Google Goggles (www.google.com/mobile/goggles/) is a mobile application that lets users search the web using pictures taken from their mobile phones.

On a research level, a few approaches have been presented that attempt to exploit the benefits of visual recognition in a mobile environment. The authors of [29] propose an augmented reality system, that recognizes landscapes within a database which is cached locally in the mobile phone, and presents related links to the user. They utilize a combined visual and location (GPS) based retrieval scheme and the whole system runs exclusively on the phone. In [18], a mobile application that uses visual recognition in order to find relevant documents in a locally hosted database is proposed. The authors of [33] propose lighter mobile versions of two popular feature descriptors so that the adopted natural scene tracking system can perform at frame rates of up to 20Hz.

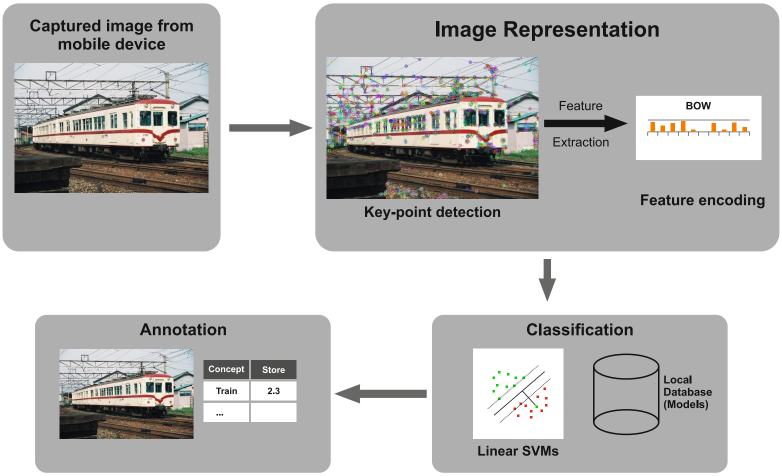
In this work we attempt to examine the visual recognition pipeline in a mobile environment by simultaneously evaluating the image representation algorithms with respect to performance and computational cost. Our work can be considered more closely related to [13], which evaluates a retrieval scheme and focuses on the various architectures that can be employed in a mobile visual search framework.

On the other hand, we select to implement the architecture which dictates that all the processing takes place on the phone and focus our experiments on comparing the various algorithmic configurations in an effort to balance performance and computational cost.

3 Framework Overview

A visual recognition framework can be considered to consist of two parts, the image representation and the classification part. In the last decade, the most popular approach that has been employed for image representation is referred to as bag of visual words (BOW) [9]. A typical pipeline for this approach is shown in Figure 2. For each image, points of interest are detected and for every key-point a set of features are extracted so as to be encoded into a single vector using the BOW approach. In order to train the vocabulary of visual words, key-point detection and feature extraction is applied on a large database of images and the features are clustered into visual words. Each component is analyzed in more detail in Section 4.

Finally, after representing each image with a single feature vector, a model that learns the correspondence between image labels and features needs to be trained. A very popular algorithm that learns to map features to concepts is the SVMs [8], which aims at splitting the input feature space so that the different classes are separated. For every concept, the images are either labelled as positive or negative if they depict or not the concept respectively. The algorithm attempts to split the feature space by a hyperplane that maximizes the margin between the positive and the negative side. A new image is then classified as positive or negative depending on the placement of its feature vector according to the hyperplane.



Concept	Store
Train	2.3
...	

Fig. 2. Framework overview

Table 1. Description of key-point detection algorithms (*** stars represent the fastest and * star the slowest)

Algorithm	Speed	Translation invariance	Description	Ref.
Dense	***	✗	Key-points are selected on a grid using a step of a fixed number of pixels.	-
FAST	***	✗	Key-points are selected based on the intensities of the pixels around the examined pixel	[26]
GFTT	**	✓	Pixels are ranked based on a quality measure that is extracted using eigen analysis	[28]
HARRIS	**	✗	Detects high intensity variations of sliding windows around the pixel and defines a corner based on a score	[14]
MSER	*	✓	Detects connected regions with little change across several intensity thresholdings of the image	[21]
SIFT	**	✓	Locations at minima and maxima of a Difference of of Gaussian function are selected as key-points	[20]
STAR	*	✓	Detects the extrema of the Gaussian operator's Laplacian across scale	[1]
SURF	**	✓	Uses integral images and box filtering techniques	[3]
ORB	**	✗	Augments the FAST detector with a pyramid scheme and the Harris corner measure	[27]

4 Image Representation

4.1 Key-Point Detection

Key-point detection refers to the detection of well defined positions in an image that can robustly characterize its conent. They are usually points of high interest that are used to represent important aspects of the image. Visual recognition requires repeatable key-points under several transformations such as rotation, scale changes, translation and lighting variations, whilst maintaining the detection time to the minimum. Mikolaijczyk et al. [22] review several key-point detectors on a benchmark dataset. The key-point detection algorithms that were used for our experiments can be seen at the first column of Table 1. In the second column, the relative speed of each detector is indicated by stars (more/less stars means the detector is faster/slower). In the third column, the translation invariant detectors are checked and the non invariant are crossed.

4.2 Feature Extraction

The feature extraction component attempts to describe the surrounding environment of each key-point so that it captures characteristic and indicative information of its visual content. Each key-point that was previously detected is represented as a multidimensional feature vector (descriptor). Several state-of-the-art descriptors are evaluated in [30]. In this work the evaluated descriptors are shown in Table 2, along with information about the extraction time and whether they are rotation and/or scale invariant.

Table 2. Description of feature extraction algorithms (*** stars represent the fastest and * star the slowest)

Algorithm	Speed	Size	Rotation invariance	Scale invariance	Description	Ref.
BRIEF	***	32	✗	✗	Bit string description of an image patch, constructed by a set of binary tests	[6]
ORB	***	32	✓	✗	Steered version of BRIEF according to the orientation of key-points	[27]
SURF	**	128	✓	✓	Uses box filtering techniques and integral images	[3]
SIFT	*	128	✓	✓	A histogram of orientations is computed in a 16x16 area around the key-point	[19]

4.3 Feature Encoding

After applying the key-point detection and the feature description algorithms, each image is represented by a set of multidimensional vectors. In order to transform these vectors into a single vector representation, a feature encoding algorithm is applied. The most popular method, borrowed from text retrieval, is the bag of visual words. A vocabulary of visual words is constructed from a large independent training set of images by clustering the extracted descriptors in n clusters/visual words, using an algorithm such as k-means. Afterwards, the feature encoding algorithm is applied. An extensive empirical study on encoding methods can be found at [7]. We have implemented and tested three different algorithms for encoding:

Hard Assignment: The local feature descriptors of the image are matched with the visual words of the vocabulary. A histogram of the visual descriptors is populated by adding ones to the corresponding bins.

Soft (kernel codebook) Assignment [31]: In this case instead of assigning a descriptor to a single corresponding visual word we assign it to k bins in a soft manner. More specifically, for every descriptor we add a quantity q to the bins of the k top nearest visual words. This quantity q is the Gaussian kernel (Radial Basis Function) distance of the descriptor and the visual word.

Vector of Locally Aggregated Descriptors (VLAD) [15]: In this case, first each descriptor is assigned to its closest visual word. Then for each visual word a vector is calculated by accumulating all the differences of the assigned descriptors with the visual word. Finally, these vectors are concatenated into a single vector representation.

In all the aforementioned cases, for each descriptor the nearest visual words must be computed. The baseline method, brute force, searches for the nearest neighbours amongst all possibilities in a greedy fashion. However, as this can be computationally expensive, approximate indexing algorithms that are significantly faster are examined. One of the most popular libraries for approximate neighbour search is the Fast Approximate Nearest Neighbor Search library (FLANN) [23], which is optimized for use in high dimensional spaces.

5 Machine Learning and Classification

The second part of a typical visual recognition system is to train a classification model for every visual concept that will learn to assign unseen images to concepts. In order to train a classification model for a specific concept we need a set of positive and a set of negative images. Each image is described by a feature vector which is extracted by the representation framework described in the previous section. Then, an SVM model is trained on these data in order to learn the properties that define the examined concept. The models are trained using the one versus all (OVA) technique, i.e. all positive examples of the specific concept versus all negative examples. The model is practically the hyperplane that separates the space of the positive and negative samples. In the case of a linear kernel, it can be represented by a vector $\mathbf{w}$, i.e. the model parameters, and a bias scalar b. For a concept c and a test image I, which corresponds to a feature vector $\mathbf{f}_I$, a confidence score is extracted by computing its distance to the hyperplane of the model for the concept c:

$$confidence(I, c) = \mathbf{w}_c * \mathbf{f}_I + b_c \tag{1}$$

The distance of a vector from the hyperplane indicates our confidence that the examined image depicts the concept of interest. High positive values of this score increase our confidence that this image belongs to the positive class while high negative values provide strong confidence that the image does not depict the concept. The fact that each model can be represented by a single vector and a scalar and that the testing process is essentially a vector multiplication, renders SVMs the best solution for a real time classification framework. This allows for storing the information about all the models in the phone memory, while testing is computationally very efficient, making it possible for the image classification algorithm to run entirely on a mobile phone.

6 Experimental Evaluation

Our intention in the experimental evaluation is to perform an extensive survey of the state-of-the-art algorithms used in a typical pipeline for visual recognition. In section 6.1 the dataset and the implementation details are explained. The strategy adopted for our evaluation experiments is to examine how the performance of each configuration is connected to the required extraction time, aiming to simultaneously improve the recognition performance and computational cost of the most prominent configurations and finally identifying the one that combines these two aspects best (Sec. 6.2).

6.1 Dataset and Implementation Details

For performing our experiments we have selected the benchmarking dataset of the PASCAL VOC 2007 competition [10]. Ground truth for both the training and the testing set of the 2007 dataset were released, while the datasets for the next years' competition lack ground truth annotations for the testing set, which is the reason the dataset from 2007 was selected instead of the more recent ones. It consists of 9.963 images collected from flickr, half of which are used for training and half for evaluating the visual recognition setups. The dataset is annotated with twenty concepts in a multi label manner (person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, tv/monitor). The split that was provided with the dataset was used, i.e. divided into two equal subsets, one for training and one for testing so that equal numbers of positive examples for each concept exist in the two subsets. This dataset was used for the performance evaluation of all different settings while the computational time estimation was performed on an image with resolution 1024 x 768 was captured from the mobile phone.

OpenCV [5], one of the most popular libraries for computer vision that also provides an interface for android cell phones was used with default parameters for the detection and description algorithms. More specifically, OpenCV was used to implement all possible combinations between the key-point detection and feature extraction algorithms described in Section 4.1 and 4.2 resulting in 36 different configurations. The feature encoding algorithms described in 4.3 were subsequently implemented using the indexing algorithms provided by FLANN. K-means was used to cluster a set of descriptors into 2000 visual words for hard and soft assignment. For the VLAD encoding method, the number of the centers was chosen so that the final feature vector was 2000-dimensional (i.e. 16 centers were extracted for the 128-dimensional SIFT and SURF, and 64 for the 32-dimensional BRIEF and ORB). For the classification part the LIBLINEAR library was selected [12]. The computational costs were computed using a Samsung Galaxy S3 device, which features a quad-core processor clocked at 1,4Ghz and 1GB RAM. In order to evaluate the various pipelines we incorporate the widely used performance measure of mean Average Precision (mAP). Average precision takes into account both precision and recall, and measures the rank quality of the retrieved images. It is calculated using Eq. 2, where $P(k)$ is the precision at rank k and $rel(k)$ is an indicator function equaling 1 if the item at rank k is a relevant document and zero otherwise. mAP is the mean of the average precision over all concepts.

$$AP = \frac{\sum_{k=1}^{n} Pr(k) * rel(k)}{\#\text{ of relevant images}}, \tag{2}$$

where $rel(k)$ is an indicator function equaling 1 if the item at rank k is a relevant document and zero otherwise.

6.2 Classification Performance versus Computational Cost

In this section the classification performance with respect to the total computation cost of each configuration is reviewed. Initially, in Table 3, the cost for each pair of key-point detection - feature description using hard binning and brute force nearest neighbour search is shown side-by-side with the recognition performance. In order to continue experimenting only with the most prominent configurations in terms of balancing the performance-cost trade-off we apply the following selection strategy. For every feature extraction approach we choose the best performing key-point detection algorithm and dismiss the ones that are more computationally expensive. Similarly, for every feature extraction approach, we choose the fastest algorithm and dismiss the ones that perform worse. If we denote as $Time(k, f)$ the computational cost and $Perf(k, f)$ the performance of a configuration composed by key-point detection method k and feature extraction method f we choose to dismiss the settings (k, f) for which:

$$
\begin{aligned}
& \forall f_j = \{BRIEF, ORB, SIFT, SURF\} \\
& Time(k_i, f_j) > Time(\arg\max_{k_i} Perf(k_i, f_j)) \\
or \quad & Perf(k_i, f_j) < Perf(\arg\max_{k_i} Time(k_i, f_j))
\end{aligned}
$$

In this way we manage to discard cases that are computationally expensive without adding anything to performance. In the case of GFTT and HARRIS we can see in Table 3 that they need approximately the same time to compute while GFTT performs better in all cases. For this reason when choosing the fastest algorithm, we choose the GFTT even in the cases that it is not strictly the fastest (e.g. the configurations GFTT+SURF and HARRIS+SURF where although HARRIS+SURF is faster by 10 ms we select GFTT as faster). The pairs selected after applying the aforementioned selection strategy are written in bold in Table 3. Additionally, all settings are visualized in a 2D plot, in Figure 3, where the x axis represents the computational cost in logarithmic scale and the y axis the complement of the performance measure in mAP. The dismissed settings are depicted as red squares and the selected as black stars. It is obvious that the selection strategy we applied picked out the cases which have the best combination of performance and computational cost, i.e. in the area closer to the (0,0) origin.

For the remaining configurations we attempt to optimize both the computational cost and the performance. The first step is to replace the expensive brute force search for the nearest neighbour in the hard binning setting with its faster approximate version, FLANN. The results are shown in Table 4, from which we can see that there is a significant gain in computational time, especially in the DENSE, FAST and SURF configurations (shown in bold). This can be explained by the fact that the these detectors tend to extract more key-points than the rest, so the main processing load is in the encoding part of the algorithm. Moreover, there are only slight variations in performance with respect to the brute force

Table 3. Time versus Performance of hard binning using brute force. The configurations that are shown in bold are the most prominent ones and are selected for further experimentation.

	Time (ms)				**Performance (mAP)**			
	BRIEF	**ORB**	**SIFT**	**SURF**	**BRIEF**	**ORB**	**SIFT**	**SURF**
DENSE	**3130**	**2807**	**11546**	10847	**15,70%**	**15,17%**	**17,30%**	13,16%
FAST	3862	3095	**39055**	**13645**	13,26%	12,91%	**17,86%**	**17,59%**
GFTT	**555**	**564**	**2156**	**884**	**11,64%**	**11,39%**	**15,80%**	**13,27%**
HARRIS	603	873	2146	874	10,49%	9,90%	13,61%	10,72%
MSER	3322	3298	34645	3843	9,71%	8,87%	14,86%	12,40%
SIFT	631	605	3865	1400	10,04%	9,66%	11,49%	12,16%
STAR	1857	1712	12558	**2718**	9,72%	9,66%	14,96%	**13,68%**
SURF	2081	1845	56651	**5038**	11,51%	10,42%	16,92%	**15,75%**
ORB	761	913	5528	**1227**	11,06%	10,72%	13,00%	**14,15%**

method, which verifies that the approximate nearest neighbour algorithm works almost as good as in the brute force case. Considering that in other encoding methods we will need to search for even more nearest neighbours, the use of the approximate version is mandatory.

The next step is to increase the performance, and towards this goal we evaluate two popular substitutes to the hard encoding method, the soft assignment and the VLAD encoding algorithms. Tables 5 and 6 depict the results for each encoding algorithm. In the case of soft assignment, it is obvious that the

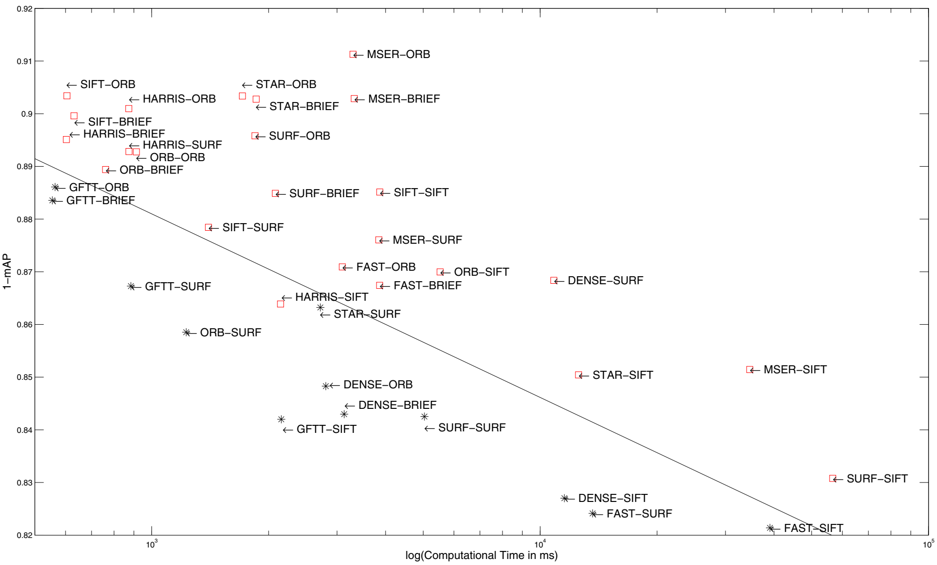


Fig. 3. Time versus Performance of hard binning using brute force

Table 4. Time versus Performance of hard binning using FLANN. The computational cost is significantly decreased with minor changes in performance. In bold the configurations where the decrease in maximum.

	Time (ms)				Performance (mAP)			
	BRIEF	**ORB**	**SIFT**	**SURF**	**BRIEF**	**ORB**	**SIFT**	**SURF**
DENSE	**1948**	**1349**	**3026**		15,51%	15,92%	18,02%	
FAST	**2506**	**1757**	**29767**	**4274**	13,26%	12,90%	18,96%	17,18%
GFTT	576	573	1863	560	11,62%	11,23%	16,21%	12,63%
STAR				2063				11,14%
SURF				**2279**				17,09%
ORB				1025				13,73%

performance is significantly improved in all cases with a relatively low loss in efficiency, especially in the cases where the number of detected key-points are limited (e.g. GFTT, STAR, ORB). Using the VLAD encoding method, the performance of all configurations increases, especially for the configurations with the SURF descriptor, where the gain is notably higher. The best performing configuration is the FAST-SURF-VLAD, which yields 31.35% mAP and requires 6.4 seconds to execute. However, we can see that for the minor performance loss of 0.25% we can use the SURF-SURF-VLAD combination in less than half the time of the previous scheme. For this reason, we select the SURF-SURF-VLAD scheme to continue with further enhancements.

An important note is that the time measurements were performed on a typical image taken by the cell phone which by default has a resolution of 1024 × 728. On the other hand, the performance measures were calculated on the PASCAL dataset which consists mainly of smaller images, on the average scale of 300×500 resolution. Thus, for the SURF-SURF-VLAD configuration, we only need to capture images at the scale of 300 × 500 in order to achieve the 31.1% mAP which decreases the time to only 0.5 seconds per image. This reduced time requirement renders the proposed configuration ideal for real-time mobile visual recognition. In order to achieve better performance, we can tweak the default values of OpenCV for the SURF detector. More specifically, if we change the Hessian threshold from 100 to 50 and 10, the performance increases to 31.72% and 32.6% mAP respectively, while the computational cost increases to 0.96 and 1.42 seconds respectively. Considering that our initial goal was to achieve a computational time no more than 1 second, we choose the configuration with the Hessian threshold set to 50, which yields 31.72% in 0.96 seconds. Finally, a few images with the proposed annotations from the selected configuration are shown in Table 7.

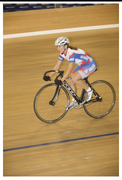
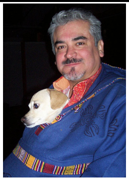
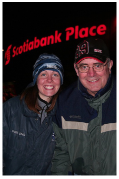
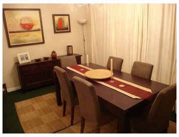
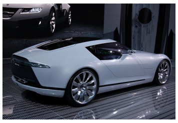
Table 5. Time versus Performance of soft binning using FLANN. The most appealing configuration is shown in bold.

	Time (ms)				Performance (mAP)			
	BRIEF	**ORB**	**SIFT**	**SURF**	**BRIEF**	**ORB**	**SIFT**	**SURF**
DENSE	3285	2684	4629		21,54%	21,62%	23,72%	
FAST	3990	3170	31278	5763	18,83%	19,01%	30,14%	26,01%
GFTT	602	613	1875	601	16,34%	15,85%	24,20%	18,58%
STAR				2134				15,06%
SURF				**2669**				**25,99%**
ORB				1050				19,49%

Table 6. Time versus Performance of VLAD encoding. The most appealing configuration is shown in bold.

	Time (ms)				Performance (mAP) (%)			
	BRIEF	**ORB**	**SIFT**	**SURF**	**BRIEF**	**ORB**	**SIFT**	**SURF**
DENSE	3381	2053	4949		20,53	20,42	27,98	
FAST			32213	6492			29,63	31,35
GFTT	526	543	1920	632	16,58	16,33	17,52	20,97
STAR				2186				17,15
SURF				**2954**				**31,10**
ORB				1019				20,84

Table 7. Images with proposed annotations by our framework. In bold the correct proposals. Some of the mistakes can be partly justified in the case of visually similar concepts, e.g. aeroplane-bird, person-statue.

	**bicycle, person**		**chair, tv-monitor**		**car**
	**person, dog**		person		aeroplane, **bird**
	**person**		**chair,** person, tv-monitor		**car**

7 Conclusions and Future Work

In this paper, we have reviewed state-of-the-art algorithms for image representation and compared their performance versus the computational cost when applied in a mobile environment for image classification. The performance was measured in a real world dataset originating from flickr, which renders the work of visual recognition a very challenging task. The results show that close to real time performance can be achieved with various configurations by sometimes sacrificing performance for the gain of speed. The case of SURF+SURF+VLAD seems the most appealing since for 300 × 500 resolution images it requires only 0.96 seconds to achieve 31.72% mAP.

Our plans for future work are mostly directed by our intention to increase the performance of the mobile recognition framework. More specifically, they include the testing of the FREAK descriptor [2] that was recently published and released by OpenCV, the evaluation of the promising super vector encoding method [34] and the enhancement of the performance by utilizing the popular spatial pyramids [16]. Finally, the exploitation of the visual recognition framework in real cases and the evaluation of the algorithm in more representative datasets for the use cases is in our intensions as well.

Acknowledgement. This work was supported by the EU 7th Framework Programme under grant number IST-FP7-288815 in project Live+Gov (www.liveandgov.eu)

References

1. Agrawal, M., Konolige, K., Blas, M.R.: Censure: Center surround extremas for realtime feature detection and matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
2. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: Fast retina keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21 (2012)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. 110(3), 346–359 (2008)
4. Berg, D.: Apple says: Mobile application performance matters, October 29 (2012), http://www.apmdigest.com/apple-says-mobile-application-performance-matters
5. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
6. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
7. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: British Machine Vision Conference (2011)
8. Cortes, C., Vapnik, V.: Support-vector networks. In: Machine Learning, pp. 273–297 (1995)

9. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) (2007) Results, `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012, VOC 2012 (2012) Results, `http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html`
12. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
13. Girod, B., Chandrasekhar, V., Chen, D.M., Cheung, N.-M., Grzeszczuk, R., Reznik, Y.A., Takacs, G., Tsai, S.S., Vedantham, R.: Mobile visual search. IEEE Signal Process. Mag. 28(4), 61–76 (2011)
14. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of Fourth Alvey Vision Conference, pp. 147–151 (1988)
15. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 3304–3311 (June 2010)
16. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition, pp. 2169–2178 (2006)
17. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. IEEE Trans. Pattern Anal. Mach. Intell. 30(6), 985–1002 (2008)
18. Liu, X., Hull, J.J., Graham, J., Moraleda, J., Bailloeul, T.: Mobile visual search, linking printed documents to digital media. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2010)
19. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, ICCV 1999, vol. 2, pp. 1150–1157. IEEE Computer Society, Washington, DC (1999)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
21. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Comput. 22(10), 761–767 (2004)
22. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
23. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Application, VISSAPP 2009, pp. 331–340. INSTICC Press (2009)
24. Over, P., Awad, G., Fiscus, J., Smeaton, A.F., Kraaij, W., Qunot, G.: TRECVID 2011 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In: Proceedings of TRECVID 2011. NIST, USA (December 2011)
25. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)
26. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)

27. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: International Conference on Computer Vision, Barcelona (2011)
28. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1994, pp. 593–600 (1994)
29. Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.-C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., Girod, B.: Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, pp. 427–434 (2008)
30. van de Sande, K., Gevers, T., Snoek, C.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 99(1) (2008)
31. van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., Geusebroek, J.M.: Visual word ambiguity. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(7), 1271–1283 (2010)
32. Vapnik, V.N.: Statistical learning theory, 1st edn. Wiley (1998)
33. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., Schmalstieg, D.: Pose tracking from natural features on mobile phones. In: Proceedings of the 7th International Symposium on Mixed and Augmented Reality (2008)
34. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)

Accuracy-Based Classification EM: Combining Clustering with Prediction

Stephanie Sapp and Armand Prieditis

Neustar Labs
Mountain View, CA 94041
armand.prieditis@neustar.biz

Abstract. Expectation-Maximization (EM) is well-known for its use in clustering. During operation, EM makes a "soft" assignment of each row to multiple clusters in proportion to the likelihood of each cluster. Classification EM (CEM) is a variant of EM that makes a "hard" assignment of each row to its most likely class. This paper presents a variant of CEM, which we call Accuracy-Based CEM (ABCEM), where the goal is prediction rather clustering. ABCEM first assigns each row to the most likely class based on the input columns, and then estimates performance of this assignment by evaluating the mean squared prediction error (MSPE) on the output columns, and proceeds as in CEM to update clusters and re-assign each row to the new clusters. Finally, the optimal clustering is selected to minimize the MSPE, selecting a local optimum from the left, and thus the procedure can also be viewed as a principled version of early stopping which uses only the training set. Our results show that ABCEM is nearly 40% more accurate than CEM.

Keywords: Expectation-Maximization, Clustering, Prediction, Mixture Models.

1 Introductxion and Motivation

Expectation-Maximization (EM) is well-known for its use in clustering (see [1] for example). During operation, EM makes a "soft" assignment of each row to multiple clusters in proportion to the likelihood of each cluster. It makes this assignment based on the current set of parameters. Once all rows are soft-assigned, EM updates the current set of parameters and the process repeats. EM is attractive because each update of the parameters is guaranteed to increase the likelihood of the data given the parameters. This means that EM is guaranteed to converge to a local maxima of the likelihood. To increase the probability of finding global maxima of the likelihood, EM is typically run with multiple initial parameters. Classification EM (CEM) is a variant of EM that makes a "hard" assignment of each row to the most likely class (see [2] for example). CEM not only has similar convergence guarantees, but it tends to run faster than EM because the likelihood function is typically much faster to compute than in the EM. For example, in a multivariate normal distribution, the EM function requires exponentials, but the CEM function does not.

In this paper, we explore the task of prediction rather than clustering. In a prediction task, each row includes an output variable; the task is to predict the output

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 458–465, 2013.

based on the remaining variables. CEM (or EM) can be used to find the parameters for one or more subclasses.

In particular, we evaluate a variant of CEM, which we call Accuracy-Based CEM (ABCEM), which operates as follows: like CEM, ABCEM assigns each row to the most likely class at each iteration, but unlike CEM, ABCEM evaluates the likelihood based on only the input columns, and selects the optimal clustering to minimize the squared difference between the actual output and the predicted output. The optimum is selected as the first local minima in the iteration sequence, and may thus be seen as a form of early stopping using only the training data.

The rest of this paper is organized as follows. Section 2 describes the inputs for the task and how it differs from traditional clustering approaches, but can yet make use of clustering algorithms such as EM and CEM.

2 Using Classification EM (CEM) Clustering for Prediction

In the classical method of clustering, one column of the training data can be viewed as missing. This column of missing data corresponds to a variable with one of *k* classes. The task of clustering is to find a "soft" or "hard" assignment of each row to each of the *k* classes. Once the parameters associated the *k* classes are determined, each row can then be clustered (grouped) into the *k* subclasses based on the parameters that result from the assignment. As we mentioned, in CEM the assignment is "hard" in that each row is assigned to only *one* of the *k* classes.

In a prediction task, the columns are partitioned into two sets of columns: input and output columns. The output columns are to be predicted from the input columns based on a model. Note that in traditional clustering, the output columns have no special significance. That is, all of the columns—both the input and output columns—are treated the same.

One way to use the results of traditional clustering at prediction time is to find the most likely cluster and then predict the output columns from the input columns given the cluster and a model that relates the output columns to the input columns. For example, assuming a multivariate normal distribution model of the columns, the output columns can be expressed as function of the input columns and the most likely cluster. Assume that observations x_i from a given class i are distributed multivariate normal with mean $\mathrm{\mu_i}$ and covariance matrix Σ_i; where x_i, $\mathrm{\mu_i}$, and Σ_i are comformably partitioned as:

$$x_i = \begin{bmatrix} x_{i_a} \\ x_{i_b} \end{bmatrix}, \mu_i = \begin{bmatrix} \mu_{i_a} \\ \mu_{i_b} \end{bmatrix}, \Sigma_\mathrm{i} = \begin{bmatrix} \Sigma_{i_a} & \Sigma_{\mathrm{i_c}} \\ \Sigma_{i_c}^T & \Sigma_{\mathrm{i_b}} \end{bmatrix} \tag{1}$$

Then the output columns *a* within class *i* can be predicted from the input columns *b* as follows:

$$\hat{\mu}_{i_a} = \mu_{i_a} + \Sigma_{\mathrm{i_c}}\Sigma_{\mathrm{i_b}}^{-1}(\mathrm{x_b} - \mathrm{\mu_{i_b}}) \tag{2}$$

That is, the predicted mean of the output columns (i.e., the most likely output columns values) is the mean of the outcome columns of class *i*, plus an adjustment based on the covariance between the input and output columns, the inverse of the covariance among the input columns, and the deviation between the input columns mean and the input column values x_b—all for the class *i.*

Computing the most likely cluster i based on the input column values x_b and the multivariate normal model is straightforward, found by simply maximizing the likelihood over the *i* classes:

$$\frac{1}{\sqrt{(\det(2\pi\Sigma_i))}} exp\left[-\frac{(x_b-\mu_i)^T\Sigma_{\mathrm{i}}^{-1}(x_b-\mu_i)}{2}\right] \tag{3}$$

Note that both the predicted mean and the most likely cluster involve computing an inversion, which is computationally expensive (between quadratic and cubic time, depending on the implementation). Computing the most likely cluster also involves computing a determinant, which is as computationally expensive as inversion.

A simpler alternative, which we adopt for purposes of producing results for this paper, is to assume that, given the cluster *i*, all columns are probabilistically independent from each other (i.e., the Naïve Bayes assumption). Under this assumption, the predicted mean output is simply the mean of the output columns of class *i*. That is, the second part of the equation, involving computing the correlation between the input and output columns, is assumed to be zero, due to $\Sigma_{\mathrm{i_c}}$ assumed to be a zero matrix. Computing the likelihood of *x* belonging to class *i* also simplifies to the following product over the *m* input columns:

$$\prod_{j=1}^{m}\frac{1}{\sigma_{i_j}\sqrt{2\pi}} exp\left[\frac{-1}{2}\left(\frac{x_j-\mu_{i_j}}{\sigma_{i_j}}\right)^2\right] \tag{4}$$

Since the task is to choose a class which maximizes the above expression, we can equivalently minimize over the negative of the log of this expression, resulting in the following simplified expression (constants common to each class are removed):

$$\sum_{j=1}^{m}\left\{\left(\frac{x_j-\mu_{i_j}}{\sigma_{i_j}}\right)^2+\log\left(\sigma_{i_j}\right)\right\} \tag{5}$$

Note that this expression does not involve any exponentials, determinants, or inversions. All that is required is to compute this expression over each class and choose the class that is associated with the lowest value. In short, this expression trades off between the weighted deviation from the mean and the standard deviation. That is, it tends to choose a class with input column means close to the observed input values (weighted by variance) and low standard deviation.

Of course it is possible to predict with other models, but this is one of the simplest models—one that offers a low time computational complexity. It is also possible to predict with "soft" assignments, such as with EM, by weighting the resulting output from each class based on the probability of each class given the inputs.

3 Clustering with CEM

Now that we have described how to predict an output once clusters have been formed, we need to describe in more detail how CEM operates during clustering to differentiate it from ABCEM. Recall that CEM makes a "hard" assignment during clustering: it chooses the most likely class for each row based on *all* the column values that are non-missing for that row. Under the Naïve Bayes assumption this amounts to computing for each row for each class the following expression (here *n* is the number of non-missing columns for a particular row being processed; note that *n* can vary from row to row):

$$\sum_{j=1}^{n}\left\{\left(\frac{x_j-\mu_{i_j}}{\sigma_{i_j}}\right)^2+\log\left(\sigma_{i_j}\right)\right\} \tag{6}$$

Once this "hard" assignment is made for each row, the parameters are updated based on the assignment and the process repeats until convergence of the overall likelihood of the data. Note that *all* the columns—input and output columns—are treated equally.

Under the Naïve Bayes assumption, missing columns are simply ignored because information from one column will not help to impute another column. The only proper inference under these conditions is to infer the mean value for a column, which essentially provides no additional information.

Note that the Naïve Bayes assumption is the same as the multivariate assumption assuming a diagonal covariance matrix and zeros on the off-diagonal entries. The Naïve Bayes assumption is also more relevant for big data (i.e., data that involves thousands of columns). In experiments, we found that more than 1000 columns created a severe computational bottleneck with inversion of a multivariate model. Thus we believe that the Naïve Bayes assumption is really the only practical assumption for data that is both tall (millions of rows) and wide (thousands of columns)

4 Clustering with ABCEM

The critical difference between ABCEM and CEM is to treat the input and output columns differently. ABCEM begins similarly to CEM: it makes a "hard" assignment by choosing the most likely class for each row. However, unlike CEM, ABCEM uses *only the input columns* to select the most likely cluster. Next, again like CEM, once the assignments for each row are made, parameters are updated based on the assignment. Like CEM, the process is iterated. However, we do not decide when to stop based on convergence criteria using the overall likelihood of the data. Instead, we select when to stop based on the mean squared prediction error (MSPE) of the current clustering result. That is, at each iteration, after assigning each row to a cluster, we predict the output column values for each row based on its assigned cluster, and use this value to evaluate prediction error. The process is stopped when the MSPE of the

9current iteration is higher than that of the previous iteration, and select the optimal clustering as the one obtained in the next-to-last iteration.

Note that ABCEM can also be viewed as a regularized form of CEM, using a principled and clever version of early stopping. Unlike standard early stopping procedures, ABCEM uses *only the training data*, and does not require splitting the training set to create a separate validation set. ABCEM is able avoid splitting the training set due to the fact that each clustering and parameter update step only use information from the input columns. Allowing the output columns to be used for valid performance evaluation.

In short, ABCEM takes advantage of the fact that the task is prediction of the output columns whereas CEM does not. The next few sections evaluate how much this buys ABCEM over CEM in terms of accuracy in predicting the output columns.

5 Experimental Results Comparing ABCEM with CEM

We ran both ABCEM and CEM on three real-world domains. The real-world domains are as follows:

- The UCI housing data (http://archive.ics.uci.edu/ml/datas/Housing). This 506 row data contains housing values in suburbs of Boston and the following attributes (the last column is the output column):
  1. **Crim**: per capita crime rate by town
  2. **Zn**: proportion of residential land zoned for lots over 25,000 sq.ft.
  3. **Indus**: proportion of non-retail business acres per town
  4. **Chas**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  5. **Nox**: nitric oxides concentration (parts per 10 million)
  6. **Rm**: average number of rooms per dwelling
  7. **Age**: proportion of owner-occupied units built prior to 1940
  8. **Dis**: weighted distances to five Boston employment centres
  9. **Rad**: index of accessibility to radial highways
  10. **Tax**: full-value property-tax rate per $10,000
  11. **Ptratio**: pupil-teacher ratio by town
  12. **B**: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  13. **Lstat**: % lower status of the population
  14. **Medv**: Median value of owner-occupied homes in $1000's.
- The UCI concrete compressive strength data set (http://archive.ics.uci.edu/ml/datas/Concrete+Compressive+Strength). This 1030 row data contains concrete compressive strength values and the following attributes (the last column is the output column):
  1. **Cement**: kg in an m3 mixture
  2. **Blast Furnace Slag**: kg in a m3 mixture
  3. **Fly Ash**: kg in an m3 mixture
  4. **Water**: kg in an m3 mixture
  5. **Superplasticizer**: kg in an m3 mixture
  6. **Coarse Aggregate**: kg in an m3 mixture

7. **Fine Aggregate**: kg in an m3 mixture
8. **Age**: Day (1-365)
9. **Concrete compressive strength**: MPa

- The UCI crime data (http://archive.ics.uci.edu/ml/datas/Communities+and+Crime). This 1994 row data contains U.S. community socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR and 128 attributes, a sample of which are shown below; the last column is the output column:
    - **Householdsize**: mean people per household
    - **Racepctblack**: percentage of population that is african american
    - **RacePctWhite**: percentage of population that is caucasian
    - **RacePctAsian**: percentage of population that is of asian heritage
    - **RacePctHisp**: percentage of population that is of hispanic heritage
    - **AgePct12t21**: percentage of population that is 12-21 in age
    - **AgePct12t29**: percentage of population that is 12-29 in age
    - **AgePct16t24**: percentage of population that is 16-24 in age
    - **AgePct65up**: percentage of population that is 65 and over in age
    - **NumbUrban**: number of people living in areas classified as urban
    - **PctUnemployed**: percentage of people 16 and over, in the labor force, and unemployed
    - **PctEmploy**: percentage of people 16 and over who are employed
    - **NumIlleg**: number of kids born to never married
    - **PctIlleg**: percentage of kids born to never married
    - **LemasSwornFT**: number of sworn full time police officers
    - **LemasSwFTPerPop**: sworn full time police officers per 100K population
    - **LemasPctPolicOnPatr**: percent of sworn full time police officers on patrol
    - **LemasGangUnitDeploy**: gang unit deployed (numeric - decimal - but really ordinal - 0 means NO, 1 means YES, 0.5 means Part Time)
    - **LemasPctOfficDrugUn**: percent of officers assigned to drug units
    - **PolicBudgPerPop**: police operating budget per population
    - **ViolentCrimesPerPop**: total number of violent crimes per 100K population

We chose these datasets mainly for simplicity because they contain only real-valued columns, and are possible to scale up to multivariate models.

For all datasets, we used 50 random restarts, 50 updates to the parameters for CEM, and maximally 50 updates to the parameters for ABCEM. We also varied the number of classes *k* from 2-6. To measure accuracy, we reserved 10% of the training data for testing. The tables below show the MSPE on the testing set for each method, as well as the percent improvement of ABCEM over CEM.

Table 1. Housing Data Accuracy

# Classes	CEM	ABCEM	% Impr.
2	72.6	55.6	23.4
3	71.2	68.7	3.5
4	59.4	57.5	3.1
5	61.2	38.6	36.9
6	72.0	39.2	45.5

Table 2. Concrete Compressive Strength Accuracy

# Classes	CEM	ABCEM	% Impr.
2	256.7	188.6	26.5
3	258.4	196.8	23.8
4	230.5	213.0	7.6
5	209.9	195.6	6.8
6	215.3	147.9	31.3

Table 3. Crime Data Accuracy

# Classes	CEM	ABCEM	% Impr.
2	0.061	0.055	8.9
3	0.060	0.045	24.7
4	0.057	0.050	12.9
5	0.057	0.035	37.5
6	0.043	0.039	7.6

6 Summary and Conclusions

The Housing Data showed the most improvement for ABCEM over CEM: a whopping 45% better than CEM for 6 classes. It's unclear how many classes are optimal for this (or any other) dataset; the point of these experiments was not to determine an optimal number of classes, but to get some idea of how well ABCEM performs in comparison to CEM. For example, the Housing Data probably shows that over-fitting starts after 5 classes. The Housing Data also showed a 3% improvement of ABCEM over CEM with 4 classes, but we suspect this result was simply from a bad initial choice of parameters for ABCEM. We expect that additional random restarts should result in even more improvement of ABCEM over CEM. Overall, the average MSPE of CEM over the cluster sizes 2 to 6 classes was 67, while the average

MSPE of ABCEM over the cluster sizes 2 to 6 was 52, leading to ABCEM improving the MSPE over CEM by an average of 23%.

The Concrete Strength Data showed a maximal 31% improvement of ABCEM over CEM at 6 classes. The average MSPE of CEM over the cluster sizes 2 to 6 was 234, while the average MSPE of ABCEM over the cluster sizes 2 to 6 was 188. On average, ABCEM resulted in a 20% improvement in performance over CEM.

Similarly, the Crime Data showed a maximal 38% improvement of ABCEM over CEM at 5 classes. The average MSPE of CEM over the cluster sizes 2 to 6 was 0.055, while the average MSPE of ABCEM over the cluster sizes 2 to 6 was 0.045. ABCEM resulted in a 19% average improvement in performance over CEM.

In short, ABCEM resulted in substantially lower MSPE at nearly all cluster sizes from 2-6, as well as close to 20% average improvement over CEM across all the cluster sizes. We believe that ABCEM is the method of choice for combining clustering with prediction.

7 Future Directions

Note that, in ABCEM the *creation* of clusters is only based on input features, while the *evaluation* of clusters is only based on output columns. Since evaluation of clustering is only based on accuracy, different clustering procedures can be plugged in. In practice, this means that we do not need to restrict ourselves to a single clustering procedure when say, building a hierarchy of classes and subclasses. Instead, we can use a variety of clustering procedures at each node in the hierarchy, and simply chose the result that produces the lowest error for that split.

We plan on exploring recently developed methods for improving the choice of initial parameters [3]. Although these methods refer to k-means clustering, we believe they will be applicable to ABCEM as well.

References

1. Moore, A.: Very Fast EM-based Mixture Model Clustering Using Multiresolution kd-trees. In: Advances in Neural Information Processing Systems. Morgan Kaufman (1999)
2. Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Computational Statistics and Data Analysis 14(3), 315–332 (1992)
3. Bahmani, B., et al.: Scalable K-Means. ArXiv 1203.6402v1 (2012), http://arxiv.org/pdf/1203.6402v1.pdf

When Classification becomes a Problem: Using Branch-and-Bound to Improve Classification Efficiency

Armand Prieditis and Moontae Lee

Neustar Labs
Mountain View, CA 94041
armand.prieditis@neustar.biz

Abstract. In a typical machine learning classification task there are two phases: *training* and *prediction*. This paper focuses on improving the efficiency of the prediction phase. When the number of classes is low, linear search among the classes is an efficient way to find the most likely class. However, when the number of classes is high, linear search is inefficient. For example, some applications such as geolocation or time-based classification might require millions of subclasses to fit the data. Specifically, this paper describes a branch-and-bound method to search for the most likely class where the training examples can be partitioned into thousands of subclasses. To get some idea of the performance of branch-and-bound classification, we generated a synthetic set of random trees comprising billions of classes and evaluated branch-and-bound classification. Our results show that branch-and-bound classification is effective when the number of classes is large. Specifically, branch-and-bound improves search efficiency logarithmically.

Keywords: Branch-and-bound, Bayesian Models, Classification.

1 Introduction and Motivation

In a typical machine learning classification task there are two phases: *training* and *prediction*. When the number of classes is low, linear search among the classes is an efficient way to find the most likely class. For example, consider a Bayesian Classifier, "where the task is to find the most likely class given an input example. More specifically, the task is to find a class *c* given input *x* such that *P(c|x)* is maximized. Using Bayes Theorem and removing the normalizing constant, the task is to find a class *c* such that *P(x|c)P(c)* is maximized. When the number of classes is low, linear search among the classes is an efficient way to find the most likely class. However, when the number of classes is high, linear search is inefficient. This paper focuses on improving the efficiency of the prediction phase when the number of classes is high (i.e. when the training data can be partitioned into thousands of subclasses) by using branch-and-bound. This paper presents preliminary results in artificial domains where we can directly control the number of subclasses. We are currently running experiments in several natural domains. The motivation for this

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 466–480, 2013.

work is threefold. First, some applications might require millions of subclasses to fit the data. In particular, those applications involving geo-location and time can have an enormous number of subclasses. Note that the subclasses do not have to be explicitly labeled in the data—they can be discovered in the training data with methods such as Expectation-Maximization. Such methods are beyond the scope of this paper. Second, fast prediction time is important in domains where a real-time response in needed. For example, a customer at a website might not be willing to wait more than a second to receive a targeted advertising whose selection is based on a prediction. Finally and in general, the more subclasses the higher the accuracy. Of course, too many subclasses can result in overfitting.

The rest of the paper is organized as follows. Section 2 presents our approach to branch-and-bound classification. Section 3 describes how to derive a lower-bound for branch-and-bound classification. Section 4 describes how to use the lower-bound function for pruning in branch-and-bound classification. Section 5 describes the particular learning method we used to build trees. Section 6 presents results of applying our approach. Section 7 describes a non-probabilistic approach to branch-and-bound pruning. Section 8 discusses related work. Finally, Section 9 discusses the conclusions of this work and outlines several promising directions for future work.

2 Branch and Bound in Tree-Structured Classification

Simple models in machine learning are typically computationally efficient. For example, a probabilistic model where each of the attributes is independent of each other conditioned on the class is easy to compute. However, such a model, which is called the "Naïve Bayes" model, is sometimes inaccurate. If the training data could be grouped (i.e., clustered) into subclasses, the results are likely to be more accurate. Of course the number of subclasses depends on the desired fit of the model to the training data: too many subclasses and the model can overfit the training data, resulting in a low training error rate but a high test error rate; too few and model has both a high training error rate and a high test error rate.

The classical method of finding subclasses is to assume that the data includes a column with *k* values, which correspond to each of the *k* subclasses and are missing for every row in the training data. (A *row* in the training data corresponds to an example pattern and a *column* corresponds to a feature of the rows.) The learning task is to find the probability of each of the *k* subclasses for each row in the training data. Once the parameters associated with each subclass are determined, each row can then be "clustered" into one of the *k* subclasses based on the parameters. Many different methods can be used to fit those subclasses including the k-means algorithm [1] and the EM algorithm [2].

In order to exploit branch-and-bound's efficiency, the classes described in this paper are assumed to be organized as a tree, which represents the training data at various levels of generality. For purposes of this paper, how the tree is built is unimportant; what is important is that there exists a method to find a most likely leaf node at a particular level. However, for reproducibility, we will describe the particular tree-building method that we used later.

The most important aspect of the tree is that there exists a way to evaluate degree of fit for an input test case and node. At prediction time, branch-and-bound finds a node in the tree that it believes to be the *most likely node* in the tree. Other methods are possible to determine degree of fit; we choose likelihood because our particular tree is a probabilistic one. Branch-and-bound then returns the prediction associated with that node. In general likelihood (or degree of fit) is correlated with the accuracy of the prediction generated from a test case. Indeed, this is one of the cornerstone assumptions in most machine learning: that nearness of inputs correlates to nearness of outputs. Note that if the learning was not structured as a tree, branch-and-bound would not make sense. For example, if the *k* subclasses were not in a hierarchy, the pruning as described here would not make sense.

To understand how each leaf node in the tree corresponds to a subclass, consider a binary tree. For example, in a binary tree, the left child of the root might have subclass label “Root/Left” and the right child “Root/Right.” Assuming a tree with two levels, there are four subclasses, which correspond to the each of the leaf nodes and which can be identified by the following labels: “Root/Left/Left,” “Root/Left/Right,” “Root/Right/Left,” and “Root/Right/Right.”

Note that the indexing scheme does not have to explicitly name each node using such hierarchical names. For example, in a binary tree, with root node having index 0, a node having index *i* can have as a left child the index $2i+1$ and as a right child the index $2i+2$. For ease of retrieval, we choose this method of numbering the tree’s nodes. Other number schemes are possible.

We mentioned that each node is associated with a set of training examples. When we say “associated” we do not necessarily mean that *any* training examples are stored at a node. Instead, each node contains information that is *summarizes* the training examples at that node. Specifically, the nodes in the tree are probabilistic and each node is associated with:

- The probability $p(t|s)$ of a node *t* given the parent node *s*: the number of training examples at the node relative to the number of training examples at the parent of the node.
- A mean vector μ: the average value of each column at the node.
- A variance vector σ^2: the variance of each column at the node. Note that the variance typically decreases along a path
- A min vector: the minimum value of each column at the node. Note that the min value is non-decreasing along a path.
- A max vector: the minimum value of each column at the node. Note that the max value is non-increasing along a path.

For simplicity, we used a Naïve Bayes model to compute likelihood. Under the Naïve Bayes model, the columns are assumed to be conditionally independent from each other given the subclass. Other models are possible such as a multivariate normal; we chose the Naïve Bayes model because it has a prediction-time complexity that is linear in the number of columns.

Using the Naïve Bayes assumption and assuming a Gaussian distribution, the likelihood at a node for a particular *j* length vector *x* is defined as:

$$p(x;\mu,\sigma^2) = \prod_{i=1}^{j} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2\ \right\} \tag{1}$$

More specifically, to find the most likely node in the tree we need to find that node which maximizes the path probability to the node multiplied by the probability of the node given the input (the *p* function). The path probability is the product of the probability of the child given its parent, for each child node in the path. Note that the path probability to a node is the same as the probability of the subclass associated with node.

Since we are interested in finding the most likely node, the *p* function can be simplified by taking the log and removing constants that are the same for any node:

$$nlp(x;\mu,\sigma^2) = \sum_{i=1}^{j} \left\{\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2 + \log(\sigma_i)\right\} \tag{2}$$

The term “*nlp*” means negative of the log of the likelihood. This form is simpler to compute because it involves a sum rather than a product. Using the *nlp* function, the task of branch-and-bound is to find a leaf node in the tree that minimizes the *nlp* function plus the sum of the path costs, which are derived from the negative of the log of the likelihood of each node given the parent node, over all the ancestor nodes in the path to the node. Once the most likely node is found, its most likely output value is returned. Here again, a variety of methods exist to produce the most likely output. Under the Naïve Bayes assumption and a Gaussian distribution, the most likely output at a node is the node’s mean output, which is simply that part of the mean vector μ that pertains to the output.

Note that the *nlp* function says that there’s a tradeoff at getting close to the mean of a leaf node (as weighted by the inverse of the standard deviation) and finding a leaf node with a low standard deviation.

3 Deriving a Lower-Bound

A lower-bound function is important in branch-and-bound algorithms because they guarantee never to prune a potential least cost solution. It’s possible to derive several different lower-bounds for the *nlp* function; here we describe a few that we derived. For a more general theory of how to derive lower-bound functions in search see [3]. One simple lower-bound is based on bounding boxes for each column. For example, if the value *x* for a particular column is outside of the bounding box for of the *min* and the *max* for that particular column, we know that the distance from *x* to the closest of *min* or *max* for that particular column is a lower-bound on the distance to the *mean* for that column. Therefore, we can substitute the closest of min or max for the mean in that case. If *x* is within the bounding box of *min* and *max* then return 0 for that particular column.

Another lower-bound is the minimum of the sum of the negative of the log of the *p(t|s)*, where *s* is the parent and *t* from a node to any leaf node. For some applications, the combination of these lower-bounds might be tight. However, for the bounding-box style of a lower-bound, when *x* is within the bounding box of *min* and *max*, no pruning occurs (at least none based on that particular column).

A better lower-bound is based on the *nlp* function itself. Note that when we say "lower bound," we mean that the function is a lower bound on the value returned by the most likely leaf node below that node. Note that a lower-bound on $\log(\sigma_i)$ is easy to derive simply by adding 1 to the standard deviation of each node. Therefore, a lower-bound on $\log(\sigma_i)$ is 0. That leaves the accuracy burden of the lower-bound *l* on the following expression:

$$l(x;\mu,\sigma^2) = \sum_{i=1}^{j}\left\{\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2\right\} \tag{3}$$

One way to guarantee that this expression produces a lower-bound is to ensure that the standard deviation along a path does not increase for each particular column. How reasonable is this assumption? Suppose that a node is associated with the following data set: {-5,-1,1,5}. The standard deviation at that node is 4.16333. Suppose further that we split this data set into two subsets. Taking into account mirror images, there are exactly three potential splits:

Child		**Standard Deviation for Each Child**	
1	**2**	**1**	**2**
{-5,-1}	{1,5}	2.82843	2.82843
{-5,1}	{-1,5}	4.24264	4.24264
{-5,5}	{-1,1}	7.07107	1.41421

In at least two of these splits at least one child has a standard deviation that *increases* from the root node. In particular, the second split results in *two* children whose standard deviation increases for both children. For the third split, the standard deviation increases for one child and decreases for another. For the first split, the standard deviation decreases for both children. Intuitively, this split groups the negative values together and the positive values together, which makes sense from a clustering point of view.

It may be that any reasonable machine learning algorithm will attempt to reduce the "disorder" at a node by choosing those splits that reduce the standard deviation for a column [4]. Therefore, any reasonable machine learning algorithm would prefer the first split, which does reduce the standard deviation. However, with multiple columns it may be that one column's standard deviation goes up while another column's standard deviation goes down. So, it is *not* reasonable to assume that the standard deviation will be reduced along a path for each particular column because the machine learning algorithm might attempt to "average" out the reduction of "disorder" among *all* the columns.

We believe that any reasonable machine learning algorithm will split a node into a set of child nodes (while building the tree) such that the function (3) is reduced from the node to any child. We haven't proven this yet, but we believe that *any* node can be split into a set of nodes that is guaranteed to reduce this function from the node to any child. For want of a better name, let us call this function the "disorder" function. Intuitively, this is what a machine learning algorithm is supposed to do: reduce the "disorder" for the purposes of classification. We also believe that even if a reasonable machine learning algorithm can't always reduce the "disorder," it will come close. More specifically, the above expression can be viewed as the *average* "disorder" at a node. (Technically, the average is simply the above expression divided by *j*. Since *j* is a constant for all nodes, it doesn't matter if we divide by *j* or not.)

4 Using the Lower-Bound Function for Pruning

The general branch-and-bound strategy is well-known; here we describe how to apply it in the context of prediction. More specifically, a lower-bound function can be plugged in standard branch-and-bound algorithms for effectively searching the tree. There are dozens of different branch-and-bound algorithms, developed as early as the 1960's [5]. The objective of each is to compute the function *f(s)* below:

$$f(s) = \begin{cases} t(s) & if\ T(s) \\ \min\{c(s,t) + f(t) | t \in d(s)\} & otherwise \end{cases} \tag{4}$$

Here, *s* corresponds to a node (or state), *t(s)* is a value for a leaf node *s*, *T(s)* is true if s is a leaf node; false otherwise, *c(s,t)* is the cost of reaching node *t* from node s, and *d(s)* returns the set of child nodes for node *s*. Note that the arc cost function *c(s,t)* is the negative of the log of the *p(t|s)*, where *s* is the parent and *t* is the child.

The basic idea behind a branch-and-bound algorithm is that if a *lower-bound l(s)* for a node *s* in the tree exceeds an *upper-bound u(t)* for some other node *t*, then the first node can be pruned. Typically, this is accomplished by recording the minimum upper-bound seen among all the paths explored so far, where a path corresponds to sequence of states generated by the child function *d(s)*. For example, one can obtain an initial upper-bound for a node by randomly searching the tree below the node and returning the cost of the path (i.e., the sum of the *c(s,t)* values along the path) plus the leaf node value. When the upper bound is the same as the lower bound, the procedure halts; at that point the upper-bound = lower-bound = *f(s)*.

However, this method is inefficient as it will evaluate nodes whose path cost + lower-bound estimate exceeds *f(s)*. That is, this type of search can go "deeper" than the leaf node associated with *f(s)*. A more efficient method is to evaluate only those nodes whose path cost + lower-bound estimate does *not* exceed *f(s)*. One way to compute the *f(s)* function with this idea is the classic A* algorithm [6]. However, this algorithm requires exponential space. Instead, we chose the IDA* algorithm [6], a well-known search algorithm, because it only requires linear space. This algorithm trades off memory for computation by repeating a series of ever deeper searches until it finds a solution. At any depth of search, it prunes off all those nodes whose cost so far + underestimated cost exceeds the current bound.

The IDA* algorithm is shown below. As input it takes in an index or pointer to a node.

Algorithm 1.

```
def f(s:node):
bound = 0
newbound = search(s,bound)
while (newbound > bound):
  bound = newbound
  newbound = g(s,bound)
return newbound

def g(s:node,bound:real):
if T(s) then return t(s)
else
  if l(s) > bound then return l(s)
  else return min{g(t,bound-c(s,t))+c(s,t)|t in d(s)}
```

The IDA* algorithm comprises two loops. The outer loop (embodied by the "f" function), incrementally increases the bound until a solution is found within the bound. The inner loop does the heavy lifting: it prunes all nodes whose cost so far plus estimate exceeds the bound. Once the most likely leaf node is found, its most likely output value (and node) can be returned. For simplicity, the algorithm shown above only shows return output value and not the node.

5 The Particular Method that We Used to Build the Trees

We mentioned that the particular method we used to build the tree is unimportant for purposes of branch-and-bound classification. For reproducibility, here we describe the particular method that we used to build the tree. More generally, the training procedure that builds the tree should be able to split the training data into *k* subclasses using any standard clustering method. Specifically, we used an EM-like procedure assuming two subclasses per node. Of course the number of subclasses depends on the desired fit of the model to the training data: too many subclasses and the model can overfit the training data, resulting in a low training error rate but a high test error rate; too few and model has both a high training error rate and a high test error rate. Two subclasses is the smallest number of subclasses with which a tree can be built. Moreover, a binary tree can "simulate" multiple subclasses by repeated division of each node into finer and finer subclasses.

The EM-like procedure that we used to build the tree assumes that the data includes a column with *k* values, which correspond to each of the *k* subclasses.

This column's data is missing for every row in the testing data. We built the tree by, at each split, finding *k* mean vectors, *k* variance vectors, and *k* subclass probabilities such that the probability of the data given the parameters is maximized under the assumed probability model. In our case, we used *k=2* but *k* can be any integer.

Once these parameters (mean, variance, and subclass probabilities) are determined, each row can then be "clustered" into one of the *k* subclasses based on the parameters and the process can be recursively repeated on each of the subclasses. By "recursively repeated," we mean that each row is assigned to one of the *k* subclasses and the process repeats recursively for each of the rows associated with each of the *k* subclasses.

Given a set of rows in a data set and a set of *k* subclasses, each of which is characterized by such a probability distribution function and a prior probability (i.e., frequency), the task is to find an assignment of each row to a single class.

This task is accomplished by finding the subclass that minimizes the *nlp* function plus the negative of the log of the subclass. This functional form tells us that there is a tradeoff between finding a subclass whose mean is close to the row (as weighted by the inverse of the standard deviation), finding a subclass with a low standard deviation (due to the log of the standard deviation term), and finding a subclass with high path probability. We used a "hard" assignment tree-building procedure: once such a subclass is found, the assignment of the row to the subclass is "hard" in that each example is assigned to only one subclass, the most likely one. In contrast, in traditional EM, each row has a probability of belonging to each class.

Other tree-building procedures exist in machine learning. For example, decision trees build a structure by splitting on attribute values or ranges. However, such resulting structures may not be amenable to pruning.

6 Summary of Results with Random Trees

With only a handful of subclasses, linear search for the most likely subclass will outperform branch-and-bound because there is little to structure into a tree. Thus the branch-and-bound algorithm is not meant for small data sets that yield only a few subclasses. Therefore, to get some idea of how well classification works, it is essential to test it on large trees, ones stemming from billions of training examples. Unfortunately data to build such trees is not easily available nor is it readily shareable.

As a result, we built a random set of data selected from up to 60 billion mixtures (each mixture corresponds to a subclass) and then ran our algorithm and compared it to the results for linear search.

The trees we used for the experiments here all have a uniform branching factor of *k,* where the *k* sibling nodes represent a split of the data into *k* subclasses. Each node of this tree is associated with a set of training examples and the children of a node represent a partition of those training examples from the parent. That is, the training examples at the parent node correspond to the union of those at the child nodes. In general, the trees do not have to have a uniform branching factor but it made experiment generation simpler.

We actually built the tree on the fly in such a way that the random tree was implicitly generated depth-first and exactly re-created each time the tree was traversed. Note that our comparison is not one in terms of accuracy because exactly the same answer will be returned both branch-and-bound and linear search. We are only interested in the number of nodes visited as the yardstick for comparison for both algorithms.

First, we implicitly create every path cost between a parent and a child node (i.e., negative log probability of child node given parent node) based on the parent index by the following process. Assuming the branching factor is denoted by ***B***, the system

1) Determines the random seed based on the parent node index and unique prime number given for path cost generation.
2) Samples B real values uniformly in [0, 1] and multiplies the irrational Knuth coefficient to avoid getting periodic fractions.
3) Takes only the fractional parts of the real values and normalizes them so that the sum will be 1.

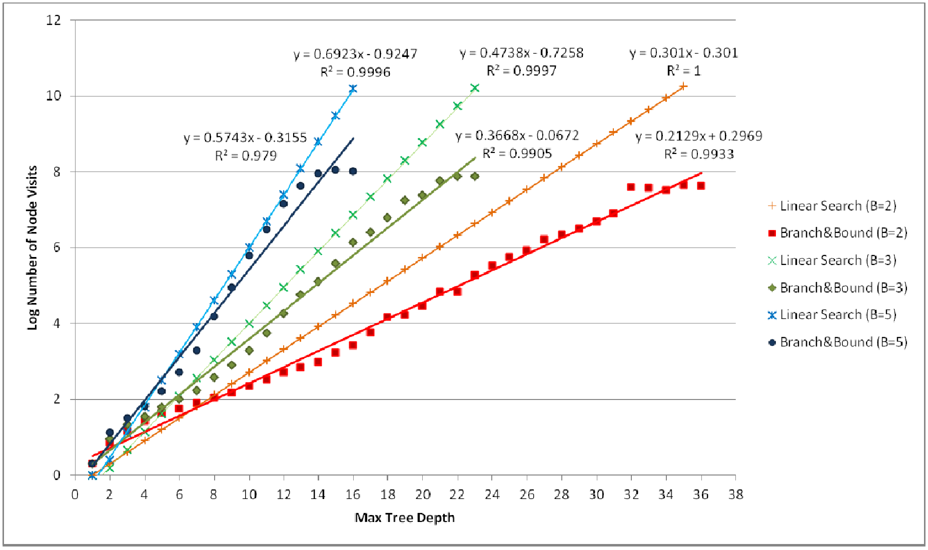


Fig. 1. Tree Depth vs. Log Number of Nodes Visited for Three Trees

As a random seed deterministically generates the pseudo-random numbers, we can always recreate any path cost in the tree based only on the parent index *without explicitly storing all of the nodes in the memory*. This is an important point for testing prediction algorithms on extremely large training data: we have developed a method that does not require storing the training data. Instead, we generate the tree on the fly and recreate it as needed for purposes of prediction. Note that if we initialize the unique prime number given for path cost generation to a different value, it guarantees that a different random tree will result without loss of generality.

Second, we implicitly generate the range of each column of each node given the column range of its parent node by the following process. The system:

1) Determines the random seed based on the parent node index and unique prime number for division generation.

2) Uniformly samples b-1 points between parent's min-max range. Clearly these b-1 points can be again recreated as they are sampled deterministically based on the seed that we generated on 1).

3) If the current node is ith child of parent node, the ith subdivision will have its own min-max range.

Though we could not generate a min-max range for each node in one shot, we can create it by following the label path starting from the root. For example, in the binary tree, the label path such as "LEFT/RIGHT/RIGHT/LEFT" can uniquely indicate the range of given node in an efficient manner.

Finally, the system generates the mean and variance vectors for each node given its range as follows:

1) Determine the random seed for mean and variance based on the current node index and unique prime numbers for mean and variance respectively.
2) Uniformly sample one point between its min-max range based on the mean seed.
3) Compute a simple variance by averaging the sum of squared distances between min and sampled mean, and the max and sampled mean.
4) Add a small normalized Gaussian noise based on the variance seed.
5) Add 1 to the variance computed from 4)

Here adding 1 in the step 5) prevents variance from being closer to 0, which causes a numerical problem.

Put together, these steps enable the creation of implicit random trees that simulate billions of mixture distributions, each with their own mixture proportions, min-max ranges, mean vectors, and variance vectors as a function of the node index and a random seed. It is important to understand that is it not possible to explicitly store all of these parameters for billions of nodes. We believe that such implicit random trees are a good vehicle to evaluate machine learning algorithms on enormous training sets with millions of subclasses.

Figure 1 shows the results of these experiments. As the figure shows, Branch-and-Bound search significantly beats linear search. In particular for binary trees, it shows nearly two orders of magnitude speedup. In fact, as depth *x* increases the speedup is $e^{1.37x}$ for a binary tree: this means the deeper the tree, the more effective the Branch-and-Bound. This means that with Branch-and-Bound search it is possible to do prediction with trees that are roughly 1.37 times deeper than for exhaustive (linear) search. This puts certain prediction problems within reach that were not possible before. Note that the comparison to linear search is *not* a strawman: this is the technique that is typically used in prediction and that fails with a large number of subclasses.

7 Branch-and-Bound Classification for Non-probabilistic Models

Probabilistic models are attractive because they can automatically weight and normalize each attribute based on the variance. One non-probabilistic approach that we consider here is k-means clustering [1], a popular clustering method. In k-means clustering, a set of k initial means is generated, one per subclass. Each row is assigned to the subclass whose sum of distances squared to each attribute is minimized. Once the assignments are complete, the means are reset based on the assignments of rows to classes and the procedure repeats. Many different methods can be used to terminate this procedure, but one popular method is to terminate when the sum of the distances squared does not significantly change between iterations.

Note that k-means can be applied hierarchically: each node can again be split into k subclasses. This hierarchy is essential for the type of branch-and-bound pruning described here because it produces a tree. Note also that k-means does not suffer from an exponential branching factor as the number of attributes increases. This is because k-means combines multiple attributes by considering their sum.

To understand how to prune with k-means trees, consider that along any path the extrema (min and max) of an attribute tightens: the min increases and the max decreases. This means that the sum of the distances squared to the nearest extrema will decrease along a path. And that means that this sum can be used for pruning as a lower-bound in branch-and-bound algorithms such as IDA*. The downside of k-means trees is that they treat each attribute as being equal (i.e., the distance measure is the same for each attribute). The attributes can be normalized by dividing each attribute's deviation from the mean by the variance (just as in the *nlp* function). However, this normalization does not solve the problem of equal weighting.

Note that clustering and classification are two different tasks: clustering involves grouping and classification involves prediction. However, clustering algorithms can be used to find subclasses for classification tasks.

8 Related Work

The Nearest Neighbor (NN) method is one of the oldest machine learning methods [8]. Instead of learning a classifier through training process, this method finds the nearest training example to an input query and then returns an output associated with that example. With a large number of training examples, this method often outperforms other machine learning methods such as neural nets, Bayesian nets, and decision trees, especially when many outputs are not strongly determined by one mathematical function of given input while they have weak correlations.

One straightforward but inefficient way to find the nearest training example is through *linear* search: compare the input to *every* training example and evaluate the output based on the nearest. The average time complexity of the linear approach is *n/2* for *n* training examples. Linear search is *instantly* incremental because no processing is required to add a new training example. However this approach repeats the entire search process again for each input query, becoming highly inefficient as *n* grows large.

The training examples can be organized into an approximately balanced tree called a *kd*-tree for faster prediction through branch-and-bound methods [9]. More specifically, the tree can be built in *O(nlog n)* time through a linear time median-finding algorithm. For example, one way is to sort the data on one dimension and then split the data at the median for that dimension. The process can then be repeated on each of the two parts. The fundamental concept in all of these ways is to split the data approximately in half for a particular dimension and then repeat the process on the two parts for the next dimension. Once all the dimensions have been split, this process repeats again from the first dimension. For *k* dimensions, this yields a tree with branching factor 2^k. For example, for two dimensions, the data is cut into two sets for the first dimension and each of those sets into two more sets, thus yielding a quad tree.

Assuming that $n = B^d$, where *B* is the average branching factor and *d* is the depth of the search for an input query can be completed in $O(B^{d/k})$ time, where k is a constant greater than 1. Although this method is not instantly incremental, standard tree rebalancing methods attempt to keep the tree balanced with adding new training examples efficiently (i.e., in *O(log n)* time).

One shortcoming of the *kd* tree is that it does not scale up as the number of input dimensions grows: the tree's branching factor is 2^k, where *k* is the number of input dimensions (i.e., number of features or columns). This is because the *kd* trees are created by splitting on one dimension at a time. Therefore, the search complexity for a nearest neighbor in high-dimensional spaces can be worse than linear search.

Once the *kd* tree has been built by whatever method, various branch-and-bound methods can be applied to find the nearest neighbor. These methods use lower-bounds that can be derived from bounding hyper-rectangles associated with each node. That is, the distance from a particular input to the bounding hyper-rectangle is a lower-bound on the distance to any training example within the hyper-rectangle. This property is true for any node in the tree. For example, in two dimensions, there are nine cases for input location relative to a bounding rectangle, each having their own distance formula: within the rectangle (lower-bound is zero in that case), directly below, directly above, directly right, directly left, above-left, above-right, below-right, or below-left. The distance to this rectangle from an input is a lower-bound on the distance to any training example within the rectangle. This lower-bound property can be exploited by various branch-and-bound methods to dramatically reduce the search space for a nearest neighbor by pruning off branches of the tree that are guaranteed to be farther away than a given reasonable upper-bound.

Moore describes an EM-based variant of the standard *kd* tree-building procedure [10]. This variant uses bounding boxes to determine how deep to build the tree—that is, it can prune off parts of the tree *at training time*, thus helping to control over-fitting at prediction time. Note that the system described by Moore is focused on tree-building rather than prediction with a particular input, which is the focus of this paper. For example the EM-based variant described by Moore does not search the tree for a particular input as described here.

Other tree-building procedures exist in machine learning. For example, decision trees build a structure by splitting on attribute values or ranges. For continuous values, the procedure is remarkably similar to that of building kd-trees: choose a dimension, determine a splitting point, and then split the data. However, not all of such trees are amendable to the branch-and-bound approach because it is not easy to derive lower-bounds.

The key insight with *kd*-trees and other learned structures is that more time spent during training to build easily searchable structures can pay off in less time at prediction. The purpose of this paper is to explore the application of those insights in a probabilistic context. Part of the question that drove this research is whether or not these methods are applicable in a probabilistic context and under what conditions?

Even without a *kd*-tree, nearest neighbor (NN) methods suffer from other shortcomings. First, they do not generalize a classifier from training examples. That is, NN methods are likely to overfit the data if the training set has a relatively close example to an input query. In contrast, NN has no way to predict if none of training examples are closer or multiple training examples are exist in neighborhood close to a query.

Intuitively, adding a simple training process such as averaging multiple nearest neighbors might improve accuracy. However, it is difficult to choose an appropriate k, the number of training examples for prediction and to define the 'nearness' appropriately to balance across multiple dimensions. (Note that when $k=n$, this method returns the mean of all training examples as the output.)

Second and for the same reasons, nearest neighbor methods are sensitive to outliers. For example, a single nearest example from the training set could be an outlier, but the *k*-nearest neighbors might suggest a different output, smoothed away from that outlier. The problem is that it is hard to determine a proper *k* and ideally *k* should be varied for each prediction. Third, the standard distance metric in nearest neighbor weighs each input dimension equally, thus potentially exaggerating the importance of some insignificant dimensions. Locally weighted linear regression can help [11] in the sense that it generalizes based on relatively important examples, but still determining the neighborhood of locality is difficult and not feasible without structurally differentiating the weight of each dimension from the beginning. Finally, nearest neighbor methods cannot easily handle missing data, which frequently happens in real world data.

The probabilistic method in this paper addresses these shortcomings in a statistically sound framework.

9 Conclusions and Future Work

This paper described a new method for efficient classification when the number of classes is large. We showed its application in the context of a probabilistic model. The results show that the deeper the tree, the more efficient this algorithm gets as compared to linear search. With Branch-and-Bound search it is possible to do prediction with trees that are roughly 1.37 times deeper than for linear search.

This puts certain prediction problems within reach that were not possible before. We believe algorithms such as classification might form the cornerstone of handling enormous data sets, which result in an enormous number of subclasses.

We are currently exploring several promising avenues of future work. First, it is possible to apply lazy over-fitting control in the context of trees as described here. That is, it is possible to apply over-fitting control methods *after* training rather than during training. For example, the most likely leaf node might not be the most likely node in the entire tree. While exhaustively searching for the most likely node in the tree is not feasible, it might be possible to search a local neighborhood of the path leading to the most likely leaf node. The idea is that it might be more likely to find that node off the path than anywhere else. We are currently evaluating that hypothesis. It may also be that the most likely node at a lower depth is more likely than one at a deeper depth. This suggests the following iterative method of overfitting control: find the most likely node at level 1, continue iterating by level, replacing the most likely node if the likelihood is higher than the one being replaced, stop when the likelihood starts to decrease (this suggest overfitting).

Second, such as sampling might be useful to search through the resulting tree instead of pruning. We are currently evaluating such methods.

Third, it may be possible to derive better heuristics so that we can get a better lower-bound on the log of the standard deviation. It may also be possible to get more accurate heuristics through learning or caching searches in the tree.

Finally, the artificial dataset on which we ran our experiments could be biased towards our strategy. To rule that out, we are currently running experiments on two natural domains: the Reuters-RCV1 text classification dataset [12] and the Gene Ontology dataset [13], both of which are large and with tree-structured classes.

References

1. Moore, A.: K-Means and Hierchical Clustering (2001), tutorial at `http://www.autonlab.org/tutorials/kmeans11.pdf`
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1–38 (1977)
3. Prieditis, A.E.: Machine discovery of effective admissible heuristics. Machine Learning 12(1-3), 117–141 (1993)
4. Nowozin, S.: Improved information gain estimates for decision tree induction. arXiv preprint arXiv:1206.4620 (2012)
5. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. Econometrica 28(3), 497–520 (1960)
6. Nilsson, N.: Principles of Artificial Intelligence. Tioga Publishing Company (1980)
7. Korf, R.: Depth-first Iterative-Deepening: An Optimal Admissible Tree Search. Artificial Intelligence 27, 97–109 (1985)
8. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann (2006)
9. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM 18(9), 509–517 (1975)

10. Moore, A.: Very Fast EM-based Mixture Model Clustering Using Multiresolution kd-trees. In: Advances in Neural Information Processing Systems. Morgan Kaufmann (1999)
11. Moore, A., Schneider, J., et al.: Efficient Locally Weighted Polynomial Regression Predictions. In: Fourteenth International Conference on Machine Learning. Morgan Kaufmann (1997)
12. Lewis, D.D., Yang, Y., Rose, T., Li, F.: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5, 361–397 (2004), http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf
13. Gene Ontology dataset, http://www.geneontology.org/

Lazy Overfitting Control

Armand Prieditis and Stephanie Sapp

Neustar Labs
Mountain View, CA 94041
armand.prieditis@neustar.biz

Abstract. A machine learning model is said *overfit* the training data relative to a simpler model if the first model is more accurate on the training data but less accurate on the test data. Overfitting control—selecting an appropriate complexity fit—is a central problem in machine learning. Previous overfitting control methods include *penalty methods*, which penalize a model for complexity, *cross-validation* methods, which experimentally determine when overfitting occurs on the training data relative to the test data, and ensemble methods, which reduce overfitting risk by combining multiple models. These methods are all *eager* in that they attempt to control overfitting at training time, and they all attempt to improve the *average* accuracy, as computed over the test data. This paper presents an overfitting control method which is *lazy*—it attempts to control overfitting at prediction time for *each* test case. Our results suggest that lazy methods perform well because they exploit the particulars of each test case at prediction time rather than averaging over all possible test cases at training time.

Keywords: Overfitting control, Bayesian Models.

1 Introduction and Motivation

A machine learning model is said *overfit* the training data relative to a simpler model if the first model is more accurate on the training data but less accurate on the test data. Intuitively, the more complex model is fitting itself to what could be called *noise* in the training data instead of to what could be called the *signal* in the test data. The core problem is determining which parts of the data to ignore. Overfitting control—selecting an appropriate complexity fit to maximize the signal and minimize the noise—is a central problem in machine learning. For example, determining an appropriate number of hidden units in a neural net is overfitting control. Too few hidden units and accuracy on both the training and test data is low; too many hidden units and accuracy on the training data is high while accuracy on the test data is low. In short, the number of hidden units determines the number of parameters with which the data is fit and too many parameters is not necessarily good when it comes to accuracy on the test data.

More generally, Figure 1 shows the error on the training data (i.e., the lower line, the blue line) decreasing as the model's complexity increases. The figure also shows the error on the test data (i.e., the upper line, the red line) decreasing until the model's

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 481–491, 2013.

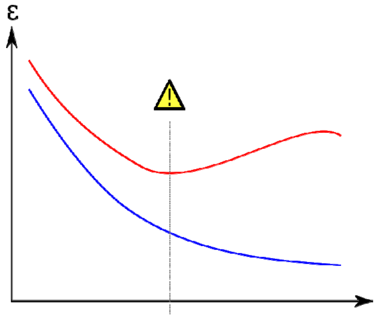


Fig. 1. Error on the Training Data (Blue; lower curve) vs the Test Data (Red; upper curve) as Model Complexity Increases

complexity is at the grey line denoted by the exclamation point; after that point, the error rate on the test data goes up. The model at this point can be said to *overfit* the data.

Previous overfitting control methods include penalty methods [1], cross-validation methods [2], and ensemble methods [3]. Penalty methods choose a model based both on the model's accuracy and its likelihood. In these methods, a model's likelihood inversely varies with its complexity.

For example, a model with features A=true and B=true is less probable than a model with only A=true because the model with A=true and B=true contains fewer hypotheses. Typically, penalty methods attempt to estimate the testing error with a linear combination of the training error and the penalty. The objective is to find a model that minimizes this linear combination. Occam's razor (i.e., prefer the simpler model) can work well as a way to estimate the error on the test data. However, there is no reason to believe that a linear combination of the training error and the penalty (or any other function) is a good functional form.

Penalty methods for decision trees include pessimistic pruning [4] and MDL pruning [5]. Penalty methods for neural networks include weight decay [6], weight elimination [7], and pruning methods [8,9]. Overfitting in neural nets is typically associated with saturated weights; hence overfitting control in neural nets often involves making the weights as small as possible and gradually increasing them until overfitting occurs. Unfortunately, these methods often ignore those combinations of weights that result in overfitting. Penalty methods for Support Vector Machines include margin maximization [10]. All of these methods bias the search towards simpler models.

Cross-validation methods partition the training data into two sets. The first set of data is used to build the model and the second set of data is used to compute the error. The second set of data serves as a proxy for the test data. The objective is to minimize the error on the second set of data while training on the first set. Multiple random partitions are then used to combine multiple models or average the error estimates. Cross-validation methods are guaranteed to perform within a constant factor of any penalty method [11]. Cross-validation methods have been applied to determine the learning rate, number of hidden units, kernel parameters, penalty size, and features. One shortcoming of cross-validation is that if the second set of data is too small, the error estimates will be noisy. Another shortcoming of cross-validation methods is that they can be computationally expensive as they involve multiple error estimates on the second set of data.

Ensemble methods don't choose a single model. Instead, they spread the risk of error on the test set over multiple models. For example, an ensemble method can involve learning multiple models, one for each set of random samples of the training set. The intuition is that many of the multiple models will fit the test data well and hence lower the error on the test set more than using a single model. Other ensemble methods include Bayesian model averaging (sampling hypotheses based on their posterior probability), bagging (iteratively removing high accuracy predictors and their corresponding data from the training set and repeating the process; the set of predictors corresponds to the model), randomized committees (train several hypotheses using different initial starting conditions), and random forests (growing multiple decision trees from randomly sampled feature subsets).

All of these previous overfitting control methods are *eager* in that they attempt to control overfitting at training time, and they all attempt to improve the *average* accuracy over the test data. This paper presents an overfitting control method which is *lazy* in that it attempts to control overfitting at *prediction time* for *each* test case.

The rest of the paper is organized as follows. Section 2 presents our approach to overfitting control. Section 3 describes the particular method we used for learning. Section 4 presents the specific tree searching method that we use. Section 5 presents the results of applying our approach. Section 6 describes related work. Finally, Section 7 discusses the conclusions of this work and outlines promising directions for future work.

2 Lazy Overfitting Control

Our approach to lazy overfitting control (LOC) assumes that it is given a particular learned structure in the form of a tree that represents the training data at various levels of generality. For purposes of this paper, how the tree is built is unimportant; what is important is that the tree has certain properties that we describe below. However, for reproducibility, we describe the particular tree-building method that we used in the next section and describe other tree building methods that are possible.

Property #1: the tree represents various levels of generality; a parent node in the tree is more general than a child node. By "generality" we mean that the parent of a node "covers" more training examples than a child and is hence "more general." We assume that each node in the tree comprises a set of parameters which summarize the data at that node (e.g., the mean and variance of each variable). The summary at each node forms the generalization hierarchy.

The tree has a uniform branching factor of *k,* where the *k* sibling nodes represent a split of the data into *k* subclasses. Each node of this tree is associated with a set of training examples and the children of a node represent a partition of those training examples from the parent. That is, the training examples at the parent node correspond to the union of those at the child nodes.

Property #2: there exists a way to evaluate degree of fit for an input test case and node. At prediction time, LOC finds a node in the tree that it believes to be the *most likely node* in the tree. Other methods are possible to determine degree of fit; we choose likelihood because our particular tree is a probabilistic one. LOC then returns the prediction associated with that node. Thus, LOC fits each test case at a flexible level of generality *at prediction time*.

Property #3: degree of fit on a test case is correlated with the accuracy of the prediction generated from the test case. As the degree of fit, LOC uses likelihood, which is correlated with accuracy: the more likely the node the higher the accuracy. At prediction time, the most likely node in the tree is the one that is nearest based on the weighting for the input columns. Note that because the tree summarizes the training data, a node "higher" up in the tree may be more likely than one "lower" down in tree. Intuitively, this aspect is what controls overfitting at prediction time. An eager method of overfitting control builds the tree to a fixed level at *training* time and uses the tree as is at prediction time. In contrast, LOC chooses the level at *prediction* time, which is what makes LOC lazy.

Property #4: the training data is grouped. Grouping the training data helps to smooth the predictions. For example, if the training data is grouped (i.e., clustered) into subclasses, the results are likely to be more accurate because a prediction can then be based on interpolating among multiple training examples.

Note that if the learning was not structured as a tree, it would be difficult to fit a test case at flexible levels of generality. For example, if the k subclasses were labeled in a flat, non-hierarchical way with labels such as "Class0," "Class1," and "Class2," it would be difficult to find sub-groupings of the classes helpful for lazy overfitting control. That is all possible subsets would need to be considered. While this is feasible with a small number of subclasses, it is clearly not feasible with hundreds of subclasses.

To understand how each leaf node in the LOC tree corresponds to a subclass, consider a binary tree. For example, in a binary tree the left child of the root might have subclass label "Root/Left" and the right child "Root/Right." Assuming a tree with two levels, there are four subclasses, which correspond to the each of the leaf nodes and which can be identified by the following labels: "Root/Left/Left," "Root/Left/Right," "Root/Right/Left," and "Root/Right/Right."

Note that the indexing scheme does not have to explicitly name each node using such hierarchical names. For example, in a binary tree, with root node having index 0, a node having index i can have as a left child the index $2i+1$ and as a right child the index $2i+2$. For ease of retrieval, we choose this method of numbering the tree's nodes. Other number schemes are possible.

We mentioned that each node is associated with a set of training examples. When we say "associated" we do not necessarily mean that *any* training examples are stored at a node. Instead, each node contains information that is *summarizes* the training examples at that node. Specifically, the nodes in the tree are probabilistic and each node is associated with:

- The probability *p(t|s)* of a node *t* given the parent node *s*: the number of training examples at the node relative to the number of training examples at the parent of the node.
- A mean vector μ: the average value of each column at the node.
- A variance vector σ^2: the variance of each column at the node.

For simplicity, we used a Naïve Bayes model to compute likelihood. Under the Naïve Bayes model, the columns are assumed to be conditionally independent from each other given the subclass. Other models are possible such as a multivariate normal; we chose the Naïve Bayes model because it has a prediction-time complexity that is linear in the number of columns.

Using the Naïve Bayes assumption and assuming a Gaussian distribution, the likelihood at a node for a particular *j* length vector *x* is defined as:

$$p(x;\mu,\sigma^2) = \prod_{i=1}^{j} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\{-\frac{1}{2}\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2\ \} \tag{1}$$

Many techniques can be used to determine the “best” matching node to a given input. Since our tree is a probabilistic one (i.e., each node is defined by a set of parameters which determines a probability distribution function), it makes sense to find the *most likely* node for given input based on the *p* function above on an input *x*.

More specifically, to find the most likely node in the tree we need to find that node which maximizes the path probability to the node multiplied by the probability of the node given the input (the *p* function). The path probability is the product of the probability of the child given its parent, for each child node in the path. Note that the path probability to a node is the same as the probability of the subclass associated with node.

Since LOC finds the most likely node, the *p* function can be simplified by taking the log and removing constants that are the same for any node:

$$nlp(x;\mu,\sigma^2) = \sum_{i=1}^{j}\left\{\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2 + \log(\sigma_i)\right\} \tag{2}$$

The term “*nlp*” means negative of the log of the likelihood. This form is simpler to compute because it involves a sum rather than a product. Using the *nlp* function, the task of LOC is to find a node in the tree that minimizes the *nlp* function plus the sum of the path costs, which are derived from the negative of the log of the likelihood of each node given the parent node, over all the ancestor nodes in the path to the node.

Once the most likely node is found, its most likely output value is returned. Here again, a variety of methods exist to produce the most likely output. Under the Naïve Bayes assumption and a Gaussian distribution, the most likely output at a node is the node’s mean output, which is simply that part of the mean vector μ that pertains to the output.

3 The Particular Method We Used to Build the Tree

We mentioned that the particular method we used to build the tree is unimportant for purposes of LOC. For reproducibility, here we describe the particular method that we used to build the tree. More generally, the training procedure that builds the tree should be able to split the training data into *k* subclasses using any standard clustering method. Specifically, we used an EM-like procedure assuming two subclasses per node. Of course the number of subclasses depends on the desired fit of the model to the training data: too many subclasses and the model can overfit the training data, resulting in a low training error rate but a high test error rate; too few and model has both a high training error rate and a high test error rate. Two subclasses is the smallest to build a tree, which in this case is a binary tree. Moreover, a binary tree can "simulate" multiple subclasses by repeated division of each node into finer and finer subclasses.

LOC sidesteps the issue of finding an appropriate number of subclasses by recursively building as many binary subclasses as possible at training time (i.e. down to leaf nodes containing singleton examples) and then selecting an appropriate node in the tree of subclasses at prediction time. That is, LOC builds a deep tree during training time and then selects a node at an appropriate depth at prediction time to avoid overfitting.

The EM-like procedure that we used to build the tree assumes that the data includes a column with *k* values, which correspond to each of the *k* subclasses. This column's data is missing for every row in the testing data. We built the tree by, at each split, finding *k* mean vectors, *k* variance vectors, and *k* subclass probabilities such that the probability of the data given the parameters is maximized under the assumed probability model. In our case, we used *k=2* but *k* can be any integer.

Once these parameters (mean, variance, and subclass probabilities) are determined, each row can then be "clustered" into one of the *k* subclasses based on the parameters and the process can be recursively repeated on each of the subclasses. By "recursively repeated," we mean that each row is assigned to one of the *k* subclasses and the process repeats recursively for each of the rows associated with each of the *k* subclasses.

Given a set of rows in a data set and a set of *k* subclasses, each of which is characterized by such a probability distribution function and a prior probability (i.e., frequency), the task is to find an assignment of each row to a single class.

This task is accomplished by finding the subclass that minimizes the *nlp* function plus the negative of the log of the subclass. This functional form tells us that there is a tradeoff between finding a subclass whose mean is close to the row (as weighted by the inverse of the standard deviation), finding a subclass with a low standard deviation (due to the log of the standard deviation term), and finding a subclass with high path probability. We used a "hard" assignment tree-building procedure: once such a subclass is found, the assignment of the row to the subclass is "hard" in that each example is assigned to only one subclass, the most likely one. In contrast, in traditional EM, each row has a probability of belonging to each class.

Other tree-building procedures exist in machine learning. For example, decision trees build a structure by splitting on attribute values or ranges. However, such resulting structures cannot be used for lazy overfitting control. The basic idea is to build a tree to a fixed level of overfitting and then determine the appropriate level for an individual test case *at prediction time.* We've chosen one type of tree—a probabilistic tree—but other trees can yield similar results with lazy overfitting control. For example, a kd-tree could work just as well as the trees we have built because the kd-tree has all of the Properties #1-#4: it has a measure: each node represents a level of generality, distance to a bounding hyper-rectangle is the degree of fit measure, and the training data is grouped by nearness.

4 Likelihood Stopping

Rather than search the entire tree for the most likely node, LOC stops searching when it decides that the likelihood will probably not increase. The procedure, which is run for each test case (i.e. unseen data that is to be predicted), comprises two iterative-deepening steps.

Step 1. Using the entire test set, iteratively increase the depth from 0 to d*+1 until the average likelihood, across all test cases, at the depth is worse than the previous depth. Here d* represents the depth associated with the best average likelihood. Store the best likelihood (and node) for each test case.

Step 2. For each individual test case, increase the depth past d* until the likelihood of the most likely node becomes worse than the previous depth's likelihood of the most likely node. Return the mean output value associated with the node having the greatest likelihood among all the depths.

5 Summary of Results

We were interested in comparing LOC to an optimal (though exhaustive and expensive) procedure: namely searching the entire tree to find the best node. Since it is not possible to do better than searching the entire tree for the best node, we wanted to see whether LOC, by searching a fraction of the nodes, could do nearly as well as searching the entire tree.

To get some idea of how LOC performs, we ran it on three domains, two real and one simulated. The two real ones involve predicting the geographic location of an IP address based on the round-trip transit time from a set of router locations on the internet. The geographic location (geo-location) is represented by three Cartesian coordinates (x, y, z). The procedure we used to infer the Cartesian coordinates involves making a prediction in terms of the x, y, z and then mapping it onto the surface of the earth. The first geo-location set comprises the roundtrip transit times from 100 routers on the internet in a 50K row training set and a 10K row test set (randomly selected). The second geo-location set comprises 82 router roundtrip transit times (plus the decimal of the IP address) in an 8731 row training set and a 2494 row test set.

The simulated data set comprises a 100K training set and a 10K test set with a single output variable and 24 input variables as follows:

x1 = Normal(0, 100)
x2 = Normal(100, 1)
x3 = Uniform(-1000, 1000)
x4 = x1^2 + x2
x5 = abs(x3)
x6 = t_5
x7 = t_20
x8 = x6^2 + x7^2
x9 = binomial(1000, 0.5)
x10 = binomial(20000, 0.3)
x11 = sqrt(x9)
x12 = sqrt(x10)
x13 = sqrt(x11)
x14 = log(x9 + 1)
x15 = log(x10 + 1)
x16 = log(x11 + 1)
x17 = log(abs(x1))
x18 = round(x1)
x19 = round(x6)
x20 = chisquare_1
x21 = chisquare_10
x22 = chisquare_100
x23 = log(x20)
x24 = sqrt(x22)

The output variable (the one LOC is predicting) is log(abs(x1 + x2 + ... + x24)).

Table 1 summarizes the results of these experiments.

Table 1. Results on Three Data Sets

Data	**Exhaustive**	**LOC**	**% Accuracy**	**% Time**
1	254	290	88	20
2	115	124	93	30
3	98	107	92	23

LOC enables us to obtain 88-92% of the accuracy of exhaustive search (where we find the most likely node in the entire tree) but at 20-30% of the time. In other words, by searching only a small fraction of the entire tree, LOC is able to obtain a result that is almost as good as what we would have found by searching the entire tree.

Since LOC controls overfitting differently for each new example, it also customizes the prediction depth for each new example. In the table below, we compare LOC to two less customized procedures: nearest neighbor (NN) and best average fixed depth. Best fixed depth uses an oracle procedure: selecting the depth with the lowest average prediction error among all test cases.

Table 2. LOC vs. Best Leaf node and Best Node at Fixed Depth

Test Set	NN	Best fixed depth	LOC
1	412	359	290
2	170	149	124

LOC performs significantly better than both of these methods.

6 Related Work

Nearest neighbor methods, one of the oldest machine learning methods [12], can be viewed as a lazy overfitting control method. Instead of learning a classifier through training process, this method finds the nearest training example to an input query and then returns an output associated with that example. Nearest neighbor is often called *lazy* machine learning because training time is zero and all of the complexity is pushed to prediction time in finding the nearest training example. With a large number of training examples, this method often outperforms other machine learning methods such as neural nets, Bayesian nets, and decision trees, especially when many outputs are not strongly determined by a single function.

Nearest neighbor methods suffer from several shortcomings. First, they do not generalize from training examples. As a result, nearest neighbor method are likely to overfit the training data. Smoothing the results by considering the *k* nearest neighbors can help improve accuracy, but it is difficult to choose an appropriate *k*, which ideally should be varied for each prediction. Second and for the same reasons, nearest neighbor methods are sensitive to outliers. Third, the standard distance metric in nearest neighbor weighs each input dimension equally, thus potentially exaggerating the importance of some insignificant dimensions. Locally weighted linear regression can help [13] in the sense that it generalizes based on relatively important examples, but determining the neighborhood of locality is still difficult and not feasible without a priori structurally differentiating the weight of each dimension. Finally, nearest neighbor methods cannot easily handle missing data, which frequently occur in real world data.

Note that the *k*-nearest neighbors method is not completely lazy because *k* is typically chosen at *training* time by cross-validation on the training set. This cross-validation method can require significant computational resources and carries some risk of overfitting on unseen data. [14] proposes an alternative that does not cross-validate at training time. Instead of choosing one neighborhood size, it averages the discriminants for multiple neighborhood sizes. They found that this approach worked well with *k*-nearest neighbors, support vector machines, hyperplane distance nearest-neighbor, and Bayesian quadratic discriminant analysis. They show that good classification performance can be attained without *any* training.

[15] considers the task of building decision trees at prediction time. Specifically, their motivation is similar to ours:

> *Building a single classifier that is good for all predictions may not take advantage of special characteristics of the given test instance that may give rise to an extremely short explanation tailored to the specific instance at hand (page 717)*

In other words, eager methods of controlling overfitting in decision trees can result in overly long "explanations," which translates to a higher error rate on the test set. Our results reinforce what Friedman, et al say.

7 Conclusions and Future Work

This paper described a new approach to overfitting control: lazy overfitting control (LOC), which pushes overfitting control to prediction time. Using LOC to customize prediction enables it to perform significantly better than fixed depth methods. We are currently exploring lazy overfitting control methods that can combine multiple hypotheses.

References

1. Borra, S., Di Ciaccio, A.: Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. Computational Statistics & Data Analysis 54(12), 2976–2989 (2010)
2. Ng, A.Y.: Preventing 'overfitting' of cross-validation data. In: Fourteenth International Conference on Machine Learning (Workshop), pp. 245–253. Morgan Kaufmann (1997)
3. Krogh, P.S.A.: Learning with ensembles: How over-fitting can be useful. In: Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference, vol. 8, p. 190. MIT Press (1996)
4. Esposito, F., Malerba, D., Semeraro, G., Kay, J.: A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5), 476–491 (1997)
5. Mehta, M., Rissanen, J., Agrawal, R.: MDL-based decision tree pruning. In: Proc. 1st Intl. Conf. Knowledge Discovery and Data Mining, KDD 1995, Montreal, Canada (1995)
6. Hinton, G.E., Van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory, pp. 5–13. ACM (1993)
7. Lawrence, S., Giles, C.L.: Overfitting and neural networks: Conjugate gradient and back-propagation. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, vol. 1, pp. 114–119. IEEE (2000)
8. Castellano, G., Fanelli, A.M., Pelillo, M.: An empirical comparison of node pruning methods for layered feed-forward neural networks. In: Proceedings of 1993 International Joint Conference on Neural Networks, IJCNN 1993, Nagoya, vol. 1, pp. 321–326. IEEE (1993)
9. Reed, R.: Pruning algorithms-a survey. IEEE Transactions on Neural Networks 4(5), 740–747 (1993)

10. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning 46(1), 131–159 (2002)
11. Kearns, M.: A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. Neural Computation 9(5), 1143–1161 (1997)
12. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann (2006)
13. Moore, A., Schneider, J., et al.: Efficient Locally Weighted Polynomial Regression Predictions. In: Fourteenth International Conference on Machine Learning. Morgan Kaufmann (1997)
14. Garcia, E.K., Feldman, S., Gupta, M.R., Srivastava, S.: Completely lazy learning. IEEE Transactions on Knowledge and Data Engineering 22(9), 1274–1285 (2010)
15. Friedman, J.H., Kohavi, R., Yun, Y.: Lazy decision trees. In: Proceedings of the National Conference on Artificial Intelligence, pp. 717–724 (1996)

Using Part of Speech N-Grams for Improving Automatic Speech Recognition of Polish

Aleksander Pohl[1,2] and Bartosz Ziółko[1]

[1] Department of Electronics
AGH University of Science and Technology
al. Mickiewicza 30, Kraków, Poland
www.dsp.agh.edu.pl
[2] Department of Computational Linguistics
Jagiellonian University
ul. Łojasiewicza 4, 30-348 Kraków, Poland
www.klk.uj.edu.pl
aleksander.pohl@uj.edu.pl, bziolko@agh.edu.pl

Abstract. This paper investigates the usefulness of a part of speech language model on the task of automatic speech recognition. The develped model uses part of speech tags as categories in a category-based language model. The constructed model is used to re-score the hypotheses generated by the HTK acoustic module. The probability of a given sequence of words is estimated using n-grams with Witten-Bell backoff.

The experiments presented in this paper were carried out for Polish. The best obtained results show that the part-of-speech-only language model trained on a 1-million manually tagged corpus reduces the word error rate by more than 10 percentage points.

1 Introduction

In the most of modern automatic speech recognition (ASR) systems the algorithm of operation is as follows: the input signal is recognized using an acoustic model (AM), i.e. a model that describes the relation between the sounds and the phonemes from the chosen alphabet. In that way a language message hypothesis is encoded using the sound. Since, typically that model is not analyzing all possible data, the result of the AM module is not a single result, but a list of hypotheses with probability estimations or a lattice. Then, a language model (LM), i.e. a model that describes the relations between the words is used to rescore the acoustic hypotheses in order to select the hypothesis that is the most plausible according to the LM and AM.

Since a LM utilizing all the various relations between the words is very hard to built, there are many approximations, word n-grams being the most popular [1]. In the word n-gram LM the relations between the words are narrowed down to the order in which the words appear. It is believed that the probability of the next word in a sequence can be reasonably estimated, given the n-1 preceding words. Given a large corpus it is possible to estimate these probabilities and

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 492–504, 2013.

use them to compute a probability of any given sequence of words. However, it should be mentioned, that the efficiency of an n-gram LM is language dependent. English, being the most important and common language for speech recognition research, caused popularity of n-grams. But, it has to be stressed, that English is positional, and as such, n-grams are very natural to be used and close to the logics of the language. In case of Polish and other inflectional languages like Finnish, Turkish and other (than Polish) Slavic languages, the order of the words is not the main logics of the language. As a result, an n-gram LM taken directly from ASR system designed for English is not necessarily the best choice.

For instance in Polish the expressions *dom Adama* and *Adama dom* (*Adam's house*) although not equally probable, express the same relation between these words. What is more the number of tokens in Polish and other inflectional languages is larger than in English, since words have many inflectional forms (e.g. *Adam, Adama, Adamowi, Adamem, Adamie, Adamowie, ...* are all forms of *Adam*).

The primary problem connected with n-gram based LMs is data sparsity – it is impossible to collect a corpus that would allow to compute the probabilities for any word sequence of a given length that might appear in the recognized speech. As a result there are sequences that lack probability estimation. Due to the fact that the number of tokens in inflectional languages is larger than in positional languages, this problem is amplified in the case of the former. Partial solution to this problem is usage of techniques such as smoothing, interpolation and backoff [2].

The second important problems are parentheses which seriously both introduce errors to LM if they appear in the training corpus, and can be very difficult to be recognized if they appear in speech.

There are also approaches that do not assume that words are the best building blocks of the LM. There are models based on morphs, morphemes as well as words clustered into categories (category-based models). The first two approaches are mostly used for agglutinative languages, where words are constituted from sub-word units having their own meaning and syntactic features [3,4]. The last approach might be applied to any language. Its idea is that several words (or morphemes) having similar meaning and syntactic features (e.g. nouns representing cities: *Washington*, *New York*, *Boston*) are substituted by a category, i.e. a words-cluster. As a result the number of distinct n-gram units is reduced and the LM is less sparse.

The advantage of such a language model is that it does not require a large corpus to be generated and since the number of categories might be substantially smaller than the number of words, it allows to use higher-order n-grams.

The method of building the LM presented in this paper follows the category-based approach to language modeling. The grammatical classes of the words are used as their categories. As a result, a sequence of words *Mary killed the bug* is converted to a sequence of a *noun* followed by a *verb*, a *determiner* and another *noun*.

The problem associated with such a language model is that in order to compute the estimated probabilities of n-grams we have to assign the grammatical classes to the words. This is a well known task of part-of-speech (POS) tagging. Although the POS tags might be assigned automatically, in our approach to compute the n-gram probabilities we use a manually tagged training corpus. Although it requires substantial amount of work, for many languages corpora large enough for building the LM are available.

Even though we do not use a POS tagger to build the model, we use this module for tagging the candidate sentences. This tagger is used to tag the n-best candidates generated by the acoustic module in order to compute the POS-sequence probability. In the conducted experiment we use an external component to perform the POS tagging, which heavily incurs the speed of the system. However, it is possible to integrate the POS tagger and optimize it for the ASR task.

The presented experiments were carried out for Polish – an inflectional language with a large number of distinct word-forms. As a result the data sparsity problem of the word-based LMs is more apparent. Recently a large (1 million) corpus of syntactically tagged texts was made available [5] which we used in our experiment. We also used the state-of-the-art POS tagger for Polish [6], which was trained on that corpus. The results for a testing corpus with 100 sentences show that using this LM to rescore the results provided by HTK acoustic model can reduce the word error rate (WER) by more than 10 percentage points.

2 Related Work

Generally, there is very little interest in using POS tags in ASR. Besides [7] and [8], which report negative results on applying POS taggers in ASR, there is little literature on this topic. In [8], a POS tagger [9] was tested as possible improvement in speech recognition of Polish [8]. However, the results were negative because the output of the tagger was too ambiguous.

In the approach described here, we limited the ambiguity by reducing the number of details considered in the model. Specifically we used only grammatical classes of the words, discarding the values of other grammatical categories. A new version of the tagger was applied which could have some impacts as well on the results. However, the difference in efficiency of taggers is probably not big enough to be the main source of the much better efficiency of LM. The tagger used in the experiment from 2008 [8] had an accuracy of 93.44%. The tagger used in the experiment described here [6] has a reported accuracy of 90.34%, which seems to be lower. Still, these numbers are not directly comparable, especially since the reference resources used to evaluate the taggers were different. In the past, it was a frequency dictionary containing approx. 600 thousands of segments, at present it is the rigorously tagged National Corpus of Polish, containing more than 1 million segments [5]. Suffice it to say, that the successor of the tagger used in [8] achieves 87.50% accuracy on that corpus.

3 Resources

The algorithm used to re-score the acoustic hypotheses requires the following resources:

- a corpus with POS tags assigned to words, which is used to generate the LM,
- an inflectional dictionary, which generates possible grammatical categories for the words,
- a POS tagger which selects the most probable grammatical category for each word,
- a test set with the hypotheses provided by the acoustic model used to test the performance of the algorithm.

In the experiment we use a 1-million subcorpus of the National Corpus of Polish (NKJP) [5], which, among other annotations, contains manually selected grammatical classes for all the text segments. The annotations are available in TEI (version 5) standard [10] in XML files. Each individual text is annotated separately on several levels:

- sentence segmentation,
- morphosyntactic tagging,
- semantic tagging,
- shallow syntactic tagging,
- named entities tagging.

The morphosyntactic tagging is the one used to build the LM. The tagging provided in NKJP covers the following phenomena:

- segmentation of the texts into sentences,
- segmentation of the sentences into segments (i.e. units with their own tags attached),
- segment lemmatization,
- grammatical classes of the segments,
- values of grammatical categories of the segments.

In NKJP there are 35 grammatical classes used to classify the segments. These do not correspond directly to the traditional parts-of-speech, such as *nouns*, *verbs* and *adjectives*. Mostly due to the fact that verbs are split into several distinct classes. These classes are described in Table 1.

Although it is easy to obtain the information about the grammatical classes from a single file using XPath queries, there are two technical problems associated with the corpus. The first is its structure – each text (3.9 thousands in total) is annotated in several files and occupies a single directory. As a result the files have to be processed individually, which is not very convenient. This is a bit surprising, since the predecessor of NKJP, IPI PAN corpus [11] was distributed in a binary form, with accompanying corpus server named Poliqarp [12]. The server simplified the access to the data and offered competitive performance.

The second problem is connected with the fact, that although the grammatical classes for each segment are available separately, the decision made by the annotator, i.e. the proper grammatical class of the segment in the given context is only available as a part of string containing the lemma, the class and the values of the grammatical categories concatenated using a colon (:). E.g. for the word *Zatrzasnął* the following tagging is provided: `zatrzasnąć:praet:sg:m1:perf`. In most of the cases this works fine, but there are segments such as *http://www.jeri.gwflota.com/main/*, which become ambiguous when concatenated with the tags. Although, it is possible to isolate the tags from the lemma by subtracting the lemma, which is available separately, it would be much easier to do, if the lemma and the tags were not concatenated.

In order to use the corpus to build the LM, all the grammatical classes found in the corpus were extracted using XPath queries, preserving the sentence segmentation. The sequences of the classes' tags from the sentences were saved in the following lines of a text file. As a result it was possible to use standard LM building tools, such as SRILM [13]. The file consisted of more than 86 thousands of lines and more than 1228 thousands of tokens. The statistic of the unigram counts are given in Table 1 (cf. [5] for the description of the tags).

The second resource used in the experiment is an inflectional dictionary. We use Morfeusz, which contains more than 200 thousands of lexemes and is distributed in the form of a finite state transducer [14]. The dictionary contains very broad general Polish vocabulary, but lacks proper names. From the point of view of speech processing, the most important feature of the dictionary is how it segments the words in a running text.

Although Polish is an inflectional language with several agglutinative features, the dictionary is quite conservative in identifying words as segments, with one important exception: the first and the second person of the past forms of verbs are split into two segments. E.g. the form *jadłem* ((I) ate) is divided into two separate segments: the core *jadł*, indicating the gender and the number of the verb and the agglutinative suffix *em*, indicating the first person. Although this is motivated by the fact, that the suffix might be attached to almost any word in the sentence and that there are several other such suffixes (mostly particles) this complicates the integration of the inflectional dictionary into the ASR framework.

It contrasts with the fact that in most of the cases the agglutinative suffix of the verb is attached to the verb. In the set of sentences used to test the performance of the model containing more than 100 examples, there were only 3 sentences with the agglutinative suffix and all of them were attached to the verb. Taking into account that the inclusion of this phenomenon would considerably complicate the model and its relatively low probability, we decided to exclude these sentences from the test set.

The third resource that was used in the experiment was the POS tagger. We decided to use the state-of-the art tagger, namely WCRFT – a tiered conditional random fields tagger [6]. This tagger is very flexible and might be used with a number of tagsets, however, the preparation of the training data as well as training of the model takes substantial amount of time, we decided to use the

Table 1. Unigram statistic of Polish parts-of-speech

POS	**Grammatical class**	**Tag**	**Count**	**P**
Noun	regular n.	subst	306 236	0.24929
Punctuation		interp	221 699	0.18048
Adjective	regular adj.	adj	125 559	0.10221
Preposition		prep	91 928	0.07483
Conjunction	coordinate conj.	conj	75 513	0.06147
Verb	finite form of v.	fin	60 164	0.04898
Verb	participle-like form of v.	praet	53 759	0.04376
Adverb		adv	51 960	0.04230
Kublik		qub	38 169	0.03107
Unknown word		ign	36 529	0.02974
Interjection		interj	28 372	0.02310
Conjunction	subordinate conj.	comp	22 753	0.01852
Verb	infinitive form of v.	inf	19 606	0.01596
Verb	passive participle	ppas	13 387	0.01090
Verb	gerund	ger	11 842	0.00964
Pronoun	3rd-person personal pron.	ppron3	11 476	0.00934
Pronoun	non-3rd-person personal pron.	ppron12	8 212	0.00669
Abbreviation		brev	8 200	0.00668
Numeral	cardinal	num	8 082	0.00658
Verb	agglutinative form of v.	aglt	7 654	0.00623
Verb	active participle	pact	5 587	0.00455
Verb	predicative form of v.	pred	3 973	0.00323
Verb	future form of the verb „to be”	bedzie	2 804	0.00228
Verb	present participle	pcon	2 644	0.00215
Verb	imperative form of v.	impt	2 524	0.00205
Noun	depreciative n.	depr	2 456	0.00200
Pronoun	reflexive pron.	siebie	2 142	0.00174
Verb	impersonal v.	imps	2 138	0.00174
Verb	„winien”-like verb	winien	813	0.00066
Burkinostka		burk	608	0.00049
Adjective	post-preposition adj.	adjp	580	0.00047
Adjective	pre-adjective adj.	adja	562	0.00046
Verb	perfective participle	pant	154	0.00013
Foreign form		xxx	146	0.00012
Numeral	collective num.	numcol	125	0.00010
Adjective	predicative adj.	adjc	55	0.00004
Total			**1228411**	**1**

model for Polish that is readily available for the tagger. It was trained on the same 1-million subcorpus of NKJP, that we use to build our POS n-gram model and it works well with the Morfeusz inflectional dictionary. It is reported that it achieves a precision of 90.34%, which makes it the best performing POS tagger for Polish[1].

108 recorded sentences or phrases were used for tests. They were spoken by one male, without any specially added noise, but in an office with working computers etc. The content of the corpus is mixed. There are some sentences with political context, like parts of parliament speeches (but not taken from Parliament transcripts). There are some lyrics of Kaczmarski song and speeches of Piłsudski and Balcerowicz.

HTK [17,18] was used to provide n-best list (limited to 600 of items) of acoustic hypotheses for sentences from the test corpus. The acoustic model was trained on CORPORA [19], which means that different speakers, sentences and recording devices were used than for the test set. The hypotheses were constructed as combinations of any words from the corpus as ordered lists of words. This model was trained in a way which allowed all possible combinations of all words in a dictionary to have more variations and to give opportunity for a language model to improve recognition.

4 Rescoring Algorithm

The general idea of the rescoring algorithm is as follows: when the acoustic model generates the n-best list of candidates, each candidate sentence is tagged with the POS tagger. Then the LM-based probability of the sequence of tags is computed and the hypotheses are scored according to the following equation:

$$P(h_i) = P(h_i)_{LM}^{\alpha} * P(h_i)_{AM}^{1-\alpha} \,, \tag{1}$$

where:

- $P(h_i)$ – the probability of the i-th hypothesis,
- $P(h_i)_{LM}$ – the probability of the i-th hypothesis according to the language model,
- $P(h_i)_{AM}$ – the probability of the i-th hypothesis according to the acoustic model,
- α – the weight of the LM component.

The weighting factor α is introduced, since the LM assigns much higher probabilities to the sequences, because it picks only one class out of 38 (35 grammatical classes + out-of-vocabulary words + start and end of a sentence), while the AM module have thousands of words to consider.

[1] In the past it was reported that several POS taggers of Polish achieve better POS tagging performance. However Radziszewski uses a refined methodology for computing the tagger performance so the results are not comparable as such. In the paper cited the results are compared directly for WMBT [15], PANTERA [16] and WCRFT.

Table 2. Some sentences from the test corpus with tags provided by WCRFT and their English translations

Platforma obywatelska wymaga funkcjonowania klubu w czasie obrad sejmu. `subst adj fin ger subst prep subst subst subst interp` Civic Platform expects the club to operate during parliament proceedings.
Łatwo skierować czynności do sądu. `adv inf subst prep subst interp` It is easy to move actions to court.
Wniosek rolniczego związku znajduje się w ministerstwie. `subs adj subst fin qub prep subst interp` The petition of the agricultural union is in the ministry.
Projekt samorządu ma wysokie oczekiwania finansowe. `subst subst fin adj subst adj interp` The municipality project has high financial expectations.
Fundusz społeczny podjął działania w ramach obecnego prawa cywilnego. `subst adj praet subst prep subst adj subst adj interp` The communal foundation took steps according to existing civil law.
Uchwała rządowa dotycząca handlu i inwestycji przedsiębiorstw państwowych w rynek nieruchomości. `subst adj pact subst conj subst subst adj prep subst subst interp` The government act on trade and investments of public enterprises in the estate market.
Panie marszałku, wysoka izbo. `subst subst interp adj subst interp` Mr speaker, House. (common way to start a speech in the Polish Parliament)
Bezpieczeństwo jest bardzo ważne. `subst fin adv adj interp` The safety is very important.
Skrzydła im ścierpły w długiej niewoli. `subst ppron3 praet prep adj subst interp` Their wings went numb in a long captivity.
Chcą być przeklęci pierwsi. `fin subst adj adj interp` They want to be cursed first.
Świat odkrywa na nowo wciąż dramaty moje. `subst fin prep adv adj subst adj interp` The world discovers my drama all over from the beginning.
Spały wilczki dwa zupełnie ślepe jeszcze. `praet subst num adv adj qub interp` Two baby wolfs slept completely blind.
Zajmują ją w imieniu władzy naczelnej rządu narodowego. `fin ppron3 prep subst subst adj subst adj interp` They take it with authority of the supreme authority of the national government.
Socjalizm zostawił w Polsce w owym roku spodlony pieniądz. `subst praet prep subst prep adj subst ppas subst interp` Socialism left in Poland that year degraded money.

The probability assigned by the LM is computed using the part-of-speech n-grams collected from the NKJP subcorpus. In general the probability of a given sequence of tags using n-grams is computed as the maximum likelihood estimation (MLE):

$$P(v_i|v_{i-N+1}...v_{i-1}) = \frac{c(v_{i-N+1}...v_i)}{c(v_{i-N+1}...v_{i-1})} \ , \tag{2}$$

where:

- $P(v_i|v_{i-N+1}...v_{i-1})$ – is the probability of the category v_i assuming the sequence of $v_{i-N+1}...v_{i-1}$ previous categories,
- $c(v_{i-N+1}...v_i)$ – is the number of sequences observed in the corpus, consisting of categories from v_{i-N+1} to v_i.

However, the direct application of MLE faces the problem of sequences that have never been seen in the training data. Their count $c(v_{i-N+1}...v_i)$ equals zero and their probability is also 0. As a result, the whole sequence has 0 probability. There are many methods used to overcome this problem, namely [20,21]:

- smoothing,
- interpolation,
- backoff.

Smoothing assigns the probability of unseen n-grams directly by estimating it using the n-grams with low frequency (e.g. n-grams which occurred only once). To compensate the mass of probability that was distributed to the unseen n-grams it discounts the counts of the n-grams that have been seen in the corpora. In interpolation lower order n-grams are combined linearly and the probability is non-zero, at least if the token in question have ever been seen in the training data. The last method – backoff – uses strategy similar to interpolation, but it uses lower order n-gram only if the count for the given sequence of the length n is 0.

Although Kneser-Ney discounting [22] is the most popular and the best performing method used for large word dictionaries, it does not work if the dictionary is very small, like in the POS-based n-grams[2]. That is why, we use Witten-Bell discounting, namely the backoff version of this method.

To define the probability of a sequence of tags, we first define the discounted frequency

$$F(v_{j-N+1}...v_j) = \frac{c(v_{j-N+1}...v_j)}{n(v_{j-N+1}...*) + c(v_{j-N+1}...v_{j-1})} \ , \tag{3}$$

where $n(v_{j-N+1}...*)$ – the number of sequences of length n with the prefix $v_{j-N+1}...v_{j-1}$ that appeared only once in the corpus.

Then the probability of a sequence of tags is estimated as

$$P(v_j|v_{j-N+1}...v_{j-1}) = \begin{cases} F(v_{j-N+1}...v_j) & c(v_{j-N+1}...v_j) > 0 \\ \beta P(v_j|v_{j-N+2}...v_{j-1}) & otherwise \end{cases} \ , \tag{4}$$

[2] Cf. `http://www.speech.sri.com/projects/srilm/manpages/srilm-faq.7.html`

where $\beta = \frac{1-\sum F(v_{j-N+1}...v_j)}{1-\sum F(v_{j-N+2}...v_j)}$ – the backoff weight.

The final probability of a sentence computed using the LM is

$$P(h_i)_{LM} = \prod_{w_j \in h_i} P(V(w_j)|V(w_{j-N+1})...V(w_{j-1})) \ , \tag{5}$$

where $V(w_j)$ is the grammatical class assigned to the word w_j.

It is assumed that the grammatical categories corresponding to the words present in the sentence are fully determined. This assumption is accomplished by incorporating the POS tagger into the system. Although many of the words have several possible interpretations defined in the inflectional dictionary, only one of them is selected according to the tagger. The remaining options are not taken into account.

The ranking of the hypotheses is defined as follows

$$\begin{gathered} R(h_i) = logP(h_i) = \alpha log(P(h_i)_{LM}) + (1-\alpha)log(P(h_i)_{AM}) = \\ \alpha \sum_{w_j \in h_i} P(V(w_j)|V(w_{j-N+1})...V(w_{j-1})) + (1-\alpha) \sum_{w_j \in h_i} P(w_j|s)_{AM} \ , \end{gathered} \tag{6}$$

where $P(w_j|s)_{AM}$ is the probability of the word w_j conditioned on the speech signal s, computed by the acoustic module. The ranking is computed in log-space, since the acoustic probabilities are very low and are subject to underflow errors.

The value of the parameter α might be optimized on the held out corpus.

5 Results

To measure the performance of the POS-based LM we used the following measures:

- word error rate reduction (WERR),
- correct sentence position improvement (CSPI) in the n-best list of hypotheses.

Word error rate is defined as the minimum edit distance [21, p. 73-77] between the correct sentence and the hypothesis with the highest probability. In our setting each edition has the same cost. Word error rate reduction is the number of percentage points the word error rate has reduced after applying the LM.

Table 3. The performance of the POS-based language model

N	**WERR**$_{best}$ %	**CSPI**$_{best}$	**WERR**$_{1/2}$%	**CSPI**$_{1/2}$
1	12.55	20.28	2.42	1.25
3	12.61	38.61	5.12	30.93
5	12.69	40.03	5.14	31.57

Table 4. The most popular 3-grams of POS tags in Polish

POS tag 3-gram	%	POS tag 3-gram	%
`subst interp </s>`	3.02	`subst interp subst`	0.69
`adj subst interp`	1.58	`adj subst subst`	0.64
`subst subst interp`	1.26	`subst subst subst`	0.62
`subst prep subst`	1.13	`subst conj subst`	0.53
`prep subst interp`	1.12	`interp interp </s>`	0.53
`prep adj subst`	0.94	`prep subst adj`	0.51
`subst adj interp`	0.89	`subst subst adj`	0.50
`prep subst subst`	0.77	`interp subst interp`	0.49
`subst adj subst`	0.72	`subst interp conj`	0.46
`adj interp </s>`	0.69	`subst interp adj`	0.45

CSPI is defined as the improvement in position of the correct sentence between the n-best list generated by the acoustic model and the list generated by the combined language and acoustic models.

The results of the experiments carried out on the testing corpus are presented in Table 3. They are reported for cases where AM and LM had the same weight (after scaling the probability in order to compensate the difference in the size of AM and LM): $WERR_{1/2}$, $CSPI_{1/2}$ and for the best cases (i.e. for the optimal value of the parameter α): $WERR_{best}$, $CSPI_{best}$. The average WER achieved by the sole AM was 37.23% and the average position of the correct hypothesis was 50 (among 600 hypotheses).

The best WERR achieved for 1-grams, 3-grams and 5-grams is very similar, but it should be noted, that it is obtained for the specifically selected parameter α. The CSPI measure shows large differences in the performance between 1-grams and 3-grams. It is especially apparent when the weight for the LM is the same as for the AM. The difference between the 3-gram LM and 5-gram LM is much smaller – both in the case of the LM with the best parameter and with the parameter set to a predefined value.

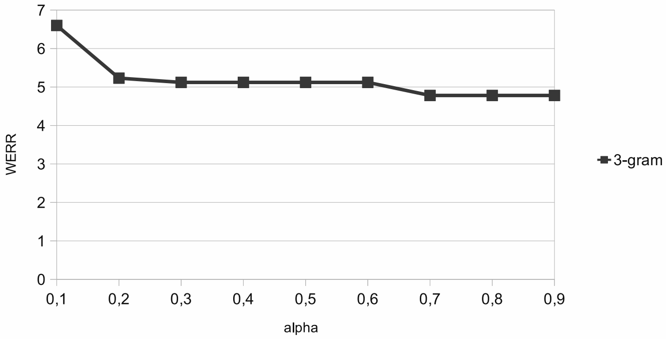


Fig. 1. Word error rate reduction (WERR) for different values of the parameter α in range 0-1.0

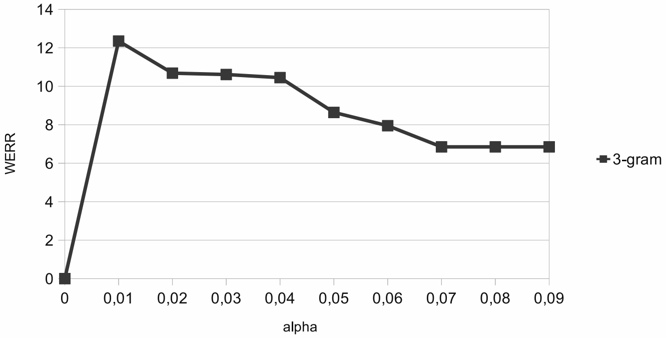


Fig. 2. Word error rate reduction (WERR) for different values of the parameter α in range 0-0.1

Figures 1 and 2 show the dependence of the performance of the rescoring algorithm on the parameter α. The proper selection of the parameter is crucial for the performance of the algorithm.

In Table 4 we also report the POS trigrams that are the most popular in Polish.

6 Conclusions

We conclude that the simplified POS tags are very good source of information for statistical language models of Polish. Applied on 108 test sentences recognized acoustically by HTK with a fixed weight they reduced WERR by 5% points and with the optimal weight – over 12 % points.

Assuming that a manually tagged corpus is available, a POS-based LM is much easier and cheaper to build than a word-base LM. This is particularly important for the inflected languages like Polish.

In the following research we are planning on integrating the POS-based model into a larger ASR system.

Acknowledgments. This work was supported by LIDER/37/69/L-3/11/ NCBR/2012 grant.

References

1. Ziółko, B., Skurzok, D.: N-grams model for Polish. Speech and Language Technologies, Book 2, pp. 107–127. InTech Publisher (2011)
2. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn. Prentice-Hall, Inc., New Jersey (2008)
3. Hirsimaki, T., Pylkkonen, J., Kurimo, M.: Importance of high-order n-gram models in morph-based speech recognition. IEEE Transactions on Audio, Speech and Language Processing 17(4), 724–732 (2009)
4. Sak, H., Saraçlar, M., Gungor, T.: Morpholexical and discriminative language models for turkish automatic speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 20(8), 2341–2351 (2012)

5. Szałkiewicz, Ł., Przepiórkowski, A.: Anotacja morfoskładniowa. In: Narodowy Korpus Języka Polskiego, pp. 59–96. Wydawnictwo Naukowe PWN (2012)
6. Radziszewski, A.: A tiered CRF tagger for polish. In: Bembenik, R., Skonieczny, Ł., Rybiński, H., Kryszkiewicz, M., Niezgódka, M. (eds.) Intelligent Tools for Building a Scientific Information Platform. SCI, vol. 467, pp. 215–230. Springer, Heidelberg (2013)
7. Niesler, T., Whittaker, E., Woodland, P.: Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 177–180. IEEE (1998)
8. Ziółko, B., Manandhar, S., Wilson, R.C., Ziółko, M.: Language model based on pos tagger. In: Proceedings of SIGMAP 2008 the International Conference on Signal Processing and Multimedia Applications, Porto (2008)
9. Piasecki, M.: Hand-written and automatically extracted rules for polish tagger. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 205–212. Springer, Heidelberg (2006)
10. Burnard, L., Sperberg-McQueen, C.: Guidelines for electronic text encoding and interchange. In: Association for Computers and the Humanities, Association for Computational Linguistics, Association for Literary and Linguistic Computing (1994)
11. Przepiórkowski, A.: Korpus IPI PAN. Wersja wstępna. Instytut Podstaw Informatyki PAN (2004)
12. Janus, D., Przepiórkowski, A.: Poliqarp 1.0: Some technical aspects of a linguistic search engine for large corpora. In: The Proceedings of Practical Applications of Linguistic Corpora (2005)
13. Stolcke, A., et al.: SRILM-an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 901–904 (2002)
14. Saloni, Z., Woliński, M., Wołosz, R., Gruszczyński, W., Skowrońska, D.: Słownik gramatyczny języka polskiego (Eng. Grammatical dictionary of Polish) (2102)
15. Radziszewski, A., Śniatowski, T.: A memory-based tagger for polish. In: Proceedings of the 5th Language & Technology Conference, Poznań (2011)
16. Acedański, S.: A morphosyntactic brill tagger for inflectional languages. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL 2010. LNCS, vol. 6233, pp. 3–14. Springer, Heidelberg (2010)
17. Young, S.: Large vocabulary continuous speech recognition: a review. IEEE Signal Processing Magazine 13(5), 45–57 (1996)
18. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: HTK Book. Cambridge University Engineering Department, UK (2005)
19. Grocholewski, S.: CORPORA - speech database for Polish diphones. In: Proceedings of Eurospeech (1997)
20. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, pp. 310–318. Association for Computational Linguistics (1996)
21. Jurafsky, D., Martin, J., Kehler, A.: Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2nd edn. Prentice Hall (2009)
22. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: 1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1995, vol. 1, pp. 181–184. IEEE (1995)

An Empirical Study of Reducing Multiclass Classification Methodologies

R. Kyle Eichelberger and Victor S. Sheng

Department of Computer Science, University of Central Arkansas,
Conway, Arkansas, USA 72035
rke06001@cub.uca.edu, ssheng@uca.edu

Abstract. One-against-all and one-against-one are two popular methodologies for reducing multiclass classification problems into a set of binary classifications. In this paper, we are interested in the performance of both one-against-all and one-against-one for basic classification algorithms, such as decision tree, naïve bayes, support vector machine, and logistic regression. Since both one-against-all and one-against-one work like creating a classification committee, they are expected to improve the performance of classification algorithms. However, our experimental results surprisingly show that one-against-all worsens the performance of the algorithms on most datasets. One-against-one helps, but performs worse than the same iterations of bagging these algorithms. Thus, we conclude that both one-against-all and one-against-one should not be used for the algorithms that can perform multiclass classifications directly. Bagging is an better approach for improving their performance.

Keywords: One-Against-All, All-at-Once, One-Against-One, C4.5, SVM, Logistic Regression, Naive Bayes, multiclass classification.

1 Introduction

There exist two most well-known approaches of breaking multiclass classification issues into several binary classification ones: one-against-all (OAA in short) [2, 3, 9, 15] and one-against-one (OAO in short) [3, 9, 15]. One-against-all converts multiclass problems into binary problems, by building a classifier for each class in the classes of a multiclass dataset. Before building each classifier for a specific class (positive), it converts all the rest classes into negative. That is, it builds n models for a dataset with n classes. One-against-one (also called pairwise coupling) decomposes the multiclass problem into binary class problems by taking any two classes as a pair and ignoring the remaining ones. For a multiclass classification problem with n classes, we have $\frac{n(n-1)}{2}$ class pairs, thus $\frac{n(n-1)}{2}$ classifiers needed to be built.

Previous research work [2, 9, 15, 23, 24, 25] made comparison between these two approaches (OAA and OAO). Few work [3, 5] made comparison for OAA and OAO against the multiclass classification algorithms directly (coined as all-at-once [2, 3],

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 505–519, 2013.

AAO in short) to see how much improvement OAA or OAO can achieve. And they only made comparison using a specific learning algorithm, either SVM [5] or naïve Bayes [3]. Hsu and Lin [5] made comparison among the three approaches (OAA, OAO, and AAO) with least squares SVMs. They concluded that OAO performs the best, and OAA and AAO are almost the same. But fuzzy membership functions are introduced into OAA and OAO. Thus, we do not know which is better among the simplest versions of the approaches. It also attracts us to study why OAA does not perform better than AAO [3, 5]. Intuitively, OAA should perform better than AAO, because it implicitly uses the ensemble technique by building n models (n is the number of classes). In addition, some previous work [23] claimed that OAO does not perform better than AAO. This is completely against the common sense and other previous work [3, 5, 9, 24, 25]. We know that OAO builds $\frac{n(n-1)}{2}$ models, normally more than the number (n) models built in OAA. Intuitively, it should perform better than OAA and AAO. This confliction work stimulates us to investigate the performance of them more comprehensively.

Besides, previous research conducted experiments on a few datasets. We are going to make thorough comparisons on 25 datasets and further make fair comparisons by bagging AAO with the same number of iterations as its opponents. In addition, most previous research investigated the performance of OAA and OAO on SVM only [15] or naïve bayes only [23]. A few also investigated on multiple learning algorithms. However, they only make comparisons on either OAA or OAO [15], without comparing with AAO at all. Thus, we do not have the general knowledge whether OAA and OAO improve the performance of multiclass learning algorithms. We are going to make general comparisons among OAA, OAO, AAO and its variants on four basic representative learning algorithms (decision tree, naïve bayes, support vector machine, and logistic regression), which also used in [15]. Further, this study would provide some generalizations that could be useful for the multi-label classifications [22].

The rest of the paper is organized as follows: Section 2 introduces the two well-known approaches (OAA and OAO), which convert multiclass classifications into a set of binary classifications. Besides, the baseline AAO and its variants are introduced in this section. Section 3 simply reviews the four basic learning algorithms (decision trees, naïve bayes, support vector machines, and logistic regression) used in our experiments. In Section 4, we describe the experiments we have made. They consist of the setting of the experiments, the experimental results, and the analysis of the experiments results. Section 5 concludes with a summary of our work and a discussion of future work.

2 Methodology

Two well-known approaches OAA and OAO reduce multiclass classifications into a set of binary classifications. In this section, we first review the two approaches. We will compare their performances under four chosen learning algorithms in Section 4.

After OAA and OAO reduce a multiclass classification into a set of binary classifications, a committee of binary classifiers will be built for these binary classifications. To make fair comparisons between AAO with either OAA or OAO, bagging is applied to AAO to build the same number of classifiers for AAO. This is called AAOvsOAA. Similarly, in order to make fair comparisons between AAO and OAO, we apply bagging to AAO to build the same number of classifiers for AAO as the number of classifiers built in OAO. This is called AAOvsOAO. The experiments are conducted in Section 4.

2.1 One-Against-One (OAO)

The fundamental principle of one-against-one (also called pairwise coupling) is to decompose the multiclass problem into binary class problems by taking any two classes as a pair and ignoring the remaining ones. For a multiclass classification problem with n classes, we have $\frac{n(n-1)}{2}$ class pairs, thus $\frac{n(n-1)}{2}$ classifiers needed to be built. For example, for a classification problem with three classes, we will have three pairs (class 0 vs. class 1, class 1 vs. class 2, class 0 vs. class 2). For each pair, only the training examples belonging to the two classes are kept. The rest which do not belong to either of the two classes are removed. That is, we convert the three class problem (assuming 0, 1 and 2 are the classes) into three binary problems. For each binary problem, we need to build a classifier which only makes prediction for the test examples i.e. finding the probability of classifying each test example into these two classes.

When creating a training dataset for a pair of classes, OAO parses through the training set and removes all other class values. Thus, this process leaves the training set with significantly fewer examples. The classifier is then built from the new smaller binary set and is used to analyze the test set.

2.2 One-Against-All (OAA)

One-against-all is another well-known standard approach used to convert multiclass problems into binary problems. Different from constructing pairs of classes in one-against-one, One-against-all builds a classifier for each class in the class set. Before building each classifier for a specific class, it converts all the rest classes into one against class. For example, if we have a multiclass problem with n classes, we keep class X unchanged and change the remaining classes into another single class Y. Thus, we have a binary problem. For an n multiclass problem, we need to transform it into n binary problems. For each binary problem, we need to build a classifier for a specific class. This classifier only makes prediction for a test example whether it belongs to the specific class or not. After n predictions are available for a test example, the classifier classifies it into the class where it has the highest probability. Comparing with OAO, it builds fewer classifiers.

OAA has an obvious drawback. When OAA creating the training set for a specific class, it converts all the rest examples belonging to other classes into an arbitrary class Y. This converting will create the imbalanced issue, which makes the problem more complicated. Besides, this procedure also breaks the universal i.i.d assumption of learning.

2.3 All-at-Once (AAO)

All-at-once is the simplest way, which performs multiclass classifications directly. That is, it classifies a test example into anyone of the multiple classes using one decision function only. In fact, it is not a methodology approach at all. In order to investigate the performance of the above two approaches (OAA and OAO), it is coined to indicate that classification algorithms perform multiclass classifications without reducing multiclass classifications to binary ones. That is, AAO usually works as the baseline. In this paper, we also empirically compare the two well-known approaches (OAA and OAO) with the baseline AAO to investigate whether they do perform better than AAO [2, 3]. Note that AAO utilizes only one model to classify the test set. Thus, its time computation cost is the lowest, comparing to OAA and OAO.

2.4 Bagging AAO

As stated the above AAO is overly simplistic in that it predicts for every class value with only one model per dataset. The other two methods (OAO and OAA) utilize different data modification techniques in order to reduce multiclass classifications into binary ones. Thus, they have to build multiple models, instead of one. OAA has to build n models (again, n is the number of classes). OAO has to build $\frac{n(n-1)}{2}$ models. Multiple models give an unfair advantage to OAA and OAO over AAO. In order to make fair comparisons, we apply bagging to AAO to build multiple models. The number of models built using bagging is exactly the same as the number of models built by its opponent. That is, when comparing with OAA, we build n models. This version of AAO is noted as AAOvsOAA since then. When comparing with OAO, we build $\frac{n(n-1)}{2}$ models. This version of AAO is called AAOvsOAO since then. It is interesting to investigate whether OAA and OAO perform better than bagging AAO.

3 Learning Algorithms

There exist more than 20 different learning algorithms, including the basic ones and improved variations. According to the categorization of WEKA [18], we choose a fundamental algorithm from each category. That is, we are going to investigate the performance of both one-against-all and one-against-one of four basic learning algorithms. They are decision tree, naïve bayes, support vector machine, and logistic regression, following the previous research [15]. In addition, these algorithms are chosen to investigate the performance of OAA and OAO, because we can compare their performance with the direct multiclass classification approach all-at-once (AAO in short).

A decision tree algorithm (DT in short) [17] partitions the input space into small sets, and labels them with one of the various output categories. That is, it iteratively selects a specific attribute to extend the learning model. According to the values of the specific attribute, a large group of cases are categorized into sub-groups. The essence of the algorithm is the criteria of selecting a specific attribute from the set of attributes available. There exist several criteria, such as accuracy improvement, entropy reduction, information gain, and gain ratio (details of these criteria can be found in [1]). Decision tree algorithms can perform multiclass classifications directly. However, it is also well-known that decision tree algorithms have a successful performance on binary classifications. It is interesting to investigate whether the two well-known approaches can further improve its performance on multiclass classifications.

Naïve bayes (NB in short) [19] is based on bayes theorem. Specifically, it is based on the properties of estimating the frequency of each value of every attribute under a given class from the obtained dataset. Like decision tree algorithms, it can perform multiclass classifications directly. We are interested in comparing the performance of applying the two approaches OAA and OAO to naïve bayes for performing multiclass classifications, and further investigating whether these approaches improve the performance of naïve byes. Intuitively, naïve bayes is a based on statistics. The conditional probabilities inside its posterior probability calculation are the same, estimating either via OAO or via AAO.

Support vector machine (SVM in short) [21] is one of the kernel approaches for classification. It constructs a hyperplane in high dimension space to classify cases into different classes. Its intuition is to find the hyperplane that has the largest distance to the nearest training cases. These nearest training cases are commonly referred as support vectors. According to the vectors on each side, a sub-hyperplane can be built for each side. The maximum margin between the two sub-hyperpanes is achieved to reduce the general classification errors. Support vector machines were originally proposed for binary classification. It has proved extremely successful for binary classification. Both OAA and OAO are the well-known approaches for extending SVM for multiclass classifications. It is interesting to make comparisons between the approaches.

Logistic regression (logistic in short) [20], like naïve bayes discussed above, is part of a category of statistical models called generalized linear models. However, unlike the independent assumption among the variables in naïve bayes, logistic regression has no such assumption. In addition, it also makes no assumption about the distribution of the independent variables. Although both logistic regression and naïve bayes are statistical models, the basic ideas of them are different. Thus, it is also interesting to study the effect of OAA and OAO on logistic regression.

4 Experiments

In this section, we will make thorough comparisons among the three approaches OAA, OAO, AAO and its two variants (AAOvsOAA and AAOvsOAO) using four representative basic learning algorithms: decision tree, naïve bayes, support vector machine, and logistic regression, which also used in [15].

Table 1. Description of the datasets used in the experiments

Dataset	#Instances	#Attributes	#Classes
Anneal	898	39	6
anneal-orig	898	39	6
arrhythmia	451	280	16
audiology	226	70	24
autos	205	26	7
balance-scale	625	5	3
cmc	1472	10	3
dermatology	365	35	6
glass	214	10	7
heart-cleveland	303	14	5
heart-hungarian	294	14	5
hypothyroid	3772	30	4
iris	150	5	3
letter	20000	17	26
lymph	148	19	4
primary-tumor	339	18	22
red-wine	1598	12	11
segment	2310	20	7
soybean	683	36	19
splice	3190	62	3
vehicle	846	19	4
vowel	990	14	11
waveform	5000	41	3
white-wine	4897	12	11
zoo	101	18	7

4.1 Experimental Setup

In this section, we conduct experiments to investigate the performance of one-against-all and one-against-one by comparing with all-at-once directly. In our experiments, we use 25 multiclass classification datasets, listed in Table 1, downloading from the UCI Machine Learning Repository [4]. These datasets are chosen because they are the multiclass classification datasets in the Repository. Besides, they have different number of classes, different number of attributes and different number of instances. Furthermore, the numbers of nominal and numeric attributes within these datasets are varied greatly. As naïve Bayes only can deal with nominal attributes, all continuous attributes in the datasets listed in Table 1 are discretized by Fayyad and Irani [14]'s entropy-based method.

The experiments are conducted within WEKA [14]. The performance of the different approaches described in Section 2 is measured in prediction accuracy. The results are obtained from the implementation of WEKA for the four basic learning algorithms with default parameter settings. That is, we use J48 for decision tree, NaiveBayes for naïve bayes, SMO for support vector machine, and Logistic for logistic regression [18]. The objective of these experiments is to investigate the performance of the two well-known methodologies (OAA and OAO), comparing with the baseline (AAO). Thus, we do not perform any kind of optimization for each learning algorithm, such as adjusting the pruning parameters for J48, and tuning the regularization parameters for SMO.

For each dataset, we repeat the experiment 10 runs. Every run randomizes the dataset, in an effort to produce variable results possible. The average results with standard deviation are reported for each methodology under each algorithm. In every run, 30% of examples were held out, as the test set from which we calculate the generalization performance of the approaches under different learning algorithms. The remaining 70% functioned as the training examples.

4.2 Experimental Results

Tables 2 through 5 show the experimental results of the reduction methodologies (OAA and OAO) and the baseline AAO with its bagging variants (AAOvsOAA and AAOvsOAO) on the 25 datasets. For each dataset, we show the average accuracy with its standard deviation of each methodology using different learning algorithms as the base learner to build the classification models. Table 2 shows the experimental results using J48 as the base learner. Table 3 shows the experimental results using NaivBayes as the base learner. Table 4 shows the experimental results using SMO as the base learner. Table 5 shows the experimental results using Logistic as the base learner.

Table 2. Test accuracies (%) using J48 as the base learner

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
anneal	98.40 ± 0.58	98.03 ± 0.74	97.85 ± 1.07	98.26 ± 0.52	98.00 ± 0.00
Anneal-orig	93.94 ± 1.83	91.49 ± 2.47	90.93 ± 1.92	92.78 ± 1.57	92.85 ± 0.00
arrhythmia	65.26 ± 3.69	71.11 ± 3.49	64.78 ± 2.93	74.19 ± 3.64	75.00 ± 1.56
audiology	62.84 ± 5.99	72.69 ± 7.96	71.47 ± 5.42	77.06 ± 0.00	77.06 ± 0.00
autos	71.31 ± 6.88	67.38 ± 7.90	72.74 ± 7.84	76.61 ± 0.00	76.29 ± 2.28
Balance-scale	78.93 ± 2.35	78.29 ± 1.89	77.23 ± 2.59	80.59 ± 2.26	80.59 ± 2.26
cmc	52.36 ± 3.03	51.88 ± 1.71	52.96 ± 2.22	52.13 ± 0.80	52.13 ± 0.80

Table 2. (*Continued*)

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
dermatology	91.38 ± 2.53	96.15 ± 1.13	92.82 ± 4.25	95.27 ± 1.93	95.55 ± 1.29
glass	65.47 ± 6.52	68.59 ± 6.27	66.77 ± 5.14	68.31 ± 4.35	72.46 ± 0.00
Heart-c	78.56 ± 3.67	79.33 ± 3.60	78.68 ± 3.56	78.57 ± 3.89	79.56 ± 6.99
Heart-h	78.64 ± 3.29	78.64 ± 2.97	78.65 ± 3.18	78.76 ± 1.59	79.66 ± 0.00
hypothyroid	99.42 ± 0.18	99.48 ± 0.21	99.51 ± 0.22	99.45 ± 0.25	99.52 ± 0.25
iris	93.18 ± 2.40	93.41 ± 3.11	94.22 ± 2.39	94.22 ± 1.57	94.22 ± 1.57
letter	85.24 ± 0.63	92.92 ± 0.38	86.78 ± 0.39	92.77 ± 0.28	93.43 ± 0.08
lymph	78.41 ± 4.45	76.59 ± 4.80	76.00 ± 3.44	77.56 ± 3.14	78.22 ± 1.57
Primary-tumor	36.14 ± 5.12	42.38 ± 2.42	39.61 ± 2.86	42.45 ± 2.08	42.94 ± 0.69
redWine	58.94 ± 2.27	61.02 ± 2.26	56.67 ± 1.94	64.77 ± 3.68	65.33 ± 4.42
segment	95.23 ± 0.79	96.91 ± 0.79	96.45 ± 0.48	96.48 ± 1.22	96.93 ± 0.71
soybean	89.75 ± 2.28	92.11 ± 1.98	90.00 ± 2.16	91.07 ± 3.10	92.10 ± 2.41
splice	94.45 ± 0.74	94.41 ± 0.40	93.81 ± 0.57	93.32 ± 1.40	93.32 ± 1.40
vehicle	70.04 ± 1.78	73.20 ± 2.16	71.57 ± 2.25	73.66 ± 0.84	73.27 ± 1.67
vowel	72.67 ± 3.73	84.73 ± 3.10	76.06 ± 3.00	85.62 ± 2.14	88.42 ± 0.48
waveform	72.81 ± 0.72	76.40 ± 1.75	75.11 ± 1.21	77.99 ± 0.14	77.99 ± 0.14
whitewine	56.96 ± 1.04	61.08 ± 1.22	56.29 ± 1.21	63.03 ± 1.64	65.72 ± 1.54
zoo	92.67 ± 4.10	93.00 ± 5.08	93.55 ± 4.81	92.26 ± 2.28	92.26 ± 4.56
Average	**77.32 ± 2.82**	**79.65 ± 2.79**	**78.02 ± 2.68**	**80.69 ± 1.77**	**81.31 ± 1.47**

Table 3. Test accuracies (%) using NaiveBayes as the base learner

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
anneal	86.02 ± 3.31	86.95 ± 3.34	86.07 ± 3.17	88.70 ± 0.26	86.33 ± 0.79
Anneal-orig	77.81 ± 2.97	80.93 ± 1.74	74.74 ± 1.30	78.74 ± 2.88	77.67 ± 3.14
arrhythmia	63.41 ± 5.22	63.11 ± 3.73	60.29 ± 4.41	63.53 ± 4.68	63.38 ± 5.20
audiology	64.33 ± 6.95	60.90 ± 6.09	64.85 ± 6.02	64.71 ± 3.12	65.00 ± 4.16
autos	59.67 ± 5.14	57.54 ± 4.19	56.61 ± 3.91	59.68 ± 4.56	58.87 ± 1.14
Balance-scale	88.77 ± 1.38	88.34 ± 0.83	88.83 ± 1.28	88.19 ± 1.13	88.72 ± 1.50
cmc	48.37 ± 2.54	47.48 ± 2.82	47.53 ± 2.60	46.99 ± 2.56	48.05 ± 1.92
dermatology	96.97 ± 1.62	96.97 ± 1.37	97.27 ± 1.48	96.91 ± 0.00	97.09 ± 0.64
glass	52.50 ± 6.88	51.88 ± 7.93	48.00 ± 4.34	50.31 ± 1.09	50.46 ± 10.88
Heart-c	82.67 ± 3.28	82.78 ± 2.88	82.86 ± 3.24	82.53 ± 6.99	83.30 ± 5.44
Heart-h	84.55 ± 3.64	83.41 ± 4.29	84.49 ± 3.38	83.82 ± 2.38	83.93 ± 2.38
hypothyroid	94.51 ± 0.60	95.34 ± 0.58	95.50 ± 0.60	95.54 ± 0.56	95.49 ± 0.31
iris	93.41 ± 3.46	94.77 ± 2.84	95.11 ± 1.75	94.89 ± 1.57	95.33 ± 1.57
letter	64.05 ± 0.58	64.31 ± 0.65	64.17 ± 0.57	64.30 ± 0.46	64.23 ± 0.31

Table 3. (*Continued*)

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
lymph	85.23 ± 3.90	83.64 ± 4.65	85.78 ± 4.22	85.56 ± 6.29	85.11 ± 0.00
Primary-tumor	49.01 ± 5.27	48.02 ± 4.65	48.24 ± 5.14	47.84 ± 6.93	48.33 ± 3.47
redWine	54.84 ± 1.76	53.47 ± 2.56	54.00 ± 2.03	53.83 ± 2.36	54.06 ± 2.80
segment	80.81 ± 1.15	81.29 ± 0.85	81.20 ± 1.04	81.31 ± 1.22	81.30 ± 1.94
soybean	91.96 ± 2.11	91.72 ± 2.30	92.29 ± 2.25	91.90 ± 3.10	91.95 ± 2.76
splice	94.82 ± 0.61	95.42 ± 0.38	95.43 ± 0.46	95.46 ± 0.00	95.34 ± 0.22
vehicle	44.23 ± 3.04	44.58 ± 3.08	44.17 ± 3.00	45.28 ± 1.67	45.67 ± 2.78
vowel	59.83 ± 3.32	61.59 ± 3.66	60.98 ± 3.11	61.38 ± 2.14	62.19 ± 0.95
waveform	78.21 ± 1.24	79.80 ± 1.03	79.86 ± 0.97	79.68 ± 0.47	79.78 ± 0.61
whitewine	46.85 ± 1.46	45.67 ± 1.34	44.95 ± 1.25	45.18 ± 0.10	45.28 ± 0.63
zoo	94.33 ± 4.73	95.00 ± 2.36	94.84 ± 2.72	94.52 ± 0.00	94.52 ± 2.28
Average	**73.49 ± 3.04**	**73.40 ± 2.81**	**73.12 ± 2.57**	**73.63 ± 2.26**	**73.66 ± 2.31**

Table 4. Test accuracies (%) using SMO as the base learner

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
anneal	96.77 ± 1.16	96.84 ± 0.79	96.96 ± 0.98	96.74 ± 1.05	96.78 ± 0.52
Anneal-orig	85.20 ± 2.09	87.47 ± 1.57	87.37 ± 1.85	88.63 ± 0.79	87.85 ± 2.88
arrhythmia	66.37 ± 4.98	64.96 ± 5.38	68.09 ± 4.26	68.68 ± 0.52	69.26 ± 1.56
audiology	66.72 ± 5.97	59.85 ± 7.43	73.53 ± 5.00	70.15 ± 2.08	69.12 ± 2.08
autos	52.95 ± 3.63	68.69 ± 2.38	66.77 ± 3.06	65.16 ± 1.14	66.77 ± 1.14
Balance-scale	86.42 ± 1.52	86.47 ± 1.10	86.49 ± 1.26	87.39 ± 0.75	86.60 ± 1.50
cmc	42.36 ± 2.42	50.29 ± 2.11	49.95 ± 2.14	49.82 ± 1.12	49.50 ± 1.12
dermatology	95.23 ± 2.07	96.33 ± 1.67	96.27 ± 1.63	96.18 ± 1.93	96.82 ± 1.93
glass	44.06 ± 6.20	48.28 ± 5.58	47.85 ± 5.30	48.15 ± 3.26	48.31 ± 5.44
Heart-c	82.22 ± 2.82	81.78 ± 2.73	82.20 ± 2.78	82.64 ± 5.44	83.30 ± 3.11
Heart-h	82.05 ± 2.56	81.59 ± 3.07	81.91 ± 2.34	82.02 ± 1.59	82.13 ± 2.38
hypothyroid	93.28 ± 0.53	93.57 ± 0.48	93.53 ± 0.49	93.52 ± 1.00	93.52 ± 0.94
iris	62.05 ± 4.55	95.68 ± 1.68	95.55 ± 1.81	95.33 ± 1.57	96.22 ± 4.71
letter	31.84 ± 0.48	82.32 ± 0.32	81.90 ± 0.27	82.32 ± 0.34	82.34 ± 0.24
lymph	86.14 ± 5.19	86.14 ± 4.07	87.11 ± 4.78	83.56 ± 4.71	85.78 ± 3.14
Primary-tumor	39.70 ± 3.38	43.17 ± 4.18	45.88 ± 3.23	47.35 ± 2.77	47.35 ± 1.39
redWine	30.75 ± 2.17	53.88 ± 5.95	57.02 ± 1.76	57.00 ± 2.06	56.83 ± 1.92
segment	76.95 ± 1.34	92.59 ± 0.71	92.60 ± 0.71	92.37 ± 0.51	92.66 ± 0.61
soybean	92.84 ± 1.85	93.33 ± 1.27	93.56 ± 1.64	93.17 ± 0.00	92.68 ± 0.00
splice	91.94 ± 0.57	93.43 ± 0.48	92.87 ± 0.58	94.10 ± 0.96	93.98 ± 0.59
vehicle	65.61 ± 3.57	72.96 ± 3.03	73.19 ± 2.78	73.35 ± 3.06	72.99 ± 5.57

Table 4. (*Continued*)

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
vowel	16.96 ± 1.04	64.93 ± 3.30	64.61 ± 3.30	64.75 ± 0.95	64.01 ± 1.90
waveform	80.11 ± 0.97	86.78 ± 1.00	86.71 ± 0.94	86.32 ± 0.52	86.53 ± 0.38
whitewine	04.87 ± 1.42	50.91 ± 3.41	52.41 ± 1.15	52.37 ± 1.49	52.37 ± 1.83
zoo	93.33 ± 5.21	95.33 ± 3.58	95.16 ± 3.80	92.26 ± 4.56	93.23 ± 4.56
Average	**77.1 ± 2.71**	**77.10 ± 2.69**	**77.98 ± 2.31**	**77.73 ± 1.77**	**77.88 ± 2.06**

Table 5. Test accuracies (%) using Logistic as the base learner

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
anneal	98.59 ± 0.58	99.10 ± 0.56	98.81 ± 0.57	98.81 ± 0.52	98.96 ± 0.52
Anneal-orig	88.88 ± 1.51	89.93 ± 2.11	88.26 ± 1.73	88.93 ± 0.52	89.04 ± 3.93
arrhythmia	46.59 ± 3.19	57.93 ± 4.62	53.75 ± 4.52	61.69 ± 1.56	63.24 ± 1.04
audiology	70.90 ± 5.28	74.33 ± 3.84	73.09 ± 3.61	75.00 ± 3.12	75.74 ± 3.12
autos	63.28 ± 4.10	70.49 ± 6.18	64.03 ± 5.84	66.94 ± 0.00	69.19 ± 1.14
Balance-scale	85.99 ± 1.12	87.86 ± 2.26	87.93 ± 2.29	88.03 ± 0.00	87.87 ± 0.75
cmc	50.41 ± 2.07	50.34 ± 1.84	50.90 ± 1.92	51.00 ± 1.28	50.68 ± 1.28
dermatology	92.75 ± 2.54	97.34 ± 1.64	96.45 ± 1.74	97.09 ± 1.93	97.55 ± 1.93
glass	58.13 ± 5.45	59.22 ± 3.49	58.46 ± 4.10	60.31 ± 2.18	61.08 ± 4.35
Heart-c	82.78 ± 2.24	82.33 ± 2.94	82.86 ± 2.33	83.19 ± 3.32	82.86 ± 3.89
Heart-h	82.95 ± 2.98	81.82 ± 3.86	83.15 ± 2.95	81.69 ± 1.59	82.70 ± 1.59
hypothyroid	95.31 ± 0.90	97.20 ± 0.62	96.64 ± 0.65	97.17 ± 1.25	97.04 ± 1.19
iris	95.45 ± 2.83	94.09 ± 2.44	94.00 ± 3.15	93.56 ± 0.00	94.22 ± 1.57
letter	72.38 ± 0.53	78.05 ± 0.33	77.45 ± 0.36	77.45 ± 0.13	77.48 ± 0.08
lymph	78.41 ± 7.52	80.68 ± 5.49	77.33 ± 6.61	84.00 ± 3.14	83.78 ± 6.29
Primary-tumor	41.29 ± 4.48	35.74 ± 4.61	37.16 ± 4.77	39.12 ± 1.39	38.73 ± 4.85
redWine	58.02 ± 1.91	58.10 ± 1.48	58.42 ± 1.97	57.98 ± 2.21	58.15 ± 2.65
segment	92.28 ± 0.43	95.58 ± 0.36	95.02 ± 0.73	95.47 ± 0.51	95.56 ± 0.71
soybean	91.57 ± 2.18	92.21 ± 1.52	91.46 ± 2.69	92.39 ± 0.69	93.32 ± 0.34
splice	90.76 ± 1.58	91.96 ± 0.74	89.62 ± 0.94	85.55 ± 2.36	85.24 ± 1.70
vehicle	78.70 ± 2.69	79.88 ± 1.83	79.41 ± 2.18	79.45 ± 0.56	79.53 ± 2.23
vowel	63.41 ± 3.20	83.07 ± 2.51	78.72 ± 2.79	78.82 ± 3.10	80.00 ± 5.24
waveform	87.04 ± 1.14	86.78 ± 1.35	86.98 ± 1.18	86.61 ± 0.90	86.84 ± 0.66
whitewine	53.89 ± 1.22	53.47 ± 0.95	53.82 ± 0.88	53.84 ± 0.87	53.68 ± 0.58
zoo	94.00 ± 3.78	95.00 ± 3.24	90.00 ± 4.42	86.45 ± 6.84	87.42 ± 6.84
Average	**76.55 ± 2.60**	**78.90 ± 2.43**	**77.75 ± 2.60**	**78.42 ± 1.71**	**78.79 ± 2.34**

In order to see clearly and to show the general knowledge of the performance of the methodologies, Table 6 integrated the average accuracy and standard deviation together from Tables 2 through 5 (the bottom row of each table).

Table 7 summarizes the comparisons among the three methodologies (OAA, OAO, AAO, and its variants AAOvsOAA and AAOvsOAO) via two-tailed t-test with a confidence level 90%. Note that AAO has two variants (AAOvsOAA and AAOvsOAO). There is no difference between them, except the number of bagging iterations. This is because they are created for making fair comparisons for AAO against OAA and OAO respectively. Thus, we only summarized the comparisons for AAOvsOAA and its opponent OAA. We did the same for AAOvsOAO against its opponent OAO only.

Table 6. Average accuracies (%) of all the approaches with the four algorithms over the 25 datasets

	OAA	OAO	AAO	AAOvsOAA	AAOvsOAO
J48	77.32 ± 1.81	79.65 ± 2.79	78.02 ± 2.68	80.69 ± 1.77	81.31 ± 1.47
NaiveBayes	73.49 ± 3.04	73.40 ± 2.81	73.12 ± 2.57	73.63 ± 2.26	73.66 ± 2.31
SMO	66.67 ± 2.71	77.10 ± 2.69	77.98 ± 2.31	77.73 ± 1.77	77.88 ± 2.06
Logistic	76.55 ± 2.60	78.90 ± 2.43	77.75 ± 2.60	78.42 ± 1.71	78.79 ± 2.34

Table 7. Summary comparisons (#wins/#ties/#loses) among the methodologies with the four algorithms over the 25 datasets

		OAO	AAO	AAOvsOAA	AAOvsOAO
J48	OAA	1/12/12	3/16/6	1/11/13	
	OAO		10/15/0		1/16/8
NaiveBayes	OAA	1/20/4	2/20/3	1/20/04	
	OAO		1/24/0		1/24/0
SMO	OAA	1/11/13	0/11/14	0/12/13	
	OAO		2/22/1		0/21/4
Logistic	OAA	1/14/10	3/15/7	2/14/9	
	OAO		9/16/0		4/18/3
Summation	**OAA**	**4/57/39**	**8/62/30**	**4/57/39**	
	OAO		**22/77/1**		**6/79/15**

From Table 6, the average accuracy over the 25 multiclass datasets shows that OAA performs worse than AAO for J48, SMO, and Logistic, although it performs a tiny better than AAO for NaiveBayes.

The statistic analysis of the experimental results on each dataset in Table 7 completely supports this. From Table 7, we can see that the comparison result of OAA against AAO using J48 as the base learner is (3/16/6). That is, OAA only wins on three datasets, but loses on six datasets, ties on the rest datasets. When using NaiveBayes instead of J48, OAA only wins AAO on two datasets, loses to AAO on three datasets, ties on the rest. When using SMO instead, OAA has no wins, but loses to AAO on 14 datasets, ties on the rest. When using Logistic instead, OAA only wins AAO on three datasets, loses to AAO on seven datasets, ties on the rest. If we sum the #wins/#ties/#loses for OAA and AAO over the four base learners, we can see OAA only wins AAO eight times, but loses to AAO 30 times, and ties with AAO 62 times, in total.

Note that OAA has the advantage in comparing with AAO directly. It creates n models (n is the number of classes in a dataset). If we further compare it with AAO with bagging n iterations, i.e., the variant AAOvsOAA, Table 6 shows the average accuracy of AAOvsOAA over the 25 datasets is 80.69%, much higher than 77.32% achieved by OAA, when using J48 as the base learner. Table 7 shows that when J48 is the base learner, OAA only wins AAOvsOAA on one dataset, but loses to AAOvsOAA on 13 datasets, and ties on the rest 16 datasets. When using NaiveBayes instead, both OAA and AAOvsOAA perform close to each other (73.49% vs. 73.63%). Table 7 shows that OAA only wins AAO on one datasets, loses to AAOvsOAA on four datasets, ties on the rest. When using SMO instead, OAA has no wins, but loses to AAOvsOAA on 13 datasets, ties on the rest. When using Logistic instead, OAA only wins AAOvsOAA on two datasets, loses to AAO on nine datasets, ties on the rest. If we sum the #wins/#ties/#loses for OAA and AAOvsOAA over the four base learners, we can see OAA only wins AAOvsOAA four times, but loses to AAOvsOAA 39 times, and ties with AAOvsOAA 57 times, in total.

All in all, our experimental results clearly show that OAA is not a good approach for multiclass classification algorithms.

Now let us analyze the performance of OAO. From Table 6, the average accuracy over the 25 multiclass datasets shows that OAO performs better than AAO for J48 and Logistic, although it performs a tiny better than AAO for NaiveBayes. But it loses to AAO for SMO.

The statistic analysis of the experimental results on each dataset in Table 7 also completely supports this. From Table 7, when using J48 as the base learner, OAO wins AAO on ten datasets, and ties on the rest. When using Logistic instead, OAO still wins AAO on nine datasets, and ties on the rest. That is, OAO never loses to OAA when using J48 and Logistic as the base learners. However, using NaiveBayes or SMO instead, OAO almost always ties with AAO, only winning on one or two datasets. If we sum the #wins/#ties/#loses for OAO and AAO over the four base learners, we can see OAO wins AAO 22 times, but loses to AAO only once, and ties with AAO 77 times, in total.

Again, note that OAO has the advantage in comparing with AAO. It creates $\frac{n(n-1)}{2}$ models (n is the number of classes in a dataset). If we further compare it with AAO with bagging $\frac{n(n-1)}{2}$ iterations, i.e., the variant AAOvsOAO, Table 6 shows the average accuracy of AAOvsOAO over the 25 datasets is 81.31%, higher than 79.65% achieved by OAO, when using J48 as the base learner. When using other three algorithms (NaiveBayes, SMO, and Logistic) as the base learner respectively, both perform similarly, one's a tiny higher than the other's. Table 7 completely supports the analysis based on Table 6. From Table 7, we can see that OAO almost always ties with AAOvsOAO, using either NaiveBayes or Logistic as the base learner. AAOvsOAO does perform better than OAO on eight datasets when using J48 as the base learner, and perform better than OAO on four datasets when using SMO. If we sum the #wins/#ties/#loses for OAO and AAOvsOAO over the four base learners, we can see OAO only wins AAOvsOAO six times, but loses to AAOvsOAO 15 times, and ties with AAOvsOAO 79 times, in total.

In summary, our experimental results clearly show that OAO is a proper approach for improving the performance of multiclass classification algorithms. However, the benefit of OAO is from the ensemble, since it creates $\frac{n(n-1)}{2}$ models. If we do bagging with the same number of iterations, OAO loses its advantage and even performs worse than AAO (i.e. AAOvsOAO).

Besides, Table 6 also shows that both AAOvsOAA and AAOvsOAO perform much better than AAO. This proves that bagging does improve the performance of all four basic algorithms. Thus, it is the approach recommended for improve the performance of multiclass classification algorithms, neither OAA nor OAO.

5 Conclusion and Future Work

This paper investigates the performance of the two well-known methodologies (OAA and OAO) for multiclass classifications. We conducted the empirical study using a wide variety of datasets with a wide range of different number of classes. Our experimental results show that OAA is not a good approach for improving the performance of multiclass classification algorithms. OAO performs much better than OAA, and also improves the performance of multiclass classification learners. However, this is due to the $\frac{n(n-1)}{2}$ models it builds, where n is the number of classes. When n is great, say 26 in the letter dataset, it has to build 325 models. If we apply bagging to these algorithms, our experimental results show that the bagging variants (AAOvsOAA and AAOvsOAO) of AAO perform better than its opponents (OAA and OAO) respectively. Thus, we can conclude that bagging is the better choice if we want to improve the performance of multiclass classifications.

We will continue to evaluate the performance of the variants of OAA and OAO, such as weighted OAA and OAO. We also notice that OAA creates imbalanced training datasets since it merges the examples of the rest classes into one. It is a potential reason why OAA performs worse than AAO. It does not obtain the benefit from the n models it builds. Another interesting topic is to investigate the potential problems for OAA and OAO if the original datasets are imbalanced. Besides, we will further study the possibility of extending the existing approaches of reducing multiclass classifications into multi-label classifications.

Acknowledgement. We thank the anonymous reviewers for the valuable comments. The work was supported by the National Science Foundation (IIS-1115417).

References

1. Mitchell, T.M.: Machine Learning, pp. 52–77. McGraw-Hill (1997)
2. Rifkin, R., Klautau, A.: In Defense of One-vs-All Classification. Journal of Machine Learning Research 5, 101–141 (2004)
3. Tsujinishi, D., Koshiba, Y., Abe, S.: Why Pairwise is Better Than One-Against-All or All-at-Once. In: IEEE International Joint Conference, pp. 693–698 (2007)
4. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science (2010), `http://archive.ica.uci.edu/ml/datasets.html?`
5. Hsu, C.W., Lin, C.W.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Tran. on Neural Networks 13(2), 415–425 (2002)
6. Moro, R.A., Auria, L.: Support Vector Machines (SVM) as a Technique for Solvency Analysis. German Institute for Economic Research (811) (2008)
7. Huang, J., Lu, J., Ling, C.X.: Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. In: Third IEEE International Conference on Data Mining, ICDM 2003, pp. 553–556 (2003)
8. Dimitoglou, G., Adams, J.A., Jim, C.M.: Comparison of the C4.5 and a Naive Bayes Classifier for the Prediction of Lung Cancer Survivability. Journal of Computing 4(8) (2012), `http://www.journalofcomputing.org/valume-4-issue-8-august-2012`
9. Anthony, Gregg, Tshilidzi: Image Classification Using SVMs: One-Against-One vs One-Against-All. In: 28th Asian Conference on Remote Sensing (2007)
10. Peng, C.J., Lee, K.L., Ingersoll, G.M.: An Introduction to Logistic Regression and Analysis Reporting. Journal of Educational Research (2002)
11. Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. Informatica 31, 249–268 (2007)
12. Fletcher, T.: Support Vector Machines Explained (2009), `http://www.tristanfletcher.co.uk`
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11 (2009)
14. Fayyad, U., Irani, K.: Multi-interval Discretization of Continuous-valued attributes for Classification Learning. In: Proceeding of Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann (1993)
15. Beygelzimer, A., Langford, J., Zadrozny, B.: Weighted One-Against-All. In: Proceeding of the 20th National Conference on Artificial Intelligence, pp. 720–725. AAAI Press (2005)
16. Allwein, E., Schapire, R., Singer, Y.: Reducing Multiclass to Binary: A unifying approach for margin classifiers. Journal of Machine Learning Research, 113–141 (2000)
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
18. Witten, I.H., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann Publishing (2011)
19. John, G.H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, pp. 338–345 (1995)
20. Hilbe, J.M.: Logistic Regression Models. Chapman & Hall/CRC Press (2009)
21. Corinna, C., Vapnik, V.: Support-Vector Networks. Machine Learning (1995)
22. Tsoumakas, G., Katakis, I.: Multi Label Classification: An Overview. International Journal of Data Warehousing and Mining 3, 1–13 (2007)

23. Sulzmann, J.-N., Fürnkranz, J., Hüllermeier, E.: On Pairwise Naive Bayes Classifiers. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 371–381. Springer, Heidelberg (2007)
24. Zadrozny, B.: Reducing Multiclass to Binary by Coupling Probability Estimates. In: Advances in Neural Information Processing Systems (2002)
25. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. The Annals of Statistics 26(2), 451–471 (1998)

EEG Feature Selection Based on Time Series Classification

Michal Vavrečka[1,2] and Lenka Lhotská[1]

[1] Biodat, Faculty of Electrical Engineering, Czech Technical University in Prague
[2] Faculty of Education, University of Southern Bohemia in Ceske Budejovice
{vavrecka,lhotska}@fel.cvut.cz

Abstract. We propose novel method of EEG signal analysis based on classification of feature time series. The algorithm classifies sequences of feature values and it calculates the error rate both for each time step and overall sequence. We compared the performance of the algorithm with a standard feature selection method based on forward inter-intra criterion. Both algorithms selected similar features. The algorithm was tested on the EEG data from 2 experiments focused on of spatial navigation and orientation. Participants traversed through the virtual tunnels and they could adopt two different reference frames (allocenctric and egocentric) to solve the task. The EEG signal was recorded within both tasks and the methods of feature extraction and both standard and timeseries selection and classification were applied to it. We identified differences between the groups of participants adopting allocentric and egocentric frames of reference in the parietal and central electrodes in right hemisphere. The novel algorithm provided more detail analysis of the EEG features compared to classic feature classification.

Keywords: frames of reference, spectral analysis, spatial navigation, EEG features.

1 Introduction

The human orientation in space is based on several frames of reference. Levinson (1996) postulated intrinsic, relative and absolute reference frames, but there are mostly two basic reference frames analyzed in the recent studies, namely allocentric and egocentric frame. The reference frame is considered as orthogonal system with the origin (deixis center) in retina, head, body or other points, objects, or array in space (McCloskey, 2001). In the egocentric frame of reference the position of objects is encoded with reference to the body of the observer or to relevant body parts. Spatial positions can also be coded in object-centered coordinates that are independent of the observer's current position. This frame of reference is referred to as allocentric and it is constituted by object-to-object relations (it refers to a framework that is independent from the subject's position).

The recent EEG studies in this area are based on the simulation of the virtual environments and the measurement of the subjects (egocetric and allocentric)

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 520–527, 2013.

navigation strategy. Gramann et al. (2006) recorded EEG signal and localized higher mean source activation in the BA 7 (parietal cortex) for subjects adopting egocentric frame of reference and BA 32 (anterior cyngulate gyrus) for subjects adopting allocentric frame of reference. Lin et al., (2008) found significantly different EEG frequency changes between the allocentric and egocentric users. Subjects who preferred allocentric frame of reference indicated stronger activation in occipital area (BA 17,18,19) and subjects using the egocentric reference frames showed stronger activation in parietal area (BA7). We extended this experiment to the 3D environment (horizontal and vertical navigation) in our last study (Vavrečka et al., 2012) and we identified intrahemispheric coherences in occipital-parietal and temporal-parietal areas as the most discriminative features.

The analysis was based on the classical method of feature selection. The features were individually evaluated by inter/intra distance to select 50 best features for each method. The inter/intra distance criterion (van der Heijden et al., 2004) is a distance-based class separability criterion that is a monotonically increasing function of the distance between the expectation vectors of different classes and a monotonically decreasing function of the scattering around the expectations. In the next step, we applied forward feature selection algorithms to the preselected features and we chose 5 best features that discriminate between egocentric and allocentric strategies were selected. The selected feature set was tested by quadratic classifier (see Vavrečka et al., 2012 for details) and we reached 77 % accuracy.

In this paper we would like to propose novel method of the EEG data analysis. It allows to study the signal in greater detail. We should represent experimental trials as a time series and analyze the changes of feature values within the time course. This method of analysis allows us to calculate trends in the data to compare, whether the error rates in the previous results were caused by the variance of the errors in each time step or there is constant error rate for the whole trials. The proposed classifier and its results are described in the following chapters.

2 Materials and Methods

2.1 Experiment Description

We collect the EEG data from the experiment focused on spatial frames of reference (see Introduction). The experimental procedure is based on the tunnel task introduced in Schönebeck et al. (2001). There are traverse through a virtual tunnel presented to the participants and they were required to keep the track of their implied virtual 3D position with respect to their starting position. They were asked to point to the origin of the traverse after each trial. The subject's homing vector corresponds to the reference frame (egocetric and allocentric) he/she adopts as the navigation system (see Fig.1).

The experimental sample consisted of 30 participants. The mean age was 28.2 years. All subjects had normal or corrected-to-normal vision. They were under no medication that affects the EEG signal and were neurologically intact. After the introduction, each participant sat comfortably in front of a computer screen in a sound attenuated room and the EEG cap was applied. The EEG signal was recorded from 64 unipolar sintered Ag/AgCl EEG electrodes. There were animations of passages through a 3D virtual tunnel consisted of 3 (straight,turned and straight) segments presented to the participants. The bend of the turned passage varied between 30 and 90 degrees at interval of 30 degrees. The length of the two straight segments and the turned segment was constant. The tunnels were administered pseudo-randomly to the participants, and there were not 2 tunnels in the same direction presented to the participant consecutively.

2.1.1 Dataset

The dataset contains data from 10 subjects The participants were instructed to adopt their native strategy in 2 block of trials. We selected participants who spontaneously adopted allocetric strategy in the first block of trials. The second block consisted of similar trials but with opposite direction of movement (backward). This change resulted in spontaneous change of the strategy and some participants adopted egocentric reference frame. We selected 10 out of 30 participants that adopt natively ((above 80% of all trials) allocetric strategy in the first block of trials and egocentric strategy in second block of trials.

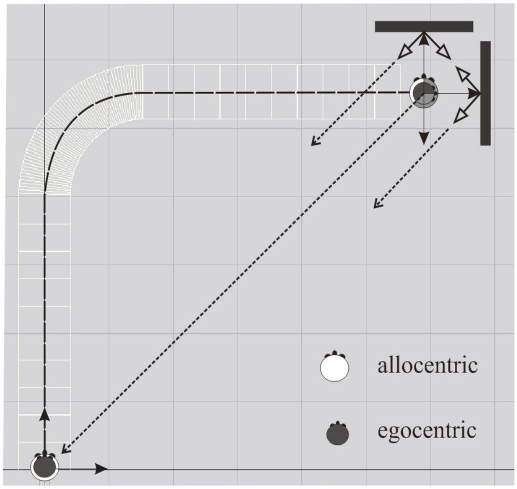


Fig. 1. Allocentric and egocentric navigation within the horizontal navigation in the virtual tunnel. The different reference frames are represented as a different head postion at the end of the tunnel (gray and white head).

2.2 EEG Preprocessing

We performed a visual inspection of each EEG signal prior to the data analysis in order to detect obvious technical and biological artifacts, and subsequently rejected these parts from further processing. We also rejected all the trials with the inconsistent strategy. Then we applied low pass, high pass and notch filter to the raw EEG data to suppress the frequencies bellow 2 Hz and above 45 Hz.

2.3 Feature Extraction

The signal was divided into the segments of constant length (1s) to calculate statistical parameters (minimum value, maximum value, mean value, standard deviation, skewness, kurtosis), mean and maximum values of the first and second derivation of the samples in the segment, wavelet and Fourier transform. The output of the last method is absolute/relative power for five EEG frequency bands, namely for delta (2 to 3 Hz), theta (3 to 7 Hz), alpha (7 to 12 Hz), beta (12 Hz to 30 Hz) and gamma (30 to 40 Hz) activities. The data were processed in PSGLab Toolbox, which was developed in our laboratory (Gerla et al., 2010).

2.4 Feature Selection and Classification

We employed PRTools (Heijdin et al., 2004) for the feature selection. First, we applied basic transforms to the data. We removed outliers and we also normalized the data. The further process of feature selection was divided into a number of steps. In the first stage two algorithms were applied for the preselection. The features were individually evaluated by inter/intra distance to select 50 best features for each method. In the next step we applied forward feature selection algorithms to the preselected features to identify a 10 best features that discriminate between particular reference frames (classes). The selected feature set was tested by a quadratic classifier to calculate error rates.

2.5 Timeseries Classification

The second type of analysis was based on the classification of time series. Feature vectors were ordered according to their time course. Then we calculate mean values for classes (reference frames) at each timestep that resulted in the mean vector for allocetric and egocentric reference frame. The size of the vector was similar to number of feature values per trial (10 timepoints standing for 10 seconds of trial). At the next stage we compared time series of specific trials with the mean vector and calculate minimum distance to the mean vectors at each time step. This allows us to identify membership of each timepoint to particular (egocentric or allocentric) class. Then we calculate the error rates for specific time points (each second of a trial is calculated separately) and mean error for whole trials. The mean error represent error rate of time series and we should compare them with the results obtained from classical feature selection described in 2.3. The pseudocode for described algorithm is as follows:

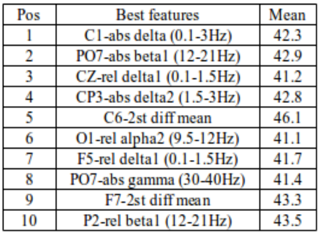
```
1. Calculate mean values for classes at each timestep
2. Find min between feature value and mean class values
3. Classify features according to minimum function
4. Compare expected and obtained classification
5. Calculate error rates for each timestep
6. Calculate mean error for whole time series
```

After this calculation we ordered the features in increasing order according their error rate and we chose the best 10 features. The both feature sets (2.4 and 2.5) were compared to identify differences between algorithms.

3 Results

In the first step we did the mathematical analysis of the EEG signal based on feature selection and classification mentioned in previous section. The analysis of allocentric and egocentric native class revealed mean error 42.6 % of the quadratic classifier for the 10 best features. The results of the classic feature selection method are summarized in Tab 1. There are mostly electrodes in parietal and central area, namely PO7 and Cz. The best features were mostly spectral bands and few statistical features. From the anatomic point of view, the best features were mostly situated in left hemisphere.

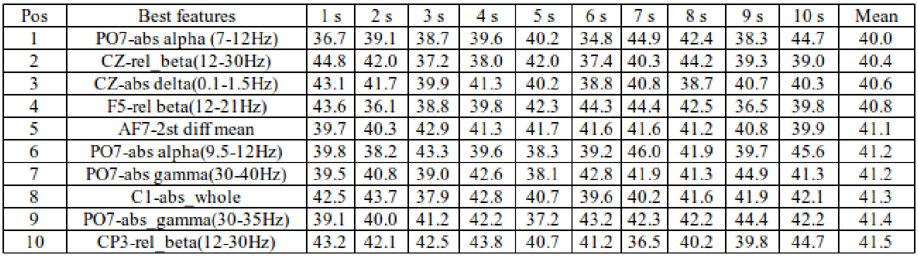
Table 1. Error rates for 10 best features based on quadratic classifier and forward feature selection



Pos	Best features	Mean
1	C1-abs delta (0.1-3Hz)	42.3
2	PO7-abs beta1 (12-21Hz)	42.9
3	CZ-rel delta1 (0.1-1.5Hz)	41.2
4	CP3-abs delta2 (1.5-3Hz)	42.8
5	C6-2st diff mean	46.1
6	O1-rel alpha2 (9.5-12Hz)	41.1
7	F5-rel delta1 (0.1-1.5Hz)	41.7
8	PO7-abs gamma (30-40Hz)	41.4
9	F7-2st diff mean	43.3
10	P2-rel beta1 (12-21Hz)	43.5

The analysis based on the proposed time series classification revealed similar features. The most discriminative electrodes were PO7 and Cz again. The mean error rate for all features were lower (40.9 %). The spectral features were the most discriminative features again. They were situated in left parietal and central area similar to previous algorithm. The detail analysis of the time series uncovered stable error rates for the time course of the experiment trials (see Tab.2). There are stable errors ranging from 36-46 %.

Table 2. Error rates for 10 best features in specific time points (1-10 seconds) and mean error of time series



Pos	Best features	1 s	2 s	3 s	4 s	5 s	6 s	7 s	8 s	9 s	10 s	Mean
1	PO7-abs alpha (7-12Hz)	36.7	39.1	38.7	39.6	40.2	34.8	44.9	42.4	38.3	44.7	40.0
2	CZ-rel_beta(12-30Hz)	44.8	42.0	37.2	38.0	42.0	37.4	40.3	44.2	39.3	39.0	40.4
3	CZ-abs delta(0.1-1.5Hz)	43.1	41.7	39.9	41.3	40.2	38.8	40.8	38.7	40.7	40.3	40.6
4	F5-rel beta(12-21Hz)	43.6	36.1	38.8	39.8	42.3	44.3	44.4	42.5	36.5	39.8	40.8
5	AF7-2st diff mean	39.7	40.3	42.9	41.3	41.7	41.6	41.6	41.2	40.8	39.9	41.1
6	PO7-abs alpha(9.5-12Hz)	39.8	38.2	43.3	39.6	38.3	39.2	46.0	41.9	39.7	45.6	41.2
7	PO7-abs gamma(30-40Hz)	39.5	40.8	39.0	42.6	38.1	42.8	41.9	41.3	44.9	41.3	41.2
8	C1-abs_whole	42.5	43.7	37.9	42.8	40.7	39.6	40.2	41.6	41.9	42.1	41.3
9	PO7-abs_gamma(30-35Hz)	39.1	40.0	41.2	42.2	37.2	43.2	42.3	42.2	44.4	42.2	41.4
10	CP3-rel_beta(12-30Hz)	43.2	42.1	42.5	43.8	40.7	41.2	36.5	40.2	39.8	44.7	41.5

We did also basic visualization of the time series in Fig.2. The thick lines stand for mean vectors for specific classes (reference frames), namely red for egocentric and blue for allocentric strategy. The dots represents feature values at specific time and the color of dot stands for allocentric or egocentric strategy (class) based on participant responses. Although we visualized the best feature, there is a overlap between classes, that results in high error rate (40% in average).

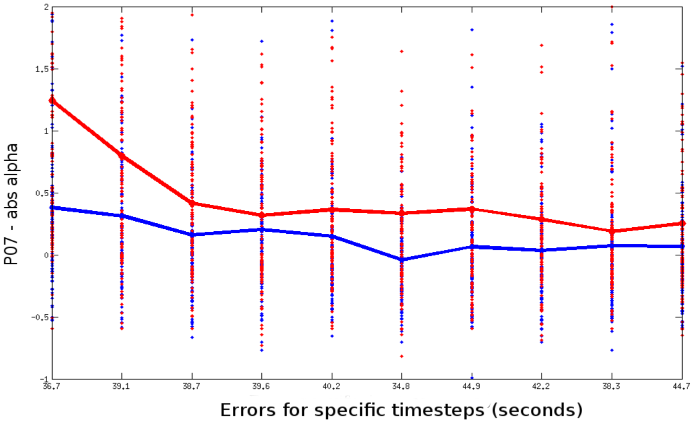


Fig. 2. Mean feature vectors for allocentric (blue) and egocetric (red) reference frame

The high error rates in the group analysis are often caused by interindividual differences in EEG signal that result in high error of its features. Hence, we did also the analysis of the separate subjects. The errors for specific subject were lower and ranged from 15-25%. We visualized the best feature of the subject n.23 with the 18.3 % mean error.

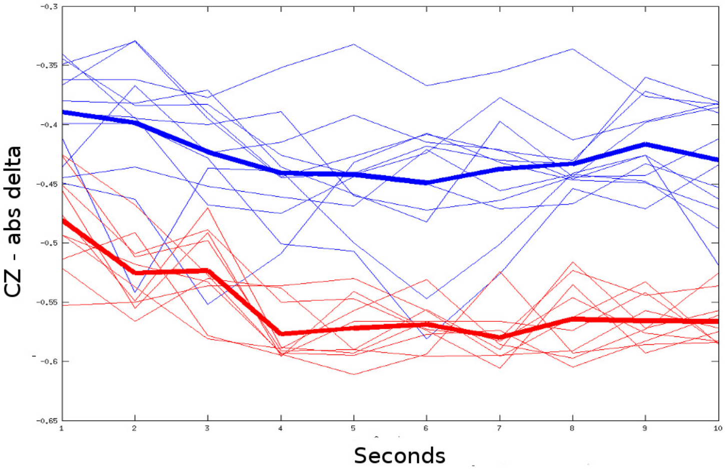


Fig. 3. Mean feature vectors (thick lines) for both reference frames (classes) in sub n.23. The thin lines stand for specific trials.

4 Discussion

4.1 Behavioral Data

We are going to briefly interpret behavioral results. The previous studies were mostly focused on the group difference between allocentric and egocentric strategy in one condition. The previous studies proved the interindividual difference in the EEG signal for allocentric and egocentric strategy but there is not study focused on intraindividual EEG differences of mentioned strategies based on group analysis.

The behavioral results uncovered the switch between forward and backward navigation in the native conditions. Previous studies (Vavrečka et al., 2012; Gramann et al., 2012) revealed also the alteration between reference frames in horizontal and vertical navigation. Hence the concept of reference frame nativeness has to be investigated in greater detail. The native reference frames are context dependent, as we are able to change participants navigation strategy based on the way of stimuli presentation (horizontal vs. vertical, forward vs. backward).

4.2 Classifier Comparison

The comparison of classical and novel algorithm led us to the conclusion, that our method suitable for EEG data analysis. There were similar results for both classic method of feature selection and proposed classifier. Moreover the novel method offers the ability to analyze EEG features in greater detail. It allows us to study the trends in the feature changes and to determine, whether the feature value changes over time or it is stable within whole trial. This is desirable especially for the experiment design with high variability in the participant's responses. We are able to detect inter trial variability and also localize important time points within the wime course. There are more detail results that should foster the interpretation and identify possible errors caused by variance in feature changes over time. The extension of the algorithms offers ability to compare and quantify trends in the data.

5 Conclusion

The novel algorithm for the analysis of feature time series was applied to the EEG data in the area of spatial cognition. The comparison with classic algorithm resulted in similar results. Moreover the novel classifier is able to calculate errors for specific time points and to capture trends in the data.

Acknowledgments. This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic (under project No. MSM 6840770012 "Transdisciplinary Biomedical Engineering Research II"), GACR grant No. GPP407/11/P696 and EU project RPP JU No. CZ.1.07/2.3.00/30.0049

References

[1] Gerla, V., Djordjevic, V., Lhotská, L., Krajča, V.: PSGLab Matlab Toolbox for Polysomnographic Data Processing: Development and Practical Application. In: Proceedings of 10th IEEE International Conference on Information Technology and Applications in Biomedicine (2010)

[2] Gramann, K., Müller, H.J., Eick, E., Schönebeck, B.: Empirical evidence for separable spatial representations in a virtual navigation task. Journal of Experimental Psychology: Humam Perception and Performance 31, 1199–1223 (2005)

[3] Gramann, K., Müller, H., Schönebeck, B., Debus, G.: The neural basis of egocentric and allocentric reference frames in spatial navigation: Evidence from spatio-coupled current density reconstruction. Brain Research 1118, 116–129 (2006)

[4] Gramann, K., Wing, S., Jung, T.-P., Viirre, E., Riecke, B.: Switching spatial reference frames for yaw and pitch navigation. Spatial Cognition and Computation 12, 159–194 (2012)

[5] Heijden, F.V.D., Duin, R., Ridder, D.D., Tax, D.M.J.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. John Wiley & Sons (2004)

[6] Levinson, S.C.: Frames of reference and Molyneux's question: Cross-linguistic evidence. In: Bloom, P., Peterson, M., Nadel, L., Garrett, M. (eds.) Language and Space, pp. 109–169. MIT press, Cambridge (1996)

[7] Lin, T.C.: Differences in EEG Dynamics between the use of Allocentric and Egocentric reference frames during VR-based Spatial Navigation. In: CACS 2008 (2008)

[8] Lin, T.C., Chiou, T.C., Ko, L.W., Duann, J.R., Gramann, K.: EEG-based spatial navigation estimation in a virtual reality driving environment. In: Proceedings of the Ninth IEEE International Conference on Bioinformatics and Bioengineering, pp. 435–438 (2009)

[9] McCloskey, M.: Spatial representation in mind and brain. In: Rapp, B. (ed.) A Handbook of Cognitive Neuropsychology, pp. 101–132. Psychology Press, Philadelphia (2001)

[10] Schönebeck, B., Thanhaüser, J., Debus, G.: Die Tunnelaufgabe: eine Methode zur Untersuchung räumlicher Orientierungsleistungen, vol. 48, pp. 339–364 (2001)

[11] Vavrečka, M., Gerla, V., Lhotská, L., Brunovský, M.: Frames of reference and their neural correlates within navigation in a 3D environment. Visual Neuroscience 29(03), 183–191 (2012)

DCA Based Algorithms for Feature Selection in Semi-supervised Support Vector Machines

Hoai Minh Le, Hoai An Le Thi, and Manh Cuong Nguyen

Laboratory of Theoretical and Applied Computer Science - LITA EA 3097,
University of Lorraine, Ile du Saulcy, 57045 Metz, France
{minh.le,hoai-an.le-thi,manh-cuong.nguyen9}@univ-lorraine.fr

Abstract. In this paper, we develop an efficient method for feature selection in Semi-Supervised Support Vector Machine (S3VM). Using an appropriate continuous approximation of the $l_0 - norm$, we reformulate the feature selection S3VM problem as a DC (Difference of Convex functions) program. DCA (DC Algorithm), an innovative approach in nonconvex programming is then developed to solve the resulting problem. Computational experiments on several real-world datasets show the efficiency and the scalability of our method.

Keywords: Feature Selection, S3VM, DC Programming, DCA.

1 Introduction

One of the most critical problems in supervised learning is the size of training sets. If the used training dataset is small, one may face the problem of overfitting. While the collection of data can be cheap, the manual labeling for the purpose of training process is usually slow, expensive and error-prone process. To overcome this difficulty, recently, in the machine learning community, there has been an attracting increasing attention in using semi-supervised learning methods. The goal of semi-supervised classification is to build a classifier using both labeled and unlabeled data. Furthermore, it has been shown that using unlabeled data for learning improves the accuracy of classifier ([2]). In [2], the authors report that a trained SVM on a well-chosen subset performs often better than on all available instances.

On another hand, feature selection is one of the fundamental problems in machine learning. In many areas of application such as text classification, web mining, gene expression, micro-array analysis, combinatorial chemistry, image analysis, etc, data sets contain a large number of features, many of which are irrelevant or redundant. Feature selection is often applied to high dimensional data prior to classification learning. The main goal is to select a subset of features of a given data set while preserving or improving the discriminative ability of a classifier. Several feature-selection methods for SVMs have been proposed in the literature (see e.g. [3,16,19,21]).

In this work, we are interested in feature selection in the context of S3VM that can be stated as follows. Given a training set which consists of m labeled

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 528–542, 2013.

points $\{(x_i, y_i) \in \mathbb{R}^n \times \{-1, 1\}, i = 1, \ldots, m\}$ and p unlabeled points $\{x_i \in \mathbb{R}^n, i = (m+1), \ldots, (m+p)\}$. We are to find a separating hyperplane $P = \{x \mid x \in \mathbb{R}^n, x^T w = b\}$, far away from both the labeled and unlabeled points, that uses the least number of features. Naturally, using the zero-norm (l_0-norm) is the best way for feature selection. Hence, replacing the l_2 norm by the l_0-norm in the classical S3VM model, the feature selection S3VM problem can be formulated as follow:

$$\min_{w,b} \alpha \sum_{i=1}^{m} L\left(y_i(\langle w, x_i\rangle + b)\right) + \beta \sum_{i=m+1}^{m+p} L\left(|\langle w, x_i\rangle + b|\right) + \|w\|_0. \tag{1}$$

Here, the first two terms correspond to, respectively, the hingle loss function of labeled and unlabeled data points (which are weighted by penalty parameters $\alpha > 0$ and $\beta > 0$), while the third term deals with the sparsity. Usually, in classical SVM one uses the hingle loss function $L(u) = \max\{0, 1-u\}$ which is convex. On contrary, the problem (1) is nonconvex, due to, firstly, the nonconvexity of the second term. A variety of optimization approaches have been recently developed for the S3VM (see e.g. [2,4,5,6,7,9,24]). There are two broad strategies for solving S3VM: the exact methods via Mixed Integer Programming ([2]), branch and Bound algorithm ([5]), deterministic annealing ([24]) and the approximate methods such as self-labeling heuristic S^3VM^{light} ([9]), gradient descent ([4]), DCA-S3VM ([7]). For a more complete review of S3VM methods, the reader is referred to [6] and the references therein. Exact methods are not available for massive data sets in real applications (*high* dimension and *large* data set). The scalability is a great challenge in semi supervised learning. Thus, major efforts have focused on efficient local algorithms.

While S3VM has been widely studied, there exist few methods in the literature for feature selection in S3VM. Due to the discontinuity of the l_0 term that represents the sparsity of the vector w, we are faced with "double" difficulties in (1) (it is well known that the problem of minimizing the zero-norm is NP-Hard ([1])).

We investigate in this work an efficient nonconvex programming approach to solve the feature selection S3VM problem (1). Our method is based on DC (Difference of Convex functions) programming and DCA (DC Algorithms) that were introduced by Pham Dinh Tao in a preliminary form in 1985. They have been extensively developed since 1994 by Le Thi Hoai An and Pham Dinh Tao (see [11,13] and the references therein). In the last decade, a variety of works in Machine Learning based on DCA have been developed. The efficiency and the scalability of DCA have been proved by various papers (see e.g. [7,10,14,15,16,20,18,22] and the list of reference in [17]). These successes of DCA motived us to investigate it for solving the hard problem (1).

The remainder of the paper is organized as follows. DC programming and DCA are briefly presented in Section 2 while Section 3 is devoted to the development of DCA for solving the feature selection S3VM problem (1). Finally, computational results are reported in the last section.

2 Outline of DC Programming and DCA

DC programming and DCA constitute the backbone of smooth/nonsmooth nonconvex programming and global optimization. They address the problem of minimizing a function f which is the difference of two convex functions on the whole space $\mathbb{R}^d$ or on a convex set $C \subset \mathbb{R}^d$. Generally speaking, a DC program is an optimisation problem of the form :

$$\alpha = \inf\{f(x) := g(x) - h(x) : x \in \mathbb{R}^d\} \qquad (P_{dc})$$

where g, h are lower semi-continuous proper convex functions on $\mathbb{R}^d$. Such a function f is called a DC function, and $g - h$ a DC decomposition of f while g and h are the DC components of f. The convex constraint $x \in C$ can be incorporated in the objective function of (P_{dc}) by using the indicator function on C denoted by χ_C which is defined by $\chi_C(x) = 0$ if $x \in C$, and $+\infty$ otherwise:

$$\inf\{f(x) := g(x) - h(x) : x \in C\ \} = \inf\{\chi_C(x) + g(x) - h(x) : x \in \mathbb{R}^d\}.$$

A convex function θ is called *convex polyhedral* if it is the maximum of a finite family of affine functions, i.e.

$$\theta(x) = \max\{\langle a_i, x\rangle + b : i = 1, ...p\}, a_i \in \mathbb{R}^d.$$

Polyhedral DC optimization occurs when either g or h is polyhedral convex. This class of DC optimization problems, which is frequently encountered in practice, enjoys interesting properties (from both theoretical and practical viewpoints) concerning local optimality and the convergence of DCA ([13]).

Let

$$g^*(y) := \sup\{\langle x, y\rangle - g(x) : x \in \mathbb{R}^d\}$$

be the conjugate function of a convex function g. Then, the following program is called the dual program of (P$_{dc}$):

$$\alpha_D = \inf\{h^*(y) - g^*(y) : y \in \mathbb{R}^d\}. \qquad (D_{dc})$$

One can prove that $\alpha = \alpha_D$, and there is the perfect symmetry between primal and dual DC programs: the dual to (D_{dc}) is exactly (P_{dc}).

For a convex function θ, the subdifferential of θ at $x_0 \in \text{dom}\ \theta := \{x \in \mathbb{R}^d : \theta(x_0) < +\infty\}$, denoted by $\partial\theta(x_0)$, is defined by

$$\partial\theta(x_0) := \{y \in \mathbb{R}^d : \theta(x) \geq \theta(x_0) + \langle x - x_0, y\rangle, \forall x \in \mathbb{R}^d\}. \qquad (2)$$

The subdifferential $\partial\theta(x_0)$ generalizes the derivative in the sense that θ is differentiable at x_0 if and only if $\partial\theta(x_0) \equiv \{\bigtriangledown_x\theta(x_0)\}$. Recall the well-known property related to subdifferential calculus of a convex function θ:

$$y_0 \in \partial\theta(x_0) \Longleftrightarrow x_0 \in \partial\theta^*(y_0) \Longleftrightarrow \langle x_0, y_0\rangle = \theta(x_0) + \theta^*(y_0). \qquad (3)$$

The complexity of DC programs resides, in the lack of practical optimal globality conditions. Local optimality conditions are then useful in DC programming.

A point x^* is said to be *a local minimizer* of $g - h$ if $g(x^*) - h(x^*)$ is finite and there exists a neighbourhood $\mathcal{U}$ of x^* such that

$$g(x^*) - h(x^*) \leq g(x) - h(x), \quad \forall x \in \mathcal{U}. \tag{4}$$

The necessary local optimality condition for (primal) DC program (P_{dc}) is given by

$$\emptyset \neq \partial h(x^*) \subset \partial g(x^*). \tag{5}$$

The condition (5) is also sufficient (for local optimality) in many important classes of DC programs, for example, when (P_{dc}) is a polyhedral DC program with h being polyhedral convex function, or when f is locally convex at x^* (see [13]).

A point x^* is said to be *a critical point* of $g - h$ if

$$\partial h(x^*) \cap \partial g(x^*) \neq \emptyset. \tag{6}$$

The relation (6) is in fact the generalized KKT condition for (P_{dc}) and x^* is also called a generalized KKT point.

Based on local optimality conditions and duality in DC programming, the DCA consists in constructing of two sequences $\{x^l\}$ and $\{y^l\}$ of trial solutions of the primal and dual programs respectively, such that the sequences $\{g(x^l) - h(x^l)\}$ and $\{h^*(y^l) - g^*(y^l)\}$ are decreasing, and $\{x^l\}$ (resp. $\{y^l\}$) converges to a primal feasible solution x^* (resp. a dual feasible solution y^*) satisfying local optimality conditions:

$$x^* \in \partial g^*(y^*), \quad y^* \in \partial h(x^*). \tag{7}$$

It implies, according to (3) that $x^*$and y^* are critical points (KKT points) of $g - h$ and $h^* - g^*$ respectively.

The main idea behind DCA is to replace in the primal DC program (P_{dc}), at the current point x^l of iteration l, the second component h with its affine minorization defined by

$$h_l(x) := h(x^l) + \ \langle x - x^l, y^l \rangle, \quad y^l \in \partial h(x^l)$$

to give birth to the primal convex program of the form

$$(P_l) \quad \inf\{g(x) - h_l(x) : \ x \in \mathrm{I\!R}^n\} \Longleftrightarrow \inf\{g(x) - \langle x, y^l \rangle \ : x \in \mathrm{I\!R}^d\}$$

whose an optimal solution is taken as x^{l+1}.

Dually, a solution x^{l+1} of (P_l) is then used to define the dual convex program (D_{l+1}) by replacing in (D_{dc}) g^* with its affine minorization defined by

$$(g^*)_l(y) := g^*(y^l) + \langle y - y^l, x^{l+1} \rangle, \quad x^{l+1} \in \partial g^*(y^l)$$

to obtain

$$(D_{l+1}) \quad \inf\{h^*(y) - [g^*(y^l) + \langle y - y^l, x^{l+1} \rangle] y \in \mathrm{I\!R}^d\}$$

whose an optimal solution is taken as y^{l+1}. The process is repeated until convergence.

DCA performs so a double linearization with the help of the subgradients of h and g^*. According to relation (3) it is easy to see that the optimal solution set of (P_l) (resp. (D_{l+1})) is nothing but $\partial g^*(y^l)$ (resp. $\partial h(x^{l+1})$). Hence, we can say that DCA is an iterative primal-dual subgradient method that yields the next scheme: (starting from given $x^0 \in \text{dom}\ \partial h$)

$$y^l \in \partial h(x^l); \quad x^{l+1} \in \partial g^*(y^l), \ \forall l \geq 0. \tag{8}$$

The generic DCA scheme is shown below.

```
DCA scheme
Initialization: Let $x^0 \in \mathrm{I\!R}^d$ be a best guess, $l = 0$.
Repeat
  – Calculate $y^l \in \partial h(x^l)$
  – Calculate $x^{l+1} \in \arg\min\{g(x) - h(x^l) - \langle x - x^l, y^l \rangle : x \in \mathrm{I\!R}^d\}$   $(P_l)$
  – $l = l + 1$
Until convergence of $\{x^l\}$.
```

A deeper insight into DCA has been described in [13]. For instant it is worth to mention the main feature of DCA: DCA is constructed from DC components and their conjugates but not the DC function f itself which has infinitely many DC decompositions, there are as many DCA as there are DC decompositions. Such decompositions play an extremely critical role in determining the speed of convergence, stability, robustness, and globality of sought solutions. It is important to study various equivalent DC forms of a DC problem. This flexibility of DC programming and DCA is of particular interest from both a theoretical and an algorithmical point of view.

For a complete study of DC programming and DCA the reader is referred to [11,13].

The solution of a nonconvex program (P_{dc}) by DCA must be composed of two stages: the search of an *appropriate* DC decomposition of f and that of a *good* initial point.

We note that the convex concave procedure (CCCP) for constructing discrete time dynamical systems mentioned in [23] is nothing else than an instant of DCA reduced to smooth optimization. Likewise, the SLA (Successive Linear Approximation) algorithm developed in [3] is a version of DCA for concave minimization program.

3 Feature Selection in S3VM by DCA

3.1 DC Formulation of the Problem (1)

Assume that the m labeled points and p unlabeled points are represented by the matrix $A \in \mathrm{I\!R}^{m \times n}$ and $B \in \mathrm{I\!R}^{p \times n}$, respectively. D is a $m \times m$ diagonal matrix where $D_{i,i} = y_i, \forall i = 1, \ldots, m$. For each labeled point x_i $(i = 1, \ldots, m)$, we introduce a new variable ξ_i which represents the misclassification error. By the same way, for each unlabeled point x_i $(i = (m+1), \ldots, (m+p))$, we define r_i and s_i for two possible misclassification errors. Then, the final class of unlabeled x_i corresponds to the one that has minimal misclassification. Hence, the feature election in S3VM problem (1) can be formulated as follows:

$$\min \{F_0(w, b, \xi, r, s) := \alpha\langle e, \xi\rangle + \beta\langle e, \min\{r, s\}\rangle + \|w\|_0 : (w, b, \xi, r, s) \in K\} \tag{9}$$

where K is polyhedral convex set defined by

$$\begin{aligned} \{(w, b, \xi, r, s) : \ & D(Aw - eb) + \xi \geq e, \\ & Bw - eb + r \geq e, -Bw + eb + s \geq e, \\ & \xi, r, s \geq 0\}. \end{aligned} \tag{10}$$

In the literature, some approximations of the step vector w_* ($w_{*i} = 1$ if $w_i \neq 0$, 0 otherwise) have been proposed in the context of feature selection in SVM. The first is the concave exponential function developed by Bradley and Mangasarian in [3] where the step function w_* is approximated by:

$$w_* \simeq e - \varepsilon^{-\lambda w}, \lambda > 0, \tag{11}$$

and then the zero-norm $\|w\|_0$ is approximated by $\|w\|_0 \simeq e^T(e - \varepsilon^{-\lambda w})$. The method, called SLA (Successive Linear Approximation), proposed in [3], for solving the resulting Feature Selection concaVe minimization problem (FSV), consists of solving at each iteration one linear program (recall that SLA is a special DCA scheme applied to FSV).

In [16], another DC formulation has been investigated. The computational experiments on real-wold datasets indicate that the proposed method performs consistently well on these datasets: while it suppresses up to more than 99% of features, it can give a good classification. Moreover, the comparative results show the superiority of this approach over the two standard methods: SLA studied in [3] (this shows the nice effect of DC decompositions for DCA) as well as SVMs using l_1-norm. Motivated by the success of the approach proposed in [16], we investigate a similar DC formulation for (9). The difference here is that the second term in the objective function of (1) is nonconvex, more precisely it is concave. Expressing this term as a DC function and then combining it with the DC decomposition of the approximate term of the l_0-norm proposed in [16], we get an appropriate DC decomposition of F_0.

For $x \in \mathbb{R}$, let η be the function defined by ($\lambda > 0$)

$$\eta(x, \lambda) = 1 - e^{-\lambda|x|}. \tag{12}$$

In what follows, for a given λ, we will use $\eta(x)$ instead of $\eta(x, \lambda)$. It is clear that for moderate values of λ, one can obtain a very adequate approximation of w_*. Then the approximation of zero-norm $\|w\|_0$ is given by: $\|w\|_0 \simeq \sum_{i=1}^n \eta(w_i)$. We first express $\eta(x)$ as a DC function as follows: $\eta(x) = g(x) - h(x)$, where

$$g(x) := \lambda|x| \text{ and } h(x) := \lambda|x| - 1 + \varepsilon^{-\lambda|x|}. \tag{13}$$

The objective function of (9) can be written as

$$F_0(w,b,\xi,r,s) = G_0(w,b,\xi,r,s) - H_0(w,b,\xi,r,s)$$

where $G_0(w,b,\xi,r,s) = \alpha\langle e,\xi\rangle + \sum_{i=1}^n g(w_i)$ and $H_0(w,b,\xi,r,s) = -\beta\langle e, \min\{r,s\}\rangle + \sum\limits_{i=1}^n h(w_i)$ are convex functions (G_0 is polyhedral convex). Hence, we can recast (9) as a polyhedral DC program:

$$\min\{G_0(w,b,\xi,r,s) - H_0(w,b,\xi,r,s) : (w,b,\xi,r,s) \in K\}. \tag{14}$$

According to Section 2, determining the DCA scheme applied to (14) amounts to computing the two sequences $\{(w^l,b^l,\xi^l,r^l,s^l)\}$ and $\{(\bar{w}^l,\bar{b}^l,\bar{\xi}^l,\bar{r}^l,\bar{s}^l)\}$ such that

$$(\bar{w}^l,\bar{b}^l,\bar{\xi}^l,\bar{r}^l,\bar{s}^l) \in \partial H_0(w^l,b^l,\xi^l,r^l,s^l)$$

and $(w^{l+1},b^{l+1},\xi^{l+1},r^{l+1},s^{l+1})$ is an solution of following convex problem

$$\min\left\{\begin{array}{r}\sum\limits_{i=1}^n \max\{\lambda w_i, -\lambda w_i\} + \alpha\langle e,\xi\rangle - \langle(\bar{w}^l,\bar{b}^l,\bar{\xi}^l,\bar{r}^l,\bar{s}^l),(w,b,\xi,r,s)\rangle \\ :(w,b,\xi,r,s)\in K\end{array}\right\}. \tag{15}$$

A subgradient of H_0 can be computed as follows: $\partial H_0(w,b,\xi,r,s) = (\bar{w},0,0,\bar{r},\bar{s})$ where

$$\bar{w}_i = \lambda(1-\epsilon^{-\lambda w_i}) \text{ if } w_i \geq 0,\ -\lambda(1-\epsilon^{\lambda w_i}) \text{ if } w_i < 0, \quad \forall i = 1,\ldots,n, \tag{16}$$

$$\bar{r}_i = -\beta \text{ if } r_i < s_i,\ 0 \text{ if } r_i > s_i,\ -\beta\mu \text{ if } r_i = s_i, \quad \forall i = 1,\ldots,p, \tag{17}$$

and

$$\bar{s}_i = 0 \text{ if } r_i < s_i,\ -\beta \text{ if } r_i > s_i,\ -\beta(1-\mu) \text{ if } r_i = s_i, \quad \forall i = 1,\ldots,p. \tag{18}$$

with ($\mu \in [0,1]$). Furthermore, solving (15) amounts to solving the following linear program:

$$\left\{\begin{array}{l}\min\ \langle e,t\rangle + \alpha\langle e,\xi\rangle - \langle(\bar{w}^l,\bar{b}^l,\bar{\xi}^l,\bar{r}^l,\bar{s}^l),(w,b,\xi,r,s)\rangle, \\ s.t.\ \ (w,b,\xi,r,s)\in K \\ \qquad t_i \geq \lambda w_i,\ t_i \geq -\lambda w_i, \forall i = 1,\ldots,n,\end{array}\right. \tag{19}$$

Therefor, DCA applied to (14) can be described as follows:

S3VM-l_0-DCA

Initialization Let τ be a tolerance sufficiently small, set $l = 0$.
Choose $(w^0,b^0,\xi^0,r^0,s^0) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^p$.

Repeat

- Compute $\partial H_0(w^l,b^l,\xi^l,r^l,s^l)$ via (16), (17) and (18).
- Solve the linear program (19) to obtain $(w^{l+1},b^{l+1},\xi^{l+1},r^{l+1},s^{l+1})$.
- $l = l+1$.

Until $\|(w^{l+1}, b^{l+1}, \xi^{l+1}, r^{l+1}, s^{l+1}) - (w^l, b^l, \xi^l, r^l, s^l)\| \leq \tau(\|(w^l, b^l, \xi^l, r^l, s^l)\| + 1)$.

Theorem 1. *(Convergence properties of Algorithm* ***S3VM-**l_0**-DCA****)*

(i) ***S3VM-**l_0**-DCA*** *generates a sequence* $\{(w^l, b^l, \xi^l, r^l, s^l)\}$ *such that the sequence* $\{F_0(w^l, b^l, \xi^l, r^l, s^l)\}$ *is monotonously decreasing.*

(ii) The sequence $\{(w^l, b^l, \xi^l, r^l, s^l)\}$ *converges to* $(w^*, b^*, \xi^*, r^*, s^*)$ *after a finite number of iterations.*

(iii) The point $(w^*, b^*, \xi^*, r^*, s^*)$ *is a critical point of the objective function* F_0 *in Problem (14).*

(iv) The point $(w^*, b^*, \xi^*, r^*, s^*)$ *is almost always a local minimizer of Problem (14).*

Proof: (i) and (iii) are direct consequences of the convergence properties of general DC programs while (ii) is a convergence property of a DC polyhedral program. Only (iv) needs a proof.

We first note that since G_0 is a polyhedral convex function, so is G_0^*. Hence the dual DC program of (14) is a polyhedral DC program and the relation

$$\emptyset \neq G_0^*(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*) \cap H_0^*(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$$

is a sufficient local optimality condition. This relation is verified at $(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$, the limit point of the sequence $(\bar{w}^k, \bar{b}^k, \bar{\xi}^k, \bar{r}^k, \bar{s}^k)$ generated by DCA, when G_0^* is differentiable at $(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$. Since G_0^* is a polyhedral function, it is almost always differentiable. Therefore, $(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$ is almost always a local solution to the dual DC program of (14).

On another hand, according to the property of transportation of local minimizers in DC programming we have the following (see [13]): let $(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$ be a local solution to the dual program of (14) and $(w^*, b^*, \xi^*, r^*, s^*) \in \partial G_0^*(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$. If H_0 is differentiable at $(w^*, b^*, \xi^*, r^*, s^*)$ then $(w^*, b^*, \xi^*, r^*, s^*)$ is a local solution of (14). Combining this property with the facts that $(\bar{w}^*, \bar{b}^*, \bar{\xi}^*, \bar{r}^*, \bar{s}^*)$ is almost always a local solution to the dual DC program of (14) and H_0 is differentiable almost everywhere (except when $r_i^* = s_i^*$), we conclude that $(w^*, b^*, \xi^*, r^*, s^*)$ is almost always a local minimizer to Problem (14). The proof is then complete. ■

4 Computational Experiments

4.1 Datasets

Numerical experiments were performed on several real-world datasets taken from UCI Machine Learning Repository and NIPS 2003 Feature Selection Challenge. The information about data sets is summarized in Table 1 (#Att is the number of features while #Train(resp. #Test) stands for the number of points in training set (resp. test set).

Table 1. Datasets

Dataset	#Att	#Train	#Test
W60	32	380	189
Ionos	34	234	117
Spam	57	1725	576
Advert	1558	2458	821
Gisette	5000	6000	1000
Leukem	7129	38	34
Arcene	10000	100	100

4.2 Concurrent Algorithms

We compare **S3VM-l_0-DCA** with the feature selection S3VM using $l_1 - norm$, namely

$$\min \{F_1(w,b,\xi,r,s) := \alpha\langle e,\xi\rangle + \beta\langle e, \min\{r,s\}\rangle + \|w\|_1 : (w,b,\xi,r,s) \in K\}\,. \tag{20}$$

Once again, problem (20) can be formulated as a DC program and then solved by DC programming and DCA. Let

$G_1(w,b,\xi,r,s) = \alpha\langle e,\xi\rangle + \|w\|_1$ and $H_1(w,b,\xi,r,s) = -\beta\langle e, \min\{r,s\}\rangle$.
The objective function of (20) can be written as:

$$F_1(w,b,\xi,r,s) = G_1(w,b,\xi,r,s) - H_1(w,b,\xi,r,s).$$

Obviously, $G_1(w,b,\xi,r,s)$ and $H_1(w,b,\xi,r,s)$ are convex polyhedral functions. Therefore (20) is a polyhedral DC program. DCA applied to (20) can be described as follows:

S3VM-l_1-DCA
Initialization Let τ be a tolerance sufficiently small, set $l = 0$.
Choose $(w^0, b^0, \xi^0, r^0, s^0) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^p$.
Repeat

- Compute $\partial H_1(w^l, b^l, \xi^l, r^l, s^l) = (0,0,0,\bar{r},\bar{s})$ with $\bar{r}$ (resp. $\bar{s}$) given as in (17) (resp. (18)).
- Solve the linear following program to obtain $(w^{l+1}, b^{l+1}, \xi^{l+1}, r^{l+1}, s^{l+1})$:

$$\begin{cases} \min\ \langle e,t\rangle + \alpha\langle e,\xi\rangle - \langle(\bar{w}^l, \bar{b}^l, \bar{\xi}^l, \bar{r}^l, \bar{s}^l), (w,b,\xi,r,s)\rangle, \\ s.t.\ \ (w,b,\xi,r,s) \in K, \\ \qquad t_i \geq w_i,\ t_i \geq -w_i \quad \forall i = 1,\ldots,n. \end{cases} \tag{21}$$

- $l = l + 1$.

Until $\|(w^{l+1}, b^{l+1}, \xi^{l+1}, r^{l+1}, s^{l+1}) - (w^l, b^l, \xi^l, r^l, s^l)\| \leq \tau(\|(w^l, b^l, \xi^l, r^l, s^l)\| + 1)$.

For studying the nice effect of DC decomposition we also compare our algorithm **S3VM-l_0-DCA** with DCA applied on the same problem (9) but with another DC formulation ([3]). By introducing a non-negative variable $v \geq 0$ and the constraint relation $-v \leq w \leq v$ ([3]), the problem (9) can be reformulated as follow:

$$\min \left\{ \begin{array}{r} F_2(w,b,\xi,r,s,v) := \alpha\langle e,\xi\rangle + \beta\langle e, \min\{r,s\}\rangle + e^T(e - \epsilon^{-\lambda v}) \\ : (w,b,\xi,r,s) \in K' \end{array} \right\}. \tag{22}$$

where $K' = K \cap \{-v \leq w \leq v\}$. The objective function of (22) can be written as:

$$F_2(w,b,\xi,r,s,v) = G_2(w,b,\xi,r,s,v) - H_2(w,b,\xi,r,s,v)$$

where
$G_2(w,b,\xi,r,s,v) = \chi_{K'}$ and $H_2(w,b,\xi,r,s,v) = -F_2(w,b,\xi,r,s,v)$.

It is clear that $G_2(w,b,\xi,r,s,v)$ and $H_2(w,b,\xi,r,s,v)$ are convex functions. Therefore (22) is a DC program. Below is the description of DCA applied to (22):

```
S3VM-$l_0$-DCA-2
Initialization Let $\tau$ be a tolerance sufficiently small, set $l = 0$.
Choose $(w^0, b^0, \xi^0, r^0, s^0, v^0) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^n$.
Repeat
  – Compute $\partial H_2(w^l, b^l, \xi^l, r^l, s^l, v^l) = (0, 0, \bar{\xi}, \bar{r}, \bar{s}, \bar{v})$ with $\bar{r}$ (resp. $\bar{s}$) given as
    in (17) (resp. (18)) and
        $$\bar{\xi} = -\alpha e, \quad \bar{v}_i = \lambda \epsilon^{-\lambda v_i} \quad \forall i = 1, \ldots, n.$$
  – Solve the linear following program to obtain $(w^{l+1}, b^{l+1}, \xi^{l+1}, r^{l+1}, s^{l+1}, v^{l+1})$:
        $$\min \left\{ -\langle (\bar{w}^l, \bar{b}^l, \bar{\xi}^l, \bar{r}^l, \bar{s}^l, \bar{v}^l), (w,b,\xi,r,s,v) \rangle : (w,b,\xi,r,s,v) \in K' \right\}. \tag{23}$$
  – $l = l + 1$.
Until $\|(w^{l+1}, b^{l+1}, \xi^{l+1}, r^{l+1}, s^{l+1}, v^{l+1}) - (w^l, b^l, \xi^l, r^l, s^l, v^l)\| \leq$
  $\tau(\|(w^l, b^l, \xi^l, r^l, s^l, v^l)\| + 1)$.
```

4.3 Experimental Results

All algorithms clustering was implemented in the Visual C++ 2008, and performed on a PC Intel i5 CPU650, 3.2 GHz of 4GB RAM. The starting point of DCA is chosen as follows: ξ^0, r^0, s^0 are set to zero and w^0 and b^0 are randomly chosen. We stop DCA with the tolerance $\tau = 10^{-6}$. The non-zero elements of w are determined according to whether $|w_i|$ exceeds a small threshold (10^{-6}). λ is set to 5 as proposed in [16]. For all experiments, we perform each algorithm 10 times from 10 random starting points and then report the average result of 10 executions.

We are interested in the classification error and the sparsity of obtained solution as well as the rapidity of the algorithms. We measure the classification error via two criteria : the maximum sum (MS) and the accuracy (ACC) which are defined as follows:

$$MS = (SE + SP)/2, \quad ACC = (TP + TN)/(TP + TN + FP + FN).$$

where TP and TN denote true positives and true negatives while FP and FN represent false positives and false negatives. $SE = TP/(TP + FN)$ (resp. $SP = TN/(TN + FP)$) is the correctness rate of positive class (resp. negative class). The sparsity of solution is determined by the number of selected features (SF) while the rapidity of algorithms is measured by the CPU time in seconds.

Experiment 1: In this experiment, we compare the effectiveness of three algorithms with a fixed percentage of unlabeled points (60% of training set will be set to be unlabeled points). The comparative results are reported in Table 2 and Figure 1 - 3.

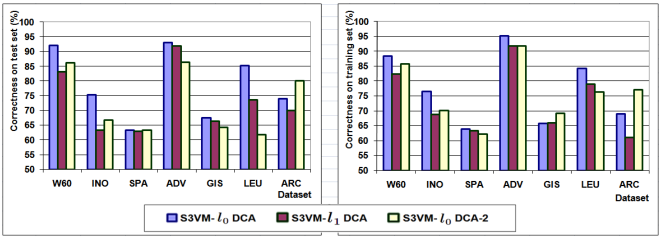


Fig. 1. Accuracy (ACC) of three algorithms on training (right) and test (left) set

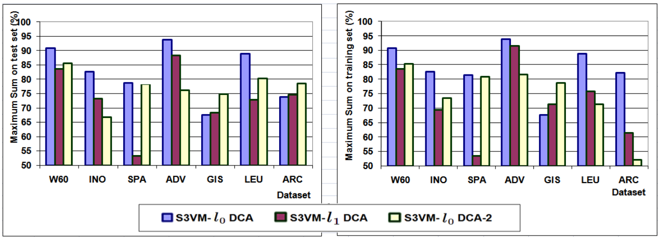


Fig. 2. Maximum sum (MS) of three algorithms on training (right) and test (left) set

From the computational results we observe that:

- **S3VM-l_0-DCA** reduces considerably the number of selected features (from 65% to 99%) while the classifier is quite good (from 65% to 95%).
- In comparison with **S3VM-l_1-DCA**, naturally **S3VM-l_0-DCA** and **S3VM-l_0-DCA-2** suppress much more features while always furnishes better accuracy (MS/ACC). The gain on number of selected features (SF) of **S3VM-l_0-DCA** with respect to **S3VM-l_1-DCA** is up to 33 times.

Table 2. Compasison of three algorithms

Dataset	SF			ACC (%)			MS (%)			CPU Time (s)		
	S3VM l_0-DCA	S3VM l_1-DCA	S3VM l_0-DCA-2	S3VM l_0-DCA	S3VM l_1-DCA	S3VM l_0-DCA-2	S3VM l_0-DCA	S3VM l_1-DCA	S3VM l_0-DCA-2	S3VM l_0-DCA	S3VM l_1-DCA	S3VM l_0-DCA-2
W60 Training set	**4 (12.5%)**	6 (18.75%)	7 (21.88%)	**88,42**	82,37	85,79	**90,82**	83,58	85,33	0,14	0,16	0,15
Test set				**92,06**	83,07	86,24	**90,82**	83,60	85,52			
INO Training set	**7 (20.59%)**	29 (85.29%)	8 (23.53%)	**76,50**	68,80	70,09	**82,65**	69,31	73,37	0,13	0,12	0,12
Test set				**75,22**	63,25	66,67	**82,65**	73,23	66,79			
SPA Training set	**20 (35.09%)**	54 (94.74%)	24 (42.11%)	**63,88**	**63,25**	**62,26**	**81,34**	53,32	80,82	0,39	0,40	0,37
Test set				**63,19**	**62,85**	**63,19**	**78,82**	53,15	**78,32**			
ADV Training set	**38 (2.44%)**	731 (45.76%)	586 (37.61%)	**95,28**	91,70	91,82	**93,81**	91,46	81,59	7,21	7,74	7,08
Test set				**93,06**	91,84	86,43	**93,81**	88,42	76,13			
GIS Training set	**1262 (38.00%)**	4363 (87.26%)	1262 (25.24%)	65,83	66,01	**69,08**	67,59	71,27	**78,74**	478,22	568,04	614,29
Test set				**67,40**	66,40	64,20	67,59	68,38	**74,71**			
LEU Training set	**28 (0.39%)**	40 (0.56%)	36 (0.50%)	84,21	78,95	76,32	**88,89**	75,85	71,25	6,31	6,12	4,67
Test set				**85,29**	73,53	61,77	**88,89**	72,81	80,31			
ARC Training set	**9 (0.09%)**	301 (3.01%)	244 (2.44%)	69,00	61,00	**77,00**	**82,19**	61,41	52,14	5,99	14,31	12,09
Test set				74,00	70,00	**80,00**	73,80	74,52	**78,55**			

Bold values indicate the best result of each experience

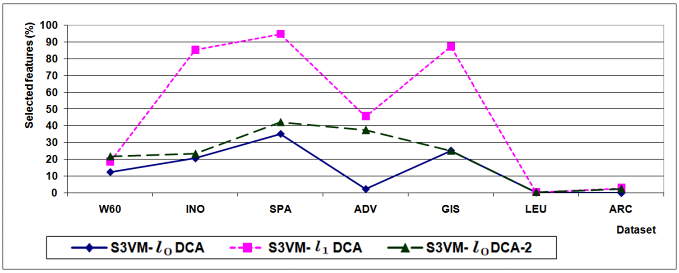


Fig. 3. Percentage of selected features of three algorithms

- As for the comparison of the effect of DC decomposition, **S3VM-l_0-DCA** always suppresses more features than **S3VM-l_0-DCA-2** (the gain is up to 27 times) while giving better accuracy for 6 out of 7 datasets.

Experiment 2: In this experiment, we are interested in the effectiveness of the three algorithms when the numbers of unlabeled points varies. For this purpose, we arbitrarily choose a data set (**Advert**) and then change the percentage of unlabeled points in training set from 20% to 80%. We report in Figure 4 (resp. Figure 5), the MS (resp. SF) of three algorithms on training and test set .

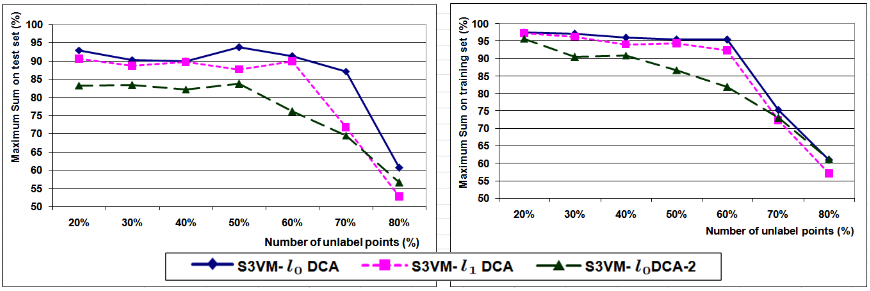


Fig. 4. Maximum sum (MS) on training (right) and test (left) of **Advert** dataset with different number of unlabeled points

We observe that:

- For most of case, **S3VM-l_0-DCA** gives better accuracy while choosing much less features than the two others algorithms (the gain on number of selected features of **S3VM-l_0-DCA** with respect to **S3VM-l_1-DCA**(resp. **S3VM-l_0-DCA-2** is up to 12 times (resp. 10 times)).
- When the numbers of unlabeled points varies, the percentage of selected features given by **S3VM-l_0-DCA** is stable (it varies between 1% and 4%), on contrary with **S3VM-l_0-DCA-2** (resp. **S3VM-l_1-DCA**) whose the interval is $[2\%, 45\%]$ (resp. $[23\%, 53\%]$).

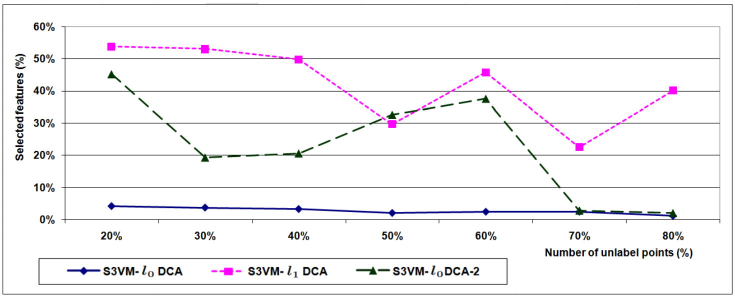


Fig. 5. Percentage of selected features on dataset **Advert** with different number of unlabeled points

- When the number of unlabeled points exceeds 70%, the accuracy of three algorithms decrease dramatically, especially for **S3VM-l_0-DCA-2** and **S3VM-l_1-DCA**.

Then, we can conclude that **S3VM-l_0-DCA** is more robust than other algorithms when the numbers of unlabeled points varies.

5 Conclusion

In this paper, we have proposed an efficient continuous nonconvex optimization approach based on DC programming and DCA for the combined feature selection and S3VM. Using an appropriate approximation function of zero-norm, we have obtained a DC polyhedral program. It fortunately turns out that the corresponding DCA consists in solving, at each iteration, one linear program. Furthermore, **S3VM-l_0-DCA** converges, after a finite number of iteration, almost always to a local solution. Numerical results on several real datasets showed the robustness, the effectiveness of the DCA based schemes. We are convinced that DCA is a promising approach for the combined feature selection and S3VMs applications.

References

1. Amaldi, E., Kann, V.: On the approximability of minimizing non zero variables or unsatisfied relations in linear systems. Theoretical Computer Science 209, 237–260 (1998)
2. Bennett, K.P., Demiriz, A.: Semi-supervised support vector machines. In: Proceedings of the Conference on Advances in Neural Information Processing Systems II, Cambridge, MA, USA, pp. 368–374 (1999)
3. Bradley, P.S., Mangasarian, O.L.: Feature Selection via concave minimization and support vector machines. In: Proceeding of ICML 1998, SF, CA, USA, pp. 82–90 (1998)
4. Chapelle, O., Zien, A.: A, Semi-supervised classification by low density separation. In: Proc. 10th Internat. Workshop on Artificial Intelligence and Statistics, Barbados, pp. 57–64 (2005)

5. Chapelle, O., Sindhwani, V., Keerthi, S.: Branch and bound for semi-supervised support vector machines. In: Advances in Neural Information Processing Systems, vol. 17, pp. 217–224. MIT Press, Cambridge (2006)
6. Chapelle, O., Sindhwani, V., Keerthi, S.S.: Optimization Techniques for Semi-Supervised Support Vector Machines. Journal of Machine Learning Research 9, 203–233 (2008)
7. Collobert, R., Sinz, F., Weston, J., Bottou, L.: Large scale transductive SVMs. J. Machine Learn. 7, 1687–1712 (2006)
8. Fung, G., Mangasarian, O.: Semi-supervised support vector machines for unlabeled data classification. Optimization Methods and Software 15, 29–44 (2001)
9. Joachims, T.: Transductive inference for text classification using support vector machines. In: 16th Inter. Conf. on Machine Learning, SF, USA, pp. 200–209 (1999)
10. Krause, N., Singer, Y.: Leveraging the margin more carefully. In: Proceeding of ICML 2004, NY, USA, pp. 63–71 (2004)
11. H.A. Le Thi, Contribution à l'optimisation non convexe et l'optimisation globale: Théorie, Algoritmes et Applications, Habilitation à Diriger des Recherches, Université de Rouen (1997).
12. Le Thi, H.A., Dinh, T.P.: Solving a class of linearly constrained indefinite quadratic problems by DC algorithms. Journal of Global Optimization 11(3), 253–285 (1997)
13. Le Thi, H.A., Dinh, T.P.: The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research 133, 23–46 (2005)
14. Le Thi, H.A., Belghiti, T., Dinh, T.P.: A new efficient algorithm based on DC programming and DCA for Clustering. Journal of Global Optimization 37, 593–608 (2006)
15. Le Thi, H.A., Le Hoai, M., Dinh, T.P.: Optimization based DC programming and DCA for Hierarchical Clustering. European Journal of Operational Research 183, 1067–1085 (2007)
16. Le Thi, H.A., Le Hoai, M., Nguyen, N.V., Dinh, T.P.: A DC Programming approach for Feature Selection in Support Vector Machines learning. Journal of Advances in Data Analysis and Classification 2(3), 259–278 (2008)
17. Le Thi, H.A.: DC Programming and DCA, `http://lita.sciences.univ-metz.fr/~lethi`
18. Liu, Y., Shen, X., Doss, H.: Multicategory ψ-Learning and Support Vector Machine: Computational Tools. Journal of Computational and Graphical Statistics 14, 219–236 (2005)
19. Neumann, J., Schnörr, C., Steidl, G.: Combined SVM-based feature selection and classification. Machine Learning 61(1–3), 129–150 (2005)
20. Thiao, M., Dinh, T.P., Le Thi, H.A.: DC programming approach for a class of nonconvex programs involving l_0 norm. In: Le Thi, H.A., Bouvry, P., Pham Dinh, T. (eds.) MCO 2008. CCIS, vol. 14, pp. 348–357. Springer, Heidelberg (2008)
21. Rakotomamonjy, A.: Variable Selection Using SVM-based Criteria. Journal of Machine Learning Research 3, 1357–1370 (2003)
22. Ronan, C., Fabian, S., Jason, W., Lé, B.: Trading Convexity for Scalability. In: Proceedings of the 23rd International Conference on Machine Learning ICML 2006, Pittsburgh, Pennsylvania, pp. 201–208 (2006)
23. Yuille, A.L., Rangarajan, A.: The Convex Concave Procedure. Neural Computation 15(4), 915–936 (2003)
24. Sindhwani, V., Keerthi, S., Chapelle, O.: Deterministic annealing for semi-supervised kernel machines. In: Proceedings of ICML 2006, NY, USA, pp. 841–848 (2006)

Information Gap Analysis for Decision Support Systems in Evidence-Based Medicine

Kari Sentz and François M. Hemez

Los Alamos National Laboratory, Los Alamos, NM 87545 USA
{ksentz,hemez}@lanl.gov

Abstract. The objective of evidence-based medicine is to come to well reasoned and justified clinical decisions regarding an individual patient's case based on the integration of case-specific knowledge, medical expertise, and the best available clinical evidence. One significant challenge implicated in this pursuit stems from the volume of relevant information that can easily exceed what can reasonably be assessed. Thus intelligent systems that can mine and synthesize vast amounts of information would be invaluable. The reconciliation of such systems with the complexity and subtlety of decision support in medicine requires specialized capabilities. One untapped capability is furnished through the gap in information between *what is known* and *what needs to be known* to justify a decision. In this paper, we explore the value of an information gap analysis for robust decision-making in the context of evidence-based medicine with an eye to the potential role in automated evidence-based reasoning systems.

Keywords: evidence-based medicine, robust decision making, information gap theory, deep question answering systems.

1 Introduction

The philosophy behind evidence-based medicine is to come to well reasoned and justified clinical decisions regarding an individual patient's case based on the integration of case-specific knowledge, medical expertise, and the best available clinical evidence. This is often defined referencing the well known editorial by Sackett et al. [8]:

> Evidence-based medicine is the conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients. The practice of evidence-based medicine means integrating individual clinical expertise with the best available external clinical evidence from systematic research.

This definition is formalized through a five-step process for evidence-based medicine, as excerpted from [3]:

1. Translation of uncertainty to an answerable question
2. Systematic retrieval of best evidence available

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 543–554, 2013.

3. Critical appraisal of evidence for validity, clinical relevance, and applicability
4. Application of results in practice
5. Evaluation of performance

One of the critical challenges that attends the practical application of this five step process is a consequence of the prodigious volume of medical publications. The number of medical papers published doubles every ten to fifteen years. [6] The majority of these publications are available electronically. While in principle this ready access should facilitate the incorporation of current best evidence into the clinical decision-making context, the onerous investment required to acquire (Step 2) and fully appraise vast quantities of evidence (Step 3) hampers its meaningful inclusion.

Clearly medical practitioners could be greatly advantaged by an automated way of obtaining and synthesizing useful evidence from an immense medical corpus. In particular, what is needed is an intelligent system that can mine and synthesize vast amounts of available information to provide accurate and meaningful answers to questions posed by human users. Such intelligent systems are called deep Question Answering (QA) systems. In particular, a deep QA system seeks to understand the relevance of evidence to a query, provide an answer based on this evidence, and communicate a level of confidence in possible answers to the query.

The current state-of-the-art of deep QA systems was demonstrated with the remarkable success of IBM's Watson in the high profile human versus machine Jeopardy! challenge that aired in 2011. Since the Jeopardy! challenge, IBM has focused on developing and applying this technology in evidence-based medicine. [4] Unlike the Jeopardy! case where there is a uniquely correct answer, the answers in the medical domain are more complex and ambiguous. As a consequence, the evolved deep QA system requires additional capabilities customized to cope with the subtlety of decision support in the medical domain. One such capability is afforded through the quantification of the informational disparity between *what is known* and *what needs to be known* in order to justify a decision. In this paper, we build on a previous example from deep QA in evidence-based medicine [4] and explore the value of this informational disparity for robust decision making in the context of a differential diagnosis with uncertain evidence.

1.1 Information Gap Analysis

The informational disparity described above is more commonly known as an information gap or info-gap. The best way to motivate the idea of an info-gap in the context of a decision is: *If you had perfect information, what information would change the decision?*. By answering this question, we identify the cruces of a decision. Then the current state of information can be viewed in terms of its distance from the perfect information case where the outcome of the decision changes. This distance between the current state of information and perfect information is the information gap. An info-gap provides a way to characterize the robustness of a model to the uncertainty in the decision space. In other words, this type of analysis can identify the outcomes that are most immune to failure due to uncertainty. [1] This idea originated by Yakov Ben-Haim has been used in wide variety of applications such as finance, conservation management, infectious disease detection, and structural dynamics. [1,7,9,5]

There are three components to an information gap analysis:

1. The process model
2. The performance requirement
3. The uncertainty model

The process model is the mathematical representation of the system or decision-making process. The performance requirement is assessed through a performance measure computed using the process model. The uncertainty model characterizes what is unknown about the parameters of the process model. [7] An info-gap model constitutes an unbounded family of consonant sets of uncertain events. The membership in the set is controlled by a horizon-of-uncertainty parameter, α. So info-gap theory proposes a structure for the uncertainty space and characterizes the clustering of uncertain events in sets. Importantly, info-gap does not assume a particular measure function for these uncertain events. As such, an info-gap model can admit *any* mathematical representation of uncertainty such as probability distributions, random sets, imprecise probabilities, etc. so long as it is consistent with the specified process model and the horizon-of-uncertainty parameter, α. [5,2] Info-gap provides an indicator of robustness to uncertainty rather than a measure of the uncertainty itself.

2 Example: Info-Gap Analysis for a Differential Diagnosis under Uncertain Evidence

One potential contribution that info-gap analysis can provide to decision making in evidence-based medicine is with the determination of the robustness of the models from which decision alternatives are obtained. As an illustration, we explore two alternative hypothetical models for aggregating evidence with respect to two disease alternatives. The current example is based on recent developments from Ferrucci et al. [4] for the extension of the Watson deep QA technology as a diagnostic support tool for differential diagnosis in evidence-based medicine. In this application of deep QA, a particular case can be mined wherein the patient's electronic medical record can be contextualized with other large volumes of structured and unstructured information. From this, a ranked list of potential diagnoses is generated which can then be evaluated along different dimensions of evidence the system determined to be relevant. Examples of evidential dimensions include: *Symptoms*, *Findings*, *Patient History*, etc.. So each ranked diagnosis is associated with a confidence score justified through a linked support of evidence broken out by dimension. [4] The connection is clear between the description of the first three steps of the evidence-based medicine process with the description of the evolution of the deep QA system to decision support system, namely: translate uncertainty to a question; retrieve the best evidence; and appraise it for validity, relevance, and applicability.

2.1 Process Model

There are many potential opportunities for an info-gap analysis in this context. For this example, we are interested in investigating the robustness of the evidence dimensions to uncertainty. The fact that these confidence scores are non-probabilistic uncertainty measures does not pose a problem for an info-gap analysis. As info-gap is a meta-analytic tool for quantifying robustness to uncertainty, it is agnostic as to the method used to quantify the uncertainty itself. In Figure 3 from [4] there is an evidential profile for the differential diagnosis between *Lyme Disease* and *Sarcoidosis 2*. There are four evidence dimensions: *Findings*, *Demographics*, *Symptoms*, and *Family History*. We estimated the values for these dimensions summarized in Table 1 and reconstructed a bar chart that is consistent with the IBM deep QA Evidence Profile visual display in Figure 1.

The most significant evidence dimensions are *Findings* and *Symptoms*. It is easy to show that diagnoses can change as a function of different process models. Suppose the following notional weights for two different weighted averages as summarized in Table 2. (Note that these are purely notional values in order to construct a simplified and illusory process model for the purposes of demonstrating the info-gap. These should not be mistaken as candidates for the real process models for combining evidential findings.) Utilizing these weights, we can construct a weighted average of each, as summarized in Table 3. As we can

Table 1. Strength of evidence

	Lyme Disease	**Sarcoidosis 2**
Findings	0.4	-0.08
Demographics	0.053	-0.027
Symptoms	0.04	0.347
Family History	-0.013	-0.173

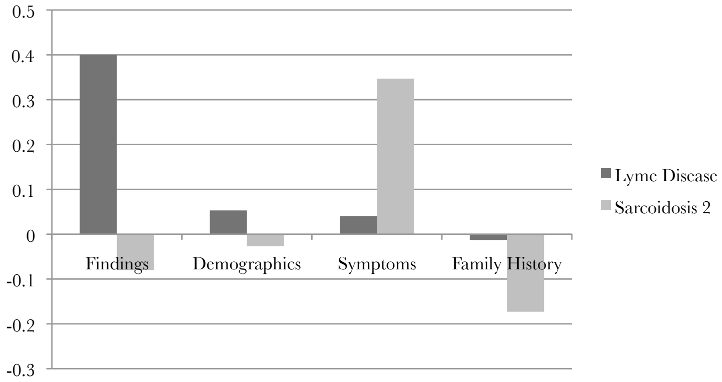


Fig. 1. Strength of evidence in favor of the two candidate diagnoses, as in [4]

Table 2. Weighting vectors for decision algorithms

	Weights	
	Weight I	**Weight II**
Findings	2	5
Demographics	1	1
Symptoms	5	5
Family History	1	1

Table 3. Different resulting diagnoses as a consequence of weighting

	Lyme Disease	**Sarcoidosis 2**
Findings	0.4	-0.08
Demographics	0.053	-0.027
Symptoms	0.04	0.347
Family History	-0.013	-0.173
Sum	0.48	0.067
Average	0.12	0.02
Weighted Average 1	0.26	**0.34**
Weighted Average 2	**0.56**	0.28

see in bold face in the last two rows of Table 3, the relative confidence based on these weighted averages change as a function of the weighting on the evidence dimension, *Findings*. The consequence of this is a change in the diagnosis from Sarcoidosis 2 to Lyme disease based on the weight of a single parameter. This contrast between results for two models based a single parametric change motivates the further examination of the two models with respect to robustness to uncertainty in the evidential dimensions.

2.2 Performance Function

We define the second component of the information gap analysis the performance function Y in terms of weights and the four evidential dimensions:

$$Y = w_1 Findings + w_2 Demographics + w_3 Symptoms + w_4 FamilyHistory \tag{1}$$

written symbolically as:

$$Y = w_1 X_1 + w_2 X_2 + w_3 X_3 + w_4 X_4 \tag{2}$$

where the weights w_k are those listed in Table 2 only scaled for convenience that the weights sum to unity:

$$w_1 + w_2 + w_3 + w_4 = 1 \tag{3}$$

The performance function is evaluated for the two diagnoses, labeled A for the Lyme disease and B for the Sarcoidosis 2 disease. A large value of the performance function indicates the strength of the evidence that supports the corresponding hypothesis. Then, a ratio between the two values of the performance function (Y_r) is calculated:

$$Y_r = \frac{Y(B)}{Y(A)} \tag{4}$$

A ratio $Y_r \geq 1$ greater than one indicates that there is more evidence in support of diagnosis B, that is, the Sarcoidosis 2 disease. Likewise, a value $Y_r \leq 1$ indicates that the evidence in favor of selecting the Lyme disease diagnosis A is stronger. The info-gap analysis of robustness presented next is applied to the ratio metric (Eqn 4).

The columns of Table 2 define different combinations of weights that make up the two notional decision algorithms. We refer to these as Algorithm A and Algorithm B, respectively. We will explore these two algorithms more closely with respect to their robustness to uncertainty and compare them for the purposes of decision-making.

2.3 Uncertainty Model

The final requirement of an info-gap analysis is to identify the uncertainty model which we define relative to the values of the evidential dimensions describes in Table 1. We consider these to be "true-but-unknown" values that can be from different the values listed in the table. Further, we assume that no additional information is available to determine where the "true-but-unknown" values may be located relative to these values (or stated equivalently that no information is available to describe the uncertainty of the values of the evidential dimensions). The uncertainty model, therefore, is defined as a model of fractional change. The "true" value of one of the four evidence dimensions, denoted as a variable X_k in Eqn 2, deviates from the nominal value $X_k^{Nominal}$ listed in Table 1 up to a percentage α:

$$-\alpha \leq \frac{(X_k^{Truth} - X_k^{Nominal})}{X_k^{Nominal}} \leq +\alpha \tag{5}$$

Eqn 5 defines α as a fractional change of variable X_k. It has no physical unit. Equally important, the magnitude of α is unknown, which expresses that the "true" value of the evidence dimension variable X_k is unknown.

2.4 Information Gap Analysis of Decision Algorithm *A*

The info-gap analysis of robustness searches for the minimum and maximum values of the ratio metric, Y_r (Eqn 4) given that the four evidence dimensions are allowed to vary away, and up to α, from the nominal values defined in Table 1. At a given level of robustness α, two optimization problems are solved to search for the minimum and maximum values of the ratio metric (Eqn 4) given that the four evidence dimension variables X_k can vary within the bounds defined in Eqn 5.

In the numerical implementation illustrated next, the value $\alpha = 1$ corresponds to ±50% variation of the nominal values of Table 1. This choice is arbitrary and requires justification. This formulation of the info-gap analysis is an eight-dimensional optimization problem since the four evidence dimension variables X_k are optimized for the first diagnosis A (Lyme disease) and the same four variables are optimized a second time, but independently, for the second diagnosis B (Sarcoidosis 2 disease). This implementation implies the assumption that the way the "true" values deviate from the nominal values for diagnosis A is uncorrelated from the way the "true" values deviate from the nominal values for Sarcoidosis 2.

The analysis of info-gap robustness is summarized in Figure 2 that shows the minimum and maximum values of the ratio metric (Eqn 4) for increasing levels of robustness α. The dotted line at $Y_r = 1$ separates the region where the evidence supports diagnosis A (Lyme) from the region where the evidence supports diagnosis B (Sarcoidosis 2). From the region defined by the maximum and minimum performance curves, it is clear that the decision algorithm favors the diagnosis B (Sarcoidosis 2) up to a level of robustness of $\alpha = 0.2$ as both curves lie to the right of the $Y_r = 1$ dotted line. (Recall that the level $\alpha = 0.2$ corresponds to ±10% variation of the values for the evidential dimensions). This means that the decision in favor of Sarcoidosis 2 as the diagnosis is best supported by the evidence profile, and this decision remains unchanged even if the evidence dimensions are incorrect up to ±10% away from the nominal values listed in

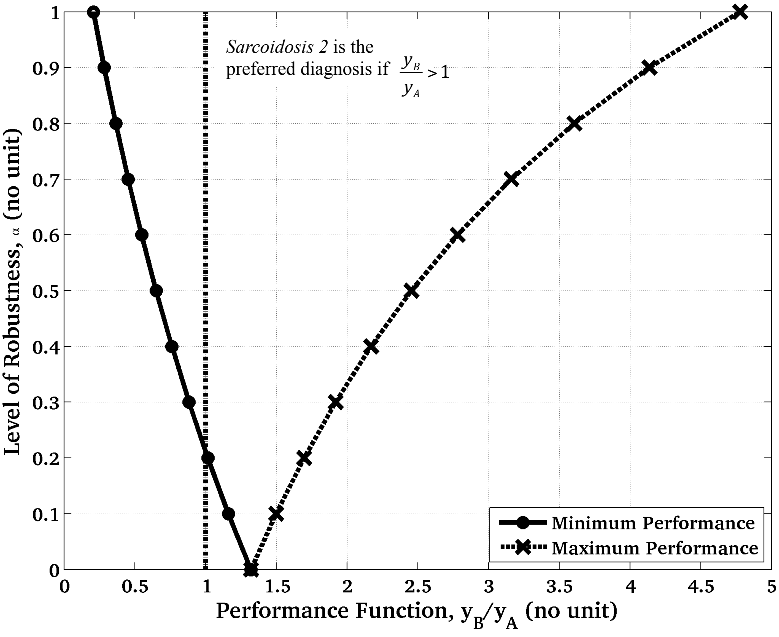


Fig. 2. Robustness of medical diagnosis with Algorithm A

Table 1. If it is plausible that the true values of the evidence dimensions can deviate from the nominal values by more than ±10%, or $\alpha \geq 0.2$, then the evidential support in favor of Sarcoidosis 2 is more questionable because the minimum ratio (indicated by the solid line in Figure 2) decreases below one. In practical terms, this means that confidence in the diagnostic decision distills to the the question of confidence in the evidence used to support it is correct to ±10%.

To illustrate how much evidence there is to support one diagnosis versus the other one, at a given level of robustness α, this can be quantified in terms of the relative percentages of support. The difference between the maximum and minimum values of the ratio metric (Eqn 4), represents the entire range of possibilities supported by the evidence profiles and the associated uncertainty. These maximum and minimum values are shown as the robustness or performance curves in Figure 3 (same as Figure 2). A simple approach to quantifying relative support is to measure how much of this range falls below the critical value for the *Performance Ratio*, $Y_r = 1$ and how much is above it. The values that fall below $Y_r = 1$ are in favor of Lyme disease (A) while the values greater than one are in favor of Sarcoidosis 2 (B). These values on either side of $Y_r = 1$ are shown on Figure 3 and summarized in Table 4 as a percentage of the total range.

Table 4. Strength of evidence in support of the two diagnoses

Level of Robustness α	**Minimum Percent of Perturbation**	**Maximum Percent of Perturbation**	**Strength of evidence in support of Lyme Disease**	**Strength of evidence in support of Sarcoidosis 2**
0.0	-0.0%	0.0%	0.0%	100.0%
0.2	-10.0%	10.0%	0.0%	100.0%
0.4	-20.0%	20.0%	16.86%	83.14%
0.6	-30.0%	30.0%	20.20%	79.80%
0.8	-40.0%	40.0%	19.56%	80.44%
1.0	-50.0%	50.0%	17.33%	82.67%

Figure 4 illustrates the "strength" of evidence in support of diagnosis A (Lyme disease), which is the fourth column of Table 4, as a function of the level of robustness α on the vertical axis. Clearly, there is no evidence in support of diagnosing the Lyme disease even if the evidence dimensions deviate from their nominal values by ±10%, which is $\alpha = 0.2$. Beyond this value of α, there is more uncertainty in the resulting diagnosis. This uncertainty peaks where the nominal values of the evidence dimensions are incorrect by up to ±30% ($\alpha = 0.6$), then there is a 20.20% "chance"[1], at most, that the correct diagnosis is Lyme disease.

[1] This "chance" is not "chance" in the sense of a probability; it may be more correct to call this metric a "strength" of the evidence collected in favor of one diagnosis or another.

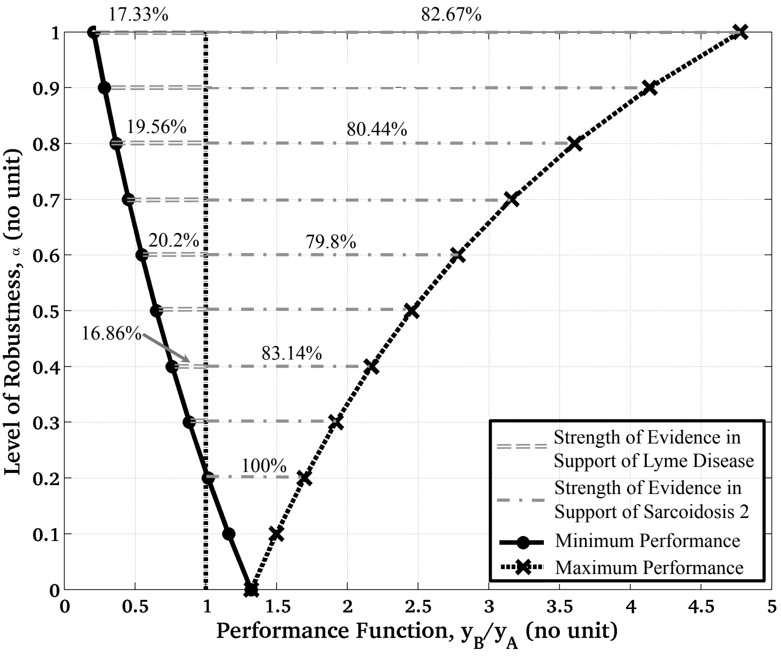


Fig. 3. Strength of evidence interpretation of robustness curves

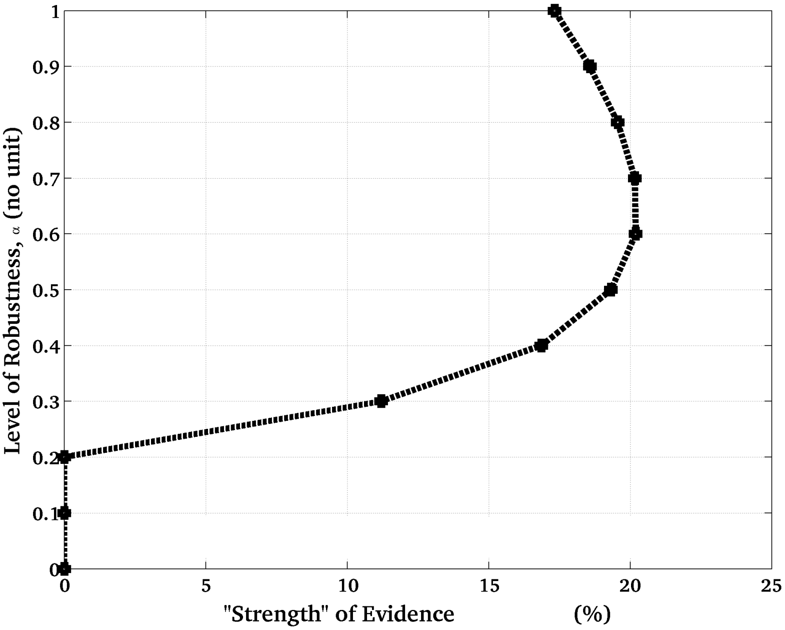


Fig. 4. Strength of evidence in support of diagnosis of Lyme disease

2.5 Information Gap Analysis of Decision Algorithm B

The info-gap analysis is repeated using Algorithm B. This second decision algorithm uses the weights w_k listed in the second column of Table 2. The definition of the uncertainty model is identical to the previous analysis as defined by Eqn 4 where the robustness level $\alpha = 1$ allows up to ±50% variability of the nominal evidence dimensions.

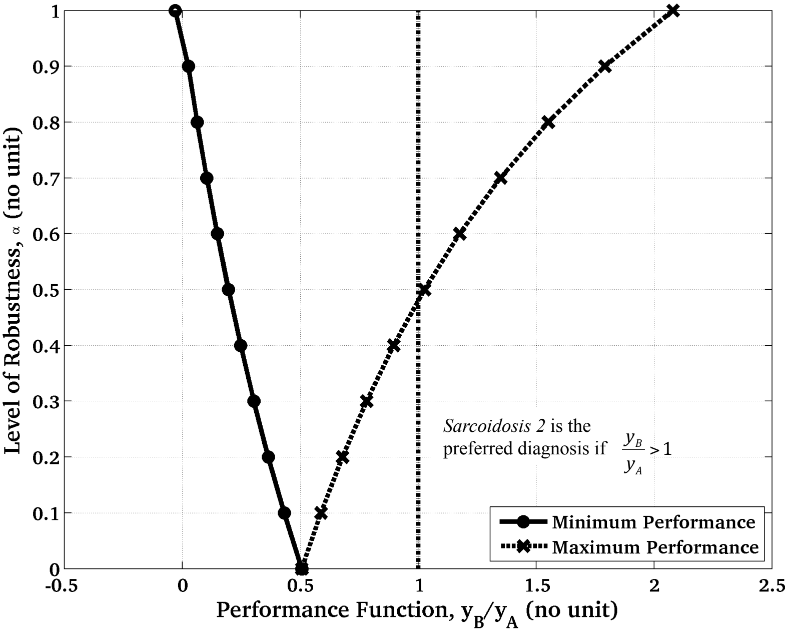


Fig. 5. Robustness of medical diagnosis with Algorithm B

Figure 5 summarizes the analysis of info-gap robustness by showing the minimum and maximum values of the ratio metric (Eqn 4) for increasing levels of robustness α. The decision to select the Lyme disease is maintained to the level of robustness $\alpha = 0.5$, approximately, where the maximum of the ratio metric, Eqn 4 starts to admit values that are greater than 1. Crossing the critical value of 1 means that the decision algorithm starts to support the possibility of the alternate diagnosis of Sarcoidosis 2.

2.6 Comparison of Robustness of the Two Decision Algorithms to Uncertainty

The info-gap analysis permits another perspective on these two different decision algorithms, namely, their relative robustness to uncertainty. When we compare Figures 2 and 5, we observe the similarity of the two plots in structural terms.

However, because the difference in location with respect to the threshold performance ratio, we find that the decision reached by Algorithm B contrasts with the previous one. Algorithm B prefers the diagnosis of Lyme disease at $\alpha = 0$ (where the nominal values of the evidence dimensions are used without variation). This difference in decision at the nominal values ($\alpha = 0$) corresponds to the findings initially discussed in Section 2.1 where a single parametric difference produced a different diagnosis. Interestingly, by looking at the relative degree of support in favor of the diagnoses and the robustness of the decision to uncertainty, we find that Algorithm B is more robust than Algorithm A because its initial decision remains unchanged up to $\pm 25\%$ (or $\alpha = 0.5$) variability of the nominal evidence dimensions as compared to $\pm 10\%$ only (or $\alpha = 0.2$) in the case of Algorithm A . From this comparison, one can say that the decision from Algorithm B is more robust to uncertainty than the decision from Algorithm A given the evidence.

3 Conclusions

In this paper, we introduce the prospect of an information gap analysis to support decision-making in the context of evidence-based medicine. In particular, an info-gap analysis provides a means of qualifying a decision based on a quantifiable measure between *what is known* (the nominal values) and *what is not known* (the amount of uncertainty). In the decision space, this furnishes the ability to specify where the decision could change as a result of uncertainty. As corresponds to intuition: if there is a low tolerance for uncertainty, then the resulting decision is not robust and could easily change. Analogously, if there is a high tolerance for uncertainty then the resulting decision is more robust and less likely to change. We demonstrate this idea with a differential diagnosis example where we studied the info-gaps of two notional decision algorithms and demonstrated the robustness of each to the uncertainty potential in the supporting evidence. As was illustrated, the info-gap could be used as a means of qualifying individual diagnoses in the context of a single process model, performance function, and uncertainty model. Moreover, the information gap could be used to compare models for their relative robustness to uncertainty.

While info-gap can be used as an independent analytic method for gauging the robustness of medical decisions under uncertainty, we imagine it could also be used as one additional unique capability in an intelligent evidence-based decision support system for clinical medicine. In the context of the five step process for evidence-based medicine, the role of an automated evidence-based decision support system would be in the gathering and synthesizing evidence from a massive corpus (Step 2-3). The role of an information gap analysis would be in the application of results and evaluation of performance of the decision themselves with respect to the uncertainty of the evidence used to support the diagnosis (Steps 4 and 5). These computational tools are intended as the complement to the evidence gathering, synthesis, and decision-making of the medical practitioner toward a collaboration that endeavors to leverage the unique and exceptional capabilities of each partner. Info-gap analysis serves as a useful method that can transit human decision-making and automated decision support.

Acknowledgements. The authors thank the reviewers for their helpful comments.

References

1. Ben-Haim, Y.: Information Gap Decision Theory, 2nd edn. Oxford University Press, New York (2006)
2. Ben-Haim, Y., Hemez, F.M.: Robustness, fidelity and prediction-looseness of models. Proceedings of the Royal Society A. 468(2137), 227–244 (2012)
3. Dawes, M., Summerskill, W., Glasziou, P., Cartabellotta, A., Martin, J., Hopayian, K., Porzsolt, F., Burls, A., Osborne, J.: Sicily statement on evidence-based practice. BMC Medical Education 5, 1 (2005)
4. Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., Mueller, E.: Watson: Beyond Jeopardy! J. Artif. Intell. (to appear, 2013)
5. Hemez, F.M., Ben-Haim, Y.: Info-gap robustness for the Correlations of Tests and Simulations of a Non-Linear Transient. Mech. Sys. and Sign. Proc. 18, 1443–1467 (2004)
6. Hook, O.: Scientific communications. History, electronic journals and impact factors. Scand J. Rehabil. Med. 31, 3–7 (1999)
7. Regan, H.M., Ben-Haim, Y., Langford, B., Wilson, W.G., Lundberg, P., Andelman, S.J., Burgman, M.A.: Robust Decision Making under Severe Uncertainty for Conservation Management. Eco. Applic. 15(4), 1471–1477 (2005)
8. Sackett, D.L., Rosenberg, W.M.C., Gray, J.A.M., Haynes, R.B., Richardson, W.S.: Evidence-Based Medicine: What it is and what it isn't. BMJ, 312–371 (1996)
9. Troffaes, M.C.M., Gosling, J.P.: Robust Detection of Exotic Infectious Diseases in Animal Herds: A Comparative Study of Three Decision Methodologies under Severe Uncertainty. IJAR (to appear, 2013)

TISA: Topic Independence Scoring Algorithm

Justin Christopher Martineau[1], Doreen Cheng[1], and Tim Finin[2]

[1] Samsung Information Systems North America
[2] University of Maryland Baltimore County

Abstract. Textual analysis using machine learning is in high demand for a wide range of applications including recommender systems, business intelligence tools, and electronic personal assistants. Some of these applications need to operate over a wide and unpredictable array of topic areas, but current in-domain, domain adaptation, and multi-domain approaches cannot adequately support this need, due to their low accuracy on topic areas that they are not trained for, slow adaptation speed, or high implementation and maintenance costs.

To create a true domain-independent solution, we introduce the Topic Independence Scoring Algorithm (TISA) and demonstrate how to build a domain-independent bag-of-words model for sentiment analysis. This model is the best preforming sentiment model published on the popular 25 category Amazon product reviews dataset. The model is on average 89.6% accurate as measured on 20 held-out test topic areas. This compares very favorably with the 82.28% average accuracy of the 20 baseline in-domain models. Moreover, the TISA model is highly uniformly accurate, with a variance of 5 percentage points, which provides strong assurance that the model will be just as accurate on new topic areas. Consequently, TISAs models are truly domain independent. In other words, they require no changes or human intervention to accurately classify documents in never before seen topic areas.

1 Introduction

Text analysis techniques, such as sentiment analysis, are valuable tools for business intelligence, predicting market trends, and targeting advertisements. This technology is especially salient because written works include tweets, Facebook posts, blog posts, news articles, forum comments, or any other sample of electronic text that has become prevalent due to the grow of the web.

Textual analysis applications need to operate over a wide and unpredictable array of topic areas, often in real-time. However, current approaches are unable to reliably and accurately operate in real-time for new domains.

Text analysis on a wide array of topic areas is difficult because word meaning is context sensitive. Word sense disambiguation issues are one reason why classifiers trained for one topic area do poorly in other topic areas. The linguistic community has spent a great deal of effort trying to understand the differences between word senses by build linguistic resources such as WordNet [5], WordNet Affect [12] and Senti-WordNet [1]. Word sense disambiguation is still challenging.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 555–570, 2013.

Fortunately, word sense disambiguation issues can be side stepped for specific problems. Consider sentiment polarity classification, which is the binary classification task where either the author approves of, or the author disapproves of the specific topic of interest. For sentiment polarity classification knowing word meaning is irrelevant, but knowing word connotation is crucial. In the following example, “I proudly wore my new shirt to the bank.” It is irrelevant whether the bank is a financial institution or a river bank because both senses of the word bank have no sentimental connotation for apparel. Thus, the word sense disambiguation problem can be simplified into a word connotation calculation. By extension to text classification: knowing the word’s sense is irrelevant, but knowing it’s class bias for a topic area is sufficient.

We introduce a method to determine topic independent class bias scores for words. These words can be used to build bag-of-words models that operate well in a wide area of diverse topics. Creating topic independent word scores is simple when there exists labeled data from multiple domains. Bias scores for a word can be calculated in each topic area using your machine learning algorithm of choice. A function can then be applied to these scores to determine a topic independent class bias score for the word. Intuitively, to measure topic independence, it makes sense to observe the variance of a word’s class bias in multiple topic areas. We introduce our Topic Independence Scoring Algorithm as a method to calculate topic independent class bias scores from a set of existing topic area specific class bias scores.

Since our Topic Independence Scoring Algorithm uses only bias scores produced by another supporting machine learning algorithm, it has several useful properties. First, the supporting machine learning can be swapped out. Machine learning experts can use our algorithm with the most appropriate algorithm for the task at hand. Second, our algorithm works on models not training data. This is very valuable in industrial settings when the training data may be lost or inaccessible due to business reasons. Alternatively, this is useful when the expertise to tune the original algorithm may no longer be available, but the model still remains. Finally, the topic independence scoring algorithm can be evaluated against the algorithm that produced the topic area specific scores. This allows us to more effectively evaluate the value of topic independence scoring.

As a use case and for evaluation purposes we build a topic independent model for sentiment analysis that is highly accurate across 20 never before seen test topic areas. Our topic independent model is even more accurate than the supporting machine learning algorithm in the test domains using 10 fold CV. Using our algorithm, we built a domain-independent sentiment model from five product review categories in the Amazon product reviews dataset [2] and evaluated it upon 20 additional product categories. Our classifier significantly outperforms the classifiers built specifically for each of the 20 product review categories. The baseline classifiers built specifically for the 20 test domains were almost twice as likely to make an error as our domain independent model.

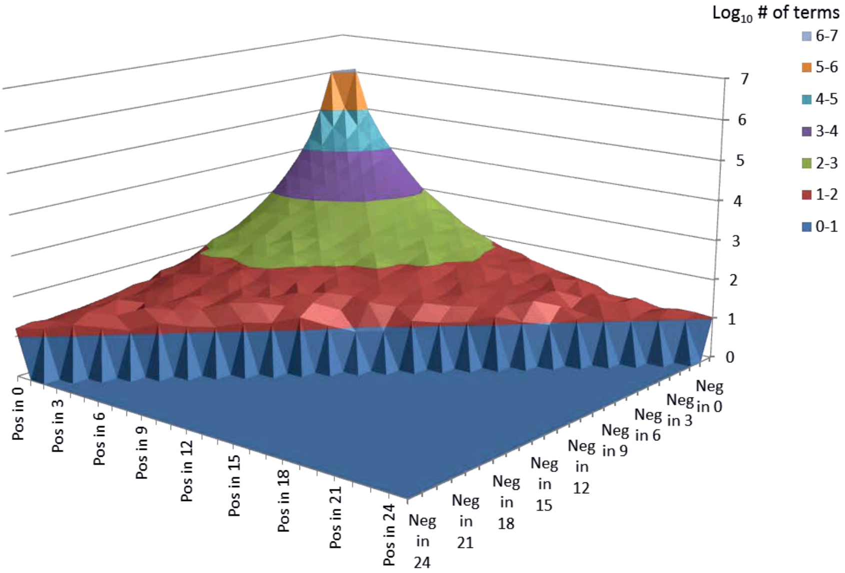


Fig. 1. Distribution of topic independence by positive vs. negative bias across 25 topic areas

2 Understanding Topic Independence

Our approach introduces the ground breaking concept of term level topic independence, which is the degree to which a terms orientation to a class remains the same when measured across multiple topics [7]. Poly-synonymous words are one reason why classifiers trained in a single domain do poorly in other domains. However, even when the sense of the word remains the same, the usage of that word implies different things in different topic areas. These problems lack a clear mathematical definition upon which machines can compute. Our Topic Independence Scoring Algorithm provides a clearly defined mathematical counting problem to eliminate word sense disambiguation issues when doing textual machine learning. The afore mentioned counting problem counts the different orientations a term has across multiple topic areas. This concept enables simple and fast computation for topic independent text analysis, and is therefore a very useful and important new concept.

We shall further explain topic independence using sentiment analysis as an example. A term can have either a positive, a negative, or a neutral connotation when it is used in context. Framed as a binary classification problem, the presence of any term is either an indicator of positive sentiment, an indicator of negative sentiment, or it has no class bias. This bias can be determined in context by determining if documents in that context (aka domain or topic area) are more likely to be positive or negative when that term is present. Given a set of different contexts we can count the number of contexts where the term is positive and the number of contexts where the term is negative. In Figure 1 we chart these values along the x and y axis for every term in the popular 25 category Amazon Product Reviews Dataset [2]. This chart shows why our Topic Independence Scoring Algorithm is so important.

Sentimental topic independence is a matter of degree: there is almost always some situation where a normally positive word or phrase will have a negative connotation. The topic independent sentiment bias of a term should be based not only upon its sentimental strength in most situations, it must also be weighted by it‘s reliability and uniformity. Put another way, the exceptions are so frequent that they must be accounted for in the general rule.

Figure 1 shows that there are only 11 terms that have a positive sentimental orientation in all 25 product review categories, while 16 terms have a negative sentiment orientation. For example, the 11 most topic independent positive terms: “excellent”, “highly recommend” ”, “the best”, “best”, “an excellent”, “I love”, “love”, “wonderful”, “a great”, “always”, and “recommend” occur at. For example, the 11 most domain independent positive terms: excellent, highly recommend , the best, best, an excellent, I love, love, wonderful, a great, always, and recommend. The most topic independent negative terms include: “don’t waste”, “ your money”, “waste”, “waste your”, “would not”, “money”, “disappointed”, “worse”. These terms are very revealing, but are not enough to cover a representative sample of any given text document.

The vast majority of all terms, over two million, are unique to exactly one product category in our dataset. From that peak, the total volume of terms falls off very rapidly according to the degree of topic independence. This implies that we need to properly scale the sentiment strength scores for terms with their degree of sentimental topic independence in-order to use the less topic independent terms without overpowering the more topic independent terms.

3 Approach

The unique idea of our approach is to build a topic independent model by scoring terms based upon how much their class bias shifts *as observed across many topics*. By doing this *irrespective of the target topic area* where the model will be applied we can be reasonably confident that the model will work well for any topic area. This contrasts quite sharply with domain adaptation methods that seek to adapt a model build for one domain into a model that will better fit another specific domain. Using domain adaptation thus ensures that you will need to domain adaptation again for the next domain. Furthermore, this kind of custom fitting to a single dataset is more likely to overfit artifacts in those datasets than a model that must fit multiple different domains since artifacts can be cross-checked with other domains. Domain independent models are much more useful than single domain models because they are more broadly applicable and less susceptible to artifacts and other noise.

Training a classifier with out-of-domain data can be accurately preformed if you can answer two key questions:

1. For any term, what is that term's class bias in the source domain(s)?
2. From this bias what can be concluded about its bias in the target domain?

The first question is fairly straight forward and easy to answer using standard techniques for supervised machine learning. Delta TFIDF [8] weights works particularly well for this task [9], but they can easily be replaced as the state-of-the-art advances.

Answering the second question is difficult for current domain adaptation approaches because they model the situation as a relationship between a predetermined training topic area and the target topic area. This setup assumes a different relationship between each pair of topics.

Question two is very difficult to answer with that assumption, so let us instead *assume that the class bias of a term is equally likely to shift between any randomly selected pair of topics.* This implies that we can predict how likely any given term's class bias is to shift when applied to another arbitrary topic area simply by observing how frequently it actually shifts class bias between multiple topic areas. Similarly, we can observe the magnitude of these class bias shifts to determine the likely magnitude of the class bias for other arbitrary topic areas.

Term level topic independence class bias scores need to measure and unify the following semantics:

– *Sensible orientation* : A term's class orientation should agree with its overall orientation in the set of topic areas.
– *Strength* : Terms with higher scores in the topic areas should have higher scores.
– *Broad applicability* : Terms that are used in more topic areas should have higher scores.
– *Uniform meaning* : Terms with more uniform topic area scores should have higher scores.

To measure these semantics we introduce our Topic Independence Scoring function in Equation 1. *Strength* can be measured with a simple average. This average should also give us a *sensible orientation.* A strongly oriented term is more valuable as it becomes more *broadly applicable* so multiplying the strength score and the applicability score makes sense. The *uniform meaning* metric is difficult. Variance is not a good choice since variance scores increase with dis-uniformity, have an undefined range, and when all the values are multiplied by a constant the variance goes up by the square of the constant. Attempting to address the dis-uniformity problem by dividing the other scores by the variance is not a good solution because this can cause divide by zero problems and because of the rate at which the variance score changes. A good way to score uniformity is to use the geometric mean of the topic area scores. The geometric mean is a good choice because it has a predefined range with a maximum equal to the arithmetic mean when the values are totally uniform and with scores dropping as uniformity decreases. This final uniformity term should be multiplied with the earlier calculations because, a strong broadly applicable term is more valuable when the strong scores are more uniform.

Given that:

D_t is the number of topics that term t occurs in.
$S_{d,t}$ is the class bias score for term t in topic area d.
$TIS(t)$ is the feature value for term t.
We calculate Topic Independence Scores with the following formula:

$$TIS(t) = \sum_{d}^{D_t} S_{d,t} \left(\prod_{d}^{D_t} |S_{d,t}| \right)^{1/D_t} \tag{1}$$

The Topic Independence Scoring Algorithm creates a topic independent model from a set of existing topic dependent models using the TIS function on each term. Algorithms, such as SVMs and Logistic Regression use weight vectors to produce judgments. With a set of topic area specific models built by such algorithms TISA can produce a new topic independent weight vector covering all the terms in the source models. This new weight vector can be used to do topic independent classification using the same classification algorithm that produced the original topic area specific weight vectors. Our topic independent classification can be easily used with a wide variety of popular machine learning algorithms: There is a very low adoption barrier.

4 Evaluation

In this evaluation we demonstrate how to build topic independent sentiment models using our topic independence scoring algorithm. We demonstrate that:

1. Topic independent sentiment models outperform in-topic models.
2. Topic independent models use additional out-of-topic training data more effectively than alternative techniques including:
   (a) Weighted voting with multiple models.
   (b) Building a single model on the union of multiple topic area datasets.
3. Topic independent sentiment models can be used to find revealing and informative topic specific vocabulary.

Our topic independent sentiment model is 89.6% accurate when measured over 20 additional held-out test topic areas with a low variance of 5.05 percentage points. Our approach is the most accurate approach published on this dataset.

4.1 Test 1: TISA vs. In-topic Models

This test evaluates our topic independence scoring algorithm as a method for domain independent sentiment classification using 20 different held-out test topic areas.

For our baseline we used the standard 10-fold cross-validation methodology in each of the 20 test topic areas. For this baseline we choose to use the Delta IDF [7] classification algorithm, which is a slight modification on the Delta TFIDF document feature weighting algorithm [8]. To train a Delta IDF model calculate each feature in the bag-of-words as shown below and add them to a weight vector. Given that:

$|P_t|$ is the number of positively labeled training documents with term t.
$|P|$ is the number of positively labeled training documents.
$|N_t|$ is the number of negatively labeled training documents with term t.
$|N|$ is the number of negatively labeled training documents.
V_t is the feature value for term t.

$$V_t = \log_2 \left(\frac{(|N| + 1)\,(|P_t| + 1)}{(|N_t| + 1)\,(|P| + 1)} \right) \tag{2}$$

We need to balance the positive vs. the negative bias because we know that the datasets have been class balanced by the original author of the dataset. Follow the procedure described below.

Bias Balancing Procedure:

1. Create a copy of the weight vector and call it the positive vector. Call the original vector the negative vector.
2. For every feature in the positive vector, if the feature value is less than zero set the value to zero.
3. For every feature in the negative vector, if the feature value is greater than zero set the value to zero.
4. L^2 normalize the positive vector
5. L^2 normalize the negative vector
6. Add the positive and negative vectors together and return the answer.

For our classification function we use the dot product of the document with the weight vector. Data points with a dot product greater than or equal to zero are positive, otherwise the point is negative.

To keep our comparison uniform and meaningful we apply the same bias balancing procedure and use the same classification function for both TISA and Delta IDF. Bias balancing is a good idea for TISA because the overall class balance is topic area dependent. For example, while most people love their digital cameras they absolutely hate their anti-virus software.

To further eliminate external factors we use the Delta IDF algorithm to produce the set of domain specific feature scores used by TISA in equation 1. Since Delta IDF does not have any tunable parameters, no one can claim that the input models for TISA were better tuned that the baseline models. These choices remove potential confounding factors that could have been responsible for TISA's better performance.

We built our topic independent model from a set of five topic dependent models using TISA as described in our approach. The five source models were built using Delta IDF on a different set of topic areas than the 20 held out test topic areas. The five source models were built on the books, dvds, electronics, kitchen appliances, and music topic areas because these are the most popular domains. This matches real world situations where there exists more labeled data for popular topic areas and far less labeled data for other areas.

Table 1. A general model built form using TISA to combine Delta IDF scores on data about books, DVDs, electronics, kitchen appliances, and music does very well on 20 different product categories when compared to in-domain models built using Delta IDF on each of the categories

Target Category	In-Dom Model	TISA Model
Apparel	89.17	89.90
Automotive	80.92	85.99
Baby	89.41	90.32
Beauty	85.38	90.62
Camera	86.54	91.56
Cell Phone	83.66	83.82
Comp Games	72.77	88.04
Food	76.41	88.86
Grocery	84.25	89.14
Health	87.36	89.31
Instruments	84.28	90.32
Jewelry	85.32	89.44
Magazines	85.40	89.68
Office	76.32	89.91
Outdoor	84.13	92.41
Software	79.44	87.43
Sports	87.09	90.24
Tools	56.67	94.74
Toys	86.87	90.40
Video	84.19	89.46
Average	82.28	89.60
Variance	55.70	5.05

On average our topic area independent model is 89.60% accurate, which is a statistically significant improvement over the 82.28% accurate product area specific Delta IDF baselines to the 99.9% confidence interval. Table 1 shows the accuracy of our TISA model compared to the baseline for each of our 20 test product review categories. Please note that the low accuracy of the tools baseline is not a mistake. We will discuss it in greater detail in the next section.

Unlike other algorithms, TISA is highly accurate on every topic area with very low variance. Even though many of the topic areas are substantially different from TISA's training data our TISA model is more accurate and nearly 11 times more stable in terms of variance than the domain specific models. While domain adaptation algorithms try to exploit the relationship between topic areas, TISA attempts to minimize the effects of these relationships. This decouples TISA's training topic areas with its testing topic areas. This has the added benefit of allowing researchers working with TISA to use labeled data from any topic area. This can allow researchers to avoid using low quality topic area datasets, such as topic areas with very little data, harder data points to classify, or low inter-annotator label agreement.

Table 2 illustrates the difference between topic independent term scores produced by TISA and topic dependent scores that were used as input to TISA. This table shows the top 50 most negative and most positive words or pairs of words for TISA and the baseline models. The terms highlighted in the figure show that TISA's most important terms are very general purpose, while the terms in the input books model are very specific to the books topic area. These example terms support our argument that TISA favors topic independent bias terms.

The product specific baselines built using Delta IDF make for an excellent comparison. These product specific baselines are not straw men; they have been shown to outperform Support Vector Machines on this dataset [7]. By correctly setting up our experiment we have eliminated confounding factors and can conclude that the quality of the models is responsible for the difference between the two algorithms. By evaluating TISA against the Delta IDF algorithm used to create its constituent sub-models we negate any potential objections that our improvement was due to the difference between the baseline algorithm and the algorithm used to create the sub-models. Thus the difference between the two models comes from either the intelligent combination of models using TISA, or the amount and quality of the training data. Both of these are good points for TISA, since TISA allows the researcher to freely select dataset without respect to the topic area that the model will be used on.

4.2 Test 2: TISA vs. Ensemble Methods

Skeptical readers might object to comparing TISA against in-domain Delta IDF because TISA is using more total labeled data. In machine learning, it is well known that using more training data will improve accuracy, but it is also well known that using training data that is not similar to the test data will hurt accuracy. One of TISA's main benefits is that it allows machine learning practitioners to leverage large amounts of dis-similar data by reducing the impact of the dis-similarity. The tools entry in Table 1 is a clear example of why this approach is so important: using more training data is the entire point of domain adaptation.

Table 2. Top 50 most positive and negative terms for the Books domain as determined by in-domain Delta IDF vs. Top Most positive and negative terms as determined by TISA using Books, DVDs, and Electronics. All terms shown have the correct sentimental orientation and are strongly oriented. However, in-domain Delta IDF identifies many features, shown in bold and highlighted in **red** , that will not generalize well to non-book data. Instead, TISA placed more importance on terms, shown in italics and highlighted in *green* , that should generalize very well to other domains.

TISA Identifies General Terms by Decreasing the Score of Topic Specific Terms

Positive Terms Books Domain	Positive Terms TISA	Negative Terms Books Domain	Negative Terms TISA
must for	*highly recommended*	waste your	waste your
magnificent	only complaint	not worth	very disappointed
only complaint	must for	two stars	two stars
worth every	worth every	very disappointing	a refund
excellent read	great addition	**worst book**	refund
a must	delighted	uninteresting	don’t waste
wonderful book	a must	very disappointed	waste of
delighted	*great buy*	sorry but	*not recommend*
definitely worth	every penny	don’t waste	your money
great resource	excellent for	a joke	not worth
excellent overview	an outstanding	waste of	zero stars
excellent reference	must-have	not waste	very disappointing
a delight	*well worth*	very poorly	complete waste
essential reading	*another great*	is poorly	save your
must-have for	definitely worth	save your	very poorly
my clients	a must-have	a disappointment	*avoid this*
weaves	and allows	poor quality	not waste
great addition	my only	no new	waste
detailed account	great way	refund	a waste
a magnificent	*highly recommend*	a poorly	*buyer beware*
great fun	*are amazing*	excuse for	total waste
offers an	*is superb*	wasted my	wasted my
pleasantly surprised	*excellent condition*	complete waste	big disappointment
every penny	exceeded my	skip this	a disappointment
great introduction	superb	big disappointment	*of junk*
pleasure to	*pleasantly surprised*	zero stars	hard earned
and accessible	*is awesome*	**terrible book**	really disappointed
be required	*great condition*	**worst books**	a joke
really helped	*great product*	**poorly organized**	*don’t buy*
be missed	not disappoint	**good reviews**	*stinks*
not disappoint	i highly	your money	*money back*
top notch	*excellent choice*	disappointing	*a poorly*
terrific book	*best ever*	**boring book**	poor quality
beautifully written	excellent i	a refund	*returned this*
excellent resource	loves this	unfortunately this	*insult to*
transcends	outstanding	**poorly written**	or money
renewed	delighted with	**factual errors**	*extremely disappointed*
great collection	recomend it	**glowing reviews**	*is terrible*
fabulous book	gem	new here	disappointment
must-have	*loves it*	disappointment i	*not buy*
first rate	*very pleased*	total waste	*not recommended*
an outstanding	*definitely recommend*	am disappointed	stay away
refreshing and	no nonsense	was boring	don’t bother
you wanting	also great	irritated	worthless
a pleasure	*can’t beat*	**even finish**	*i regret*
developing a	the raw	disappointing i	huge disappointment
teaches us	great look	had hoped	*never buy*
from home	thumbs up	disappointment	*dud*
poems and	she loves	**drivel**	disappointing
very comprehensive	*love this*	a waste	the trash

One reason why TISA is very accurate is that it preserves and intelligently uses the information captured by splitting the document pool into different domains or topic areas. Consider building a Delta IDF dot product classifier using the union of all the data that the topic independent model was trained on. That process ignores the information provided by the subdivision in the dataset between different domains. Table 3 shows that the TISA model is more accurate than a Delta IDF classifier created from the union of the same set of documents at an accuracy of 89.6% to 86.3%. This difference is significant to the 99.5% confidence level. Clearly it is better to use the information provided by domain membership than to ignore it.

Table 3. General TISA "BDEKM" model built from the Books, DVDs, Electronics, Kitchen Appliances, and Music Delta IDF models vs. Weighted Voting with these models vs. a single Delta IDF model built on the union of all the Books, DVDs, Electronics, Kitchen Appliances, and Music data. Results have been sorted by size. The 10-fold in-domain accuracies for each test domain are displayed for reference.

Target	**Dom**	**In-Dom**	**TISA**	**Union**	**Weighted**
Category	**Size**	**Model**	**Model**	**Model**	**Voting**
Tools	19	56.67	94.74	73.68	84.21
Instruments	93	84.28	90.32	88.17	87.10
Office	109	76.32	89.91	88.07	87.16
Automotive	314	80.92	85.99	81.85	80.57
Food	377	76.41	88.86	86.21	87.27
Computer Games	485	72.77	88.04	85.98	84.33
Outdoor	593	84.13	92.41	90.22	89.71
Jewelry	606	85.32	89.44	88.45	88.61
Grocery	654	84.25	89.14	88.23	88.69
Cell Phone	692	83.66	83.82	78.90	79.05
Beauty	821	85.38	90.62	87.33	88.67
Magazines	1124	85.40	89.70	86.39	87.82
Software	1551	79.44	87.43	84.53	83.75
Camera	1718	86.54	91.56	88.07	88.53
Baby	1756	89.41	90.32	89.07	89.46
Sports	2029	87.09	90.24	87.83	88.37
Apparel	2603	89.16	89.90	88.21	89.44
Health	2713	87.36	89.31	85.51	86.10
Video	4726	84.19	89.46	90.12	88.30
Toys	4929	86.87	90.40	89.06	89.53
Average	2317	82.28	**89.60**	86.30	86.83

A popular alternative technique to leverage more out of domain data is to use multiple classifiers under a weighted voting approach. Delta IDF dot product classification is particularly well suited to this approach because, when both the documents and the weight vectors are normalized to unit length, the magnitude of the dot product can serve as the vote's weight. Weighted voting using the books, DVDs, electronics, kitchen appliances, and music domains over the test domains is 86.83% accurate. The difference between weighted voting and the TISA method using the same training and test points is significant to the 99.9% confidence level. The weighted voting approach is statistically no different than the union model as indicated by a p-value of .3555 . These results are displayed in detail in Table 3.

4.3 Sentiment Feature Mining

In many case it is valuable to know what the important domain specific bias features are. For example, someone who is shopping for clothes may want to know why a specific article of clothing was rated poorly by users. While reporting to the shopper the highest scoring topic independent features for the product will clearly show that people did not like the product, it will not do a good job of showing why people did not like the article of clothing because topic independent features are very generic. To solve this sentiment mining problem we must report to the shopper the topic specific reasons why people did not like the article of clothing.

Fortunately, the topic-independent model can be used to automatically generate topic-specific sentiment models. These topic specific models can then be used to report specific reasons why people liked or disliked the topic.

Table 4. Top 50 most positive and negative terms mined for the apparel topic area using the topic independent model built by TISA on books, DVDs, electronics, kitchen appliances, and music data. The terms are strongly sentimental and are correctly oriented for apparel. The terms tend to be very specific to the apparel topic area.

Positive Terms		**Negative Terms**	
compliments on	toe is	returned them	poor customer
great quality	hubby	holes	received a
is comfortable	thick as	defective	credited
are soft	so soft	cheaply made	disappointed when
great item	wanted a	the return	recieved the
confortable	tons of	policy	post
great shoes	best bra	make sure	return shipping
ones and	locally	charged	cancelled
and comfortable	monday	the photo	i emailed
with jeans	great	to remove	never order
great bag	really great	sent the	ears
fit very	definitely buy	so thin	wont
them very	best shoes	send the	item back
khaki	are exactly	the ankle	top and
were exactly	sleek	off my	tore
comfortable they	walking shoe	ordered <num>	too wide
is slightly	good shoe	known	i see
he really	ride up	times and	the seam
love em	last forever	holes in	just about
feels great	things and	shrunk	pay to
reasonable price	under jeans	so tight	pants were
many different	very confortable	<num> sizes	thin that
bra ever	as thick	big and	opened
comfortable from	wanted something	thin and	ordered a
even in	tons	torn	uncomfortable the

This takes 3 steps: (1) gather a set of documents about the topic the user is interested in, (2) classify every document using the topic-independent model and label them as positive or negative with the classifiers decision, (3) compute

$\varDelta$IDF(t) scores for terms in the set of documents that were mechanically labeled in the previous step. The top most features of this model are the strongest reasons why people liked or disliked the product. Table 4 shows the top 50 strongest sentimental terms for the clothing topic computed using this method.

The words and phrases shown in Table 4 are good apparel specific indicators of sentiment that help explain why a user liked or disliked the piece of apparel under review. Many of these phrases express an opinion about an apparel specific product feature. For example, “Feels great” indicates a positive opinion about the feel of the clothing. Likewise, “Great quality” expresses a positive opinion about the item’s quality. Other phrases assert a good, or bad, property of apparel for the item under review. Examples include “Is comfortable” and “Are soft” both of which are desirable aspects of many apparel items. Other stop words, or near stop words, are informative components of strong apparel specific sentiment indicators. Sentiment amplifiers such as “So” ,“Very” , and “Really” are important stop words because they amplify the strength of the rest of the phrase. The presence of these phrases can indicate why a user gave a positive or negative rating to a piece of apparel.

5 Related Work

Supervised machine learning is a common approach for sentiment analysis. Normally, a classifier is trained on a hand labeled dataset for the specific topic area of interest. Training these classifiers generally takes a long time, but once they are trained they can rapidly make accurate judgments of the type they were trained to make, on the type of things they were exposed to during the training process. Using Support Vector Machines [6] with a bag-of-words feature space is one of the most popular examples of this approach, including the seminal work on sentiment analysis for movies [11].

While these in-domain methods work well in a predefined topic area with a sufficient amount of labeled data they do not work well when used outside of the predefined topic area. As a result these methods do not work well for important applications, such as personal assistants, that need to provide answers for any domain, or topic area, that the user is interested in at the moment.

Current domain-adaptation approaches such as CODA [4], SCL-MI [2], SFA [10], and Couple Spaces [3] build a model for a domain, which has no labeled data, using labeled data from a different domain. This is unacceptable because it is infeasible to train a new model in real-time whenever an electronic personal assistant encounters a question about a new domain.

To address these challenges and enable personal assistants to succeed in unexpected topic areas we took a strikingly different approach to re-score sentiment features using their domain-independence. Our work alone has been designed to build models that remain highly accurate even when they are used on unfamiliar topics that may be vastly different.

In a business setting it is highly desirable to be able to deploy trained models on new topic areas that they were not designed for. Training these models should not require any special changes for the topic area. Furthermore, these models should be highly accurate in every topic area that they will be used upon even if the list of topic areas they will be used upon is unknown. Unlike state-of-the-art Domain Adaptation approaches, our TISA fulfills these demands as summarized in Table 5.

Our approach is highly accurate across 20 never before seen test domains. Surprisingly, our algorithm is even more accurate than models that were custom tailored to the test domains.

Table 5. TISA has the easiest to satisfy training data requirements, is simple, fast, highly accurate, and reliable. Caution should be taken when directly comparing the average accuracy and variance numbers of TISA and our ΔIDF baseline to other published approaches due to the different training environments described.

Comparison Criteria	**TISA**	In-Dom ΔIDF	SCL-MI	SFA-DI	CODA with 0 Target	CODA with 1600 Target
Situations Modeled	20⋆	20⋆⋆	12⋆⋆⋆	12⋆⋆⋆	12⋆⋆⋆	12⋆⋆⋆
Requires Labeled Data from Other Domains	Yes	No	Yes	Yes	Yes	Yes
Requires In-domain Labeled Data	**No**	Yes	No	No	No	Yes
Requires Unlabeled In-domain Data	**No**	No	Yes	Yes	Yes	Yes
Average Accuracy	**89.6**	82.28	77.97	78.66	83.23	86.46
Variance	5.05	55.70	25.38	17.29	11.54	2.89

6 Conclusion

In this paper we showed that topic-independent sentiment analysis is highly important for a wide array of applications. We pointed out how state-of-the-art domain-adaptation approaches do not address these problems. To address these problems, we designed our approach with the core goal of accurate sentiment classification for unforeseen topic areas.

Our algorithm has several advantages over other approaches because it does not require any information about the topic area, including labeled or unlabeled

⋆ Each modeled situation corresponds to a product review category since each is a held-out test set.

⋆⋆ Each product review category is a topic area and is treated as a test situation. Although 10-fold cross-validation is used in each product review category folds are not counted as a test situation. Average and variance scores are computed over test situations. Please note that the average and variance reported in this table for Δ IDF includes domains that TISA was trained on.

⋆⋆⋆ Each unique source/target product review category pair is being treated as a modeled situation. Every domain adaptation source/target pair for the Books, DVDs, Electronics, and Kitchen product review categories were modeled.

data from the topic area. First, machine learning experts can use our scoring algorithm with the most appropriate algorithm for the task at hand. Second, even if the training data has been lost, is inaccessible due to business reasons, or the expertise to tune the original algorithm is no longer available, existing models can still be used with TISA to produce topic independent models. Third, training time is substantially reduced for super-linear training algorithms by cutting the number of documents down into multiple smaller pools. Fourth, TISA can leverage existing labeled data in any number of topic areas. We speculate that this reduces overfitting and leads to our demonstrated better results.

TISA is the only true scalable topic-independent sentiment analysis solution for real world problems. A single topic-independent model built using TISA is vastly preferable to using multiple models domain specific models for the following reasons: One, a single model is much easier and less costly to create and maintain. Two, topic independent models do not require topic detection to determine which domain specific model to use. Three, topic-independent models created using TISA are even more accurate than topic-specific models due to their ability to leverage more data and reduce the affects of noisy features. Four, our topic-independent models are 11 times more reliable than domain specific models. Five, TISA models require no changes to work well on a new topic area. These factors make TISA the best choice for practical real world sentiment analysis.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), Valletta, Malta. European Language Resources Association (ELRA) (2010)
2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Annual Meeting of Association For Computational Linguistics, vol. 45, pp. 440–447 (2007)
3. Blitzer, J., Kakade, S., Foster, D.P.: Domain adaptation with coupled subspaces. Journal of Machine Learning Research - Proceedings Track 15, 173–181 (2011)
4. Chen, M., Weinberger, K.Q., Blitzer, J.: Co-training for domain adaptation. In: NIPS 2011, pp. 2456–2464 (2011)
5. Fellbaum, C.: Wordnet. In: Theory and Applications of Ontology: Computer Applications, pp. 231–243 (2010)
6. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
7. Martineau, J.: Identifying and Isolating Text Classification Signals from Domain and Genre Noise for Sentiment Analysis. PhD thesis, University of Maryland, Baltimore County, Computer Science and Electrical Engineering (December 2011)
8. Martineau, J., Finin, T., Joshi, A., Patel, S.: Improving binary classification on text problems using differential word features. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 2019–2024. ACM (2009)

9. Paltoglou, G., Thelwall, M.: A study of Information Retrieval weighting schemes for sentiment analysis. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1386–1395. Association for Computational Linguistics (2010)
10. Pan, S., Ni, X., Sun, J., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th International Conference on World Wide Web, pp. 751–760. ACM (2010)
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. Association for Computational Linguistics, Morristown (2002)
12. Strapparava, C., Valitutti, A.: Wordnet-affect: an affective extension of wordnet. In: Proceedings of LREC, vol. 4, pp. 1083–1086 (2004)

Pre-release Box-Office Success Prediction for Motion Pictures

Rohit Parimi and Doina Caragea

Computing and Information Sciences,
Kansas State University,
Manhattan, KS, USA 66506
{rohitp,dcaragea}@ksu.edu
http://www.cis.ksu.edu

Abstract. In the recent past, machine learning algorithms have been used effectively to identify interesting patterns from volumes of data, and aid the decision making process in business environments. In this paper, we aim to use the power of such algorithms to predict the pre-release box-office success of motion pictures. The problem of forecasting the box-office collection for a movie is reduced to the problem of classifying the movie into one of several categories based on its revenue. We propose a novel approach to constructing and using a graph network between movies, thus alleviating the movie independence assumption that traditional learning algorithms make. Specifically, the movie network is first used with a transductive algorithm to construct features for classification. Subsequently, a classifier is learned and used to classify new movies with respect to their predicted box-office collection. Experimental results show that the proposed approach improves the classification accuracy as compared to a fully independent setting.

Keywords: Transductive Approach, Classification, Motion Pictures, Business Intelligence, Features.

1 Introduction

Movies have become an integral part of our lives as a means of relaxation and entertainment. Movies have also been a significant medium for culture exchange between different countries and regions and are thus an indispensable asset to the world. Given this, the movie industry has become a business and it has huge market profit and potential [9]. As a consequence, the knowledge and research about the movie industry is becoming deeper. Ability to accurately predict the box-office returns for a movie will help the cinema line determine the propaganda cost and period of showing the movie to maximize the profit.

The problem of predicting the box-office gross of a pre-release motion picture has been widely tackled in the past from a statistical point of view. There are many factors influencing the box-office of a movie, for example, number of screens for the movie, advertising, time of the year, number of movies that are released

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 571–585, 2013.

and so on, making the problem challenging. Some of the prior work on this problem was aimed at identifying features that influence the outcome of a movie and finding if they are positively or negatively correlated to the outcome [3].

With the success of machine learning algorithms in improving managerial decision making, researches have started using techniques to build predictive models when addressing the problem of predicting the box-office gross of a movie. Sharda et al. [1] have reviewed the past research on this problem and re-introduced the problem from a machine learning perspective. In their work, Sharda et al. [1] addressed the gross prediction problem by converting it to a classification problem and building a predictive model based on artificial neural networks. They analyzed 7 different features that influence a movie's gross and used them as inputs to a multilayer perceptron neural network. The output from the model is one of 9 classes (selected based on outcome range) to which the movie might belong. In their approach, the authors make the assumption that each movie is independent of the other movies - a basic assumption made by traditional classification algorithms.

However, in the case of movie gross prediction problem, movies are generally not independent. In fact, there is an underlying graph structure that we could identify among movies. For example, a movie can be connected to another movie if they share actors and/or directors, if they have the same genre, if one is a sequel to the other, or if they are released around the same time. If we consider common actors or directors, the intuition is that the reputation of an actor or a director who worked in a movie can be transferred to a different movie in which the actor or the director took part. Thus, we believe that the reputation of *Steven Spielberg* as the director of a yet to be released movie will have positive effect on the success of that movie compared to the success of a yet to be released movie directed by a rookie director.

Traditional classification models assume data instances are independent and identically distributed (i.i.d.) and fail to capture dependencies among instances, in our case movies. To address this limitation, there has been some prior work in the area of link-based classification. Getoor et al. [10] emphasized the importance of link information for classification and proposed a framework to model link distributions. Neville et al. [11] presented a relational Bayesian classifier with different estimation techniques to learn from linked data. Parimi et al. [12] addressed the link prediction problem in *LiveJournal* social network by combining link information with user interest features. Zhu et al. [5] proposed a matrix factorization technique to capture the structure of the graph for web-page classification. The success of the prior work in using link information for classification motivated us to construct a movie dependency network when addressing the gross prediction problem. We use the matrix factorization approach proposed by Zhu et al. [5] to generate network-based features for classification.

The main contributions of our work are as follows: a) an approach to create a graph network that captures dependency relations among movies; b) a custom weighting scheme to compute the weights on the edges; c) generation of features from the network; d) experimental results on a movie dataset showing that the best performance is obtained when movie independent features are combined with dependency features extracted from the network.

The rest of the paper is organized as follows: Section 2 describes prior work in the areas of movie gross prediction and link-based classification. We provide the details of the data used in this work and its categorization in Section 3. Section 4 presents the details of our methodology, specifically the relational setting, the baseline approach and the experimental design. In Section 6, we explain the results of the experiments. Finally, Section 7 discusses the overall contribution of this study and future research directions.

2 Related Work

Much better marketing strategies can be designed and better choices can be made by the cinema production companies in the presence of a strong estimator of a movie's anticipated success. With this in mind, researchers in the past have tried to identify factors that influence the success of a movie and computed correlations between those variables and a movie's box-office gross. Moon et al. [3] used the information from the entire lifetime of a movie to improve their gross predictions over time. One factor that they considered was the word-of-mouth, as it can be indicative of the demand associated with a movie. In addition, they identified correlations between critic ratings and advertisement, and the movie revenue. Their results show that the opening weekend collections for a movie are the strongest estimators for the lifetime gross of the movie. However, it is worth noting that the problem of predicting the gross before a movie's release, that we address, is harder than extrapolating the gross based on first week collections.

Sharda et al. [1] approached the movie gross prediction problem from a machine learning perspective. They reviewed past research on the variables influencing the success of a movie and used seven of those in their work. They converted the problem of predicting the revenue of a movie into the problem of classifying a movie based on revenue ranges. For example, a movie can be considered a *flop* if its revenue is in the range *100,000 to 200,000* and a *blockbuster* if its revenue is in the range *5 million to 10 million.* A multilayer perceptron network is used to classify a movie's gross in one of 9 possible classes. However, a drawback of this approach is that it assumes movies to be independent from each other.

The work by Zhang et al. [4] addressed the box-office prediction problem using a multilayer back-propagation neural network. Similar to the work by Sharda et al. [1], Zhang et al. [4] identified 11 input variables to the neural network based on market survey. The weights for the neural network model are selected using statistical methods to maximize the accuracy of the model. Even though the accuracy of the proposed model is better than the accuracy of the model proposed by Sharda et al. [1], the dataset used in this work is rather small, consisting of only 241 movies classified according to 6 classes.

Recent research in link-based classification has focused on ways in which inherent dependency relations between instances can be used to improve results of traditional learning algorithms, which assume instances to be independent. Many techniques have been proposed in the past to exploit the graph structure of particular problems. Getoor et al. [10] have proposed a framework to capture correlations among links using link distributions. They used an iterative classification algorithm, which combines both link information and information intrinsic to instances (e.g., web-page content) for classification. The authors have applied this approach to web and citation collections and reported that using link distribution improved accuracy in both cases. The work by Parimi et al. [12] combined content features, obtained by modeling user interests using LDA, with graph features to predict friendship links in *LiveJournal* social network. Zhou et al. [8] proposed a framework that uses the adjacency matrix of the graph for propagating labels in the graph. However, this technique like other techniques outlined above, cannot be used to propagate labels in a directed graph.

Zhou et al. [6], [7] have studied techniques, which aim at exploiting the structure of the graph globally rather than locally, thus ensuring that the classification is consistent across the whole graph. The approach proposed in [6] is motivated by the framework of *hubs* and *authorities*, while the work proposed in [7] is inspired by the *PageRank* algorithm used by Google search engine. Although the techniques proposed in [6] and [7] are applicable for classification in directed graphs, they rely solely on link structure and ignore content information (e.g., content of a web-page). To address this limitation, Zhu et al. [5] proposed an algorithm to jointly factorize the content information and link information (represented as weighted edges in a directed graph) in a supervised setting, and showed that this joint factorization improves the classification accuracy compared to just using link information.

We plan to take advantage of the approach proposed in [5] to factorize the link information from the movie graph that we construct, and hypothesize that the factors obtained as a result of this model capture the similarity between movies better than measures that use only the content. In addition, the dimensionality of the problem is reduced when representing each instances by its factors.

3 Data

The dataset that we used in this work consists of 977 movies released as ‘wide releases’ between the years 2006 and 2011. There are approximately 150 movies in each year and for each movie we collected features such as actor and director profiles (to compute star value), genre, release date, sequel information, budget, runtime, number of theaters and MPAA rating. All these features, from movie information to actor and director profiles are collected from a well-known site: Box Office Mojo. We selected these features based on the work by Sharda et al. [1], and grouped them into two categories based on how we use them in our proposed approach. Specifically, we use the budget, runtime, number of theaters and MPAA rating as features intrinsic to a movie (i.e., content features), while actors, directors, genre, release date and sequel information features are used to construct dependency relations between movies, or more precisely weighted edges in the movie graph (i.e., link features).

Intuitively, the *budget, runtime, number of theaters* and *MPAA rating* features represent information particular to a movie and directly contribute to the revenue of the movie, independent of other movies. These features can be seen as content features and are directly used when training the model (without additional processing or transformation). As opposed to these features, link features capture dependencies between features and are used to form a directed graph (precisely, weighted edges) between movies. They influence the movie gross indirectly, by the means of graph features extracted from the graph. More intuition behind using the dependency features is provided in what follow.

The *actors* and *directors* of a movie, in general, have a popularity index associated with them. The popularity that an actor or a director achieves through the success of their movies creates a positive sentiment for their up-coming movies, affecting the revenue of that movie. This is precisely what we want to capture when using actors and directors to construct dependencies between movies, i.e. weighted edges in the movie graph. Other dependency features like *genre* and *sequel* are meant to capture the percentage of audience that are loyal to that genre or franchise, respectively, when used in the graph network. For example, a person who is mostly inclined towards movies with genre ‘thriller’, would most likely watch the awaited thriller movies rather than dramas or comedies. Similarly, a person who likes movies like *Sherlock Homles-1* and *Sherlock Holmes-2* would most likely watch the upcoming movie *Sherlock Holmes-3.* The feature *release date* captures the competition that a movie might have to face when it is released. It is said by the industry experts that revenue, in general, is likely to be divided among the movies released at the same time. We should note that the feature *MPAA rating* can also be categorized as a dependency feature, but the presence of only 4 different kinds of ratings (G, PG, PG-13, R) will result in too many links in the graph network. Also, we believe that, unlike genre, MPAA rating will not be useful in capturing the interest of audience towards a particular rating. Hence, it is just used as a movie content feature.

4 Approach

As mentioned above, our objective is to design an approach that can take advantage of link features (capturing dependencies between movies), in addition to content features, for the task of predicting box-office revenues of movies before their theatrical release. In this section, we will describe in detail the construction of a dependency graph between movies, several weighting schemes used for computing weights on the edges of the graph and an algorithm to capture the structure of the graph. We will also present the baseline model against which our approach will be compared.

4.1 Relational Setting

Graph Construction and Weighting Schemes: In Section 3, we categorized features as link features (constructed based on the dependency graph) and independent features. Here, we explain how the movie graph is constructed using link features. Two movies in the dataset are connected if they:

1. have a common actor
2. have a common director
3. have the same genre
4. are sequels
5. are released around the same time (e.g., two weeks apart)

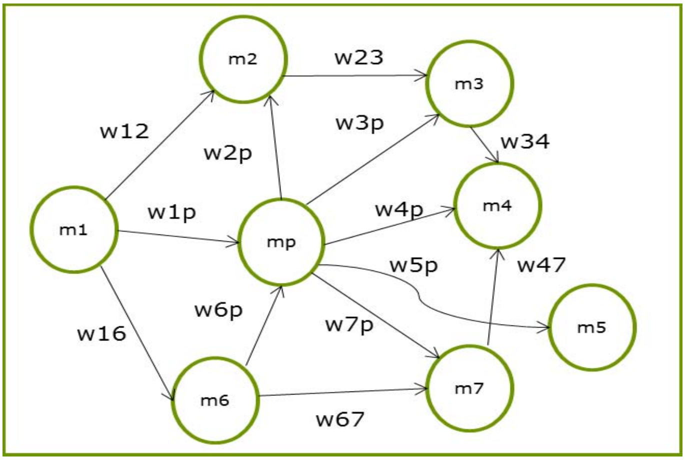


Fig. 1. Example of a movie graph created using dependency features

Figure 1 depicts an example movie graph that can be constructed using the dependency features. As seen in Figure 1, each link has a weight term which can be seen as the similarity between the two connected movies. In this work, we experiment with three different weighting schemes to see which one represents our data better. We only use dependency features in weighting schemes 2 and 3. The three schemes are described as follows:

1. **A Constant Weight '1'**: In this weighting scheme each link in the graph is assigned a constant weight value '1' without considering how similar the nodes connected by the edge are. This is a simple weighting scheme but has the disadvantage of not capturing the similarity between the nodes.
2. **Radial Basis Function (RBF) Kernel**: The weight corresponding to an edge in the graph is given by the following kernel function:

$$k(x, x') = exp(-\left\| x - x' \right\|^2 / 2\sigma^2) \quad (1)$$

where x and x' are the feature representations of the nodes connected by the edge and the features are computed in a way similar to those in the independent setting (Section 4.2). Equation 2 captures the similarity between the two nodes.

3. **Custom Weighting Scheme**: Even though weights from RBF kernel capture the similarity between the nodes, they cannot be used to capture the negative effect from the feature ‘Competition’ or the positive effect of ‘Sequel’. We shall consider two examples to understand the disadvantage of using an RBF kernel. The headers ‘1’, ‘2’, ‘3’ and ‘4’ in Tables 1 and 2 correspond to the features: ‘Star Value’, ‘Genre’, ‘Sequel’ and ‘Competition’, respectively. In the example in Table 1, the two movies are clearly connected as they are sequels and hence are very similar to each other. However, we will have a weight value less than 1 for the edge connecting these two movies because of the difference in the values for feature ‘sequel’. Consider the example in Table 2 and assume that the two movies are connected based on release date. The edge connecting these two movies will have a weight value of ‘1’ if RBF kernel is used because of identical values for all the features. Even though the movies in Table 2 are less similar than those in Table 1, the weight values from RBF kernel suggest otherwise. Because of this disadvantage, we designed a custom weighting scheme for the movie domain.

   In this weighting scheme, the weight for a link between two movies is determined by linearly combining the dependency features with coefficients as shown in Equation 2. The intuition of weights on the graph in weighting schemes 1 and 2 is that they represent similarity between the two nodes connected by the edge. However, in this weighting scheme, the intuition of weight has changed from similarity between the nodes to the positive/negative effect that can be transferred from one node to other node. The feature *Release-Date* in Equation 2 takes a value ‘1’ if the two movies are connected based on release date and ‘0’ if not. Other features in Equation 2 are computed in a way similar to those in independent setting (Section 4.2). Optimal values for the coefficients can be obtained either by using a validation set or by trial and error based on domain knowledge. Because of small number of movies in our dataset, we determined the coefficients by trial and error.

Table 1. Example 1, depicting the disadvantage of RBF Kernel

Title	1	2	3	4
Batman Begins	High	Action	0	A
The Dark Knight	High	Action	1	A

Table 2. Example 2, depicting the disadvantage of RBF Kernel

Title	1	2	3	4
IronMan II	High	Action	1	A
The Dark Knight	High	Action	1	A

$$w12 = a1 * (ActorValue) + a2 * (DirectorValue) + a3 * (Genre) + a4 * (Sequel) - a5 * (ReleaseDate) \quad (2)$$

Classification in Directed Graphs: Given a network graph with nodes and weights on edges, the objective is to find the class label for the nodes which do not have any class label. For example, in Figure 1, if we know the revenues for the movies *m1* through *m7*, we would want to find out the revenue for the movie *mp*. As explained in Section 2, there has been some work in the past to solve problems that match the above criterion. In this work, we use the algorithm proposed by Zhu et al. [5] because of its ability to combine content information into the model, and at the same time, reduce the dimensionality of the problem. This algorithm falls into the category of transductive algorithms in machine learning. Transductive algorithms are algorithms which are trained on specific training instances to reason on specific test instances. Suppose we add a new node to the graph, we would have to run the algorithm again to predict the revenue for the newly added node. The algorithm that we use in this work uses the matrix factorization technique to factorize the adjacency matrix in a graph network. The factors generated as a result of this factorization can be seen as link features and can be further used as features for classification. The factorization is given by the following equation:

$$\min_{Z,U} \left\| A - ZUZ^\top \right\|_F^2 + \gamma \left\| U \right\|_F^2 \tag{3}$$

where,

- 'A' is the adjacency matrix represented by weights
- 'Z' is an $n \times l$ feature matrix, n = #instances, l = #features
- 'U' is an $l \times l$ matrix
- $\|.\|_F$ is the Frobenius norm.

Equation 3 can be solved using optimization techniques such as *gradient descent*. The intuition behind Equation 3 is to approximate the 'A' matrix using the product $ZUZ^\top$ and as a result, we obtain link features in the 'Z' matrix which can be used to represent each instance (movie in our case).

The pseudo-code to obtain features using the above technique is as follows:

1. Construct the graph using the link features.
2. Use one of the weighting schemes to fill the adjacency matrix with weights.
3. Run the above matrix factorization algorithm to get 'Z' matrix.
4. Use the 'Z' matrix to represent each movie for classification.

Movie Representation: As explained earlier, each movie in the dataset is represented using the features obtained by factorizing the adjacency matrix. These features are henceforth referred to as link or graph features. The number of link features used to represent a movie (i.e., the number of factors) is decided using a validation set. Other features such as movie independent features can also be appended to the link features. Once all the movies in the dataset are represented either using just link features or link + movie independent features, we build predictive models to test our hypotheses. An example representation for the movie 'The Dark Knight' is shown in Equation 4. The movie independent features are appended to the graph features in this example.

$$TheDarkKnight = f_1 \cdots f_l, Budget, \#Theaters, Time, MPAA, Class \tag{4}$$

4.2 Baseline: Independent Setting

Many researchers in the past have assumed that movies are independent when addressing the gross prediction problem. In this setting, we use this independence assumption between the movies, construct features and build predictive models to classify movies into gross ranges. This approach is identical to the one described in Sharda et al. [1] and thus serves as a baseline for our approach.

Feature Construction: The features that are used to build predictive models in the independent setting are the following:

1. **Star Value**: The star value for a movie is contributed by the actors and directors of that movie. We collected information about top 5 actors and all the directors for a movie and crawled the actor/director profile from the site Box Office Mojo. The value an actor or a director contributes is determined by averaging the gross for all the movies (released) the actor or director took part in. The overall value from the actors/directors for a movie is the average of values for the actors/directors. The star value for a movie is obtained by taking a weighted average of the values contributed by all the actors and the directors (depicted in Equation 5). The coefficients are set as 0.7 for actor value and 0.3 for director value based on past research which indicates that the director value for a movie is not as significant as the actor value for that movie. We used three independent binary variables to represent the degree of star value in our model: 'High', 'Medium', and 'Low' values, by discretizing the star value computed. Based on our calculations, a movie is assigned a star value 'High' if the value from Equation 5 is greater than 65 million, 'Medium' if the value is between 25 million and 65 million and 'Low' if the value is less than 25 million.

$$StarValue(m) = 0.7 * ((a1 + a2 + a3)/3) + 0.3 * ((d1 + d2)/2) \quad (5)$$

2. **Sequel**: Similar to other prior studies, we used a binary variable to determine whether a movie is a sequel or not. Our intuition is that the sequels are positively correlated with the success of a movie as they are filmed, because of the success of the previous versions of that movie. The feature takes the value '1' if the movie is a sequel and '0' if the movie is not.
3. **Competition**: We used this feature to capture the competition that a movie faces from other movies that are released around the same time. In a study by Moon et al. [3], it is reported that the pool of entertainment dollars is shared between the movies that are released around the same time. Many studies in the past have found release date to be an important contributor to a movie's box-office success and it can be used to capture the level of competition. The feature competition is expected to negatively influence the success of a movie. We represented competition using the following values: 'High', 'Medium' and 'Low'. A value 'High' indicates high competition, a value 'Medium' indicates medium and a value 'Low' indicates low competition for a movie. Based on

the release dates of the movies in our dataset, we assign ‘High’ to the movies released in ‘January’, ‘August’, ‘September’ or ‘October’; ‘Medium’ to the movies released in ‘February’, ‘March’, ‘April’, ‘November’, or ‘December’ and ‘Low’ to the movies released in ‘May’, ‘June’ or ‘July’.

4. **Genre**: Most of the past work has identified genre as a content category determiner and used it in their work even though it is rarely found to be significant. We followed the convention and used it as a feature. Eighteen different genres are used to tag a movie and we allowed a movie to be tagged with more than one genre value. The tags we used are shown in Table 3.
5. **Budget**: In the recent past, budget for a movie seems to be one of the features highlighted in the promotions for a movie with the objective of attracting more people to watch the movie. We believe that this feature contributes to the success of a movie and is positively correlated with it. We used a positive integer to represent this feature.
6. **Run Time**: This feature is represented as a continuous variable and captures the length of the movie.
7. **Number of Theaters**: Previous work for solving this problem showed correlations between a movies’ financial success and the number of screens it is released in. We represent this feature as a continuous variable indicating the number of screens a movie is scheduled to be shown at its opening.
8. **MPAA Rating**: A commonly used variable in predicting the gross for a movie, which takes the values ‘G’, ‘PG’, ‘PG-13’ and ‘R’.

Table 3 summarizes all the features used in the independent setting and the possible values that each feature may take. An example of how the movie ‘The Dark Knight’ is represented in the independent setting is shown in Table 4. Table 5 depicts the gross ranges for each of the classes used in this work. This is the same for both independent setting and dependent setting.

Table 3. Summary of features and the values they take in the independent setting

Competition	StarValue	Sequel	Genre	RunTime	Budget	Theaters	MPAA
High Medium Low	High Medium Low	1 0	Period, Crime, Action, Romance, Thriller, Family, Historical, Sci-Fi, Horror, Drama, Comedy, Sports Fantasy, Music, War, Animation Documentary Adventure	RunTime for the movie in minutes	Budget for the movie in dollars	Positive integer	G PG PG-13 R

Table 4. Representation of **'The Dark Knight'** in independent setting

Competition	StarValue	Sequel	Genre	RunTime	Budget	# Theaters	MPAA	Class
C	High	1	Action	150	185000000	4366	PG-13	9

Table 5. Discretization of the movie gross into 9 classes

Class No	1	2	3	4	5	6	7	8	9
Range (in Millions)	< 10 (flop)	> 10 < 20	> 20 < 30	> 30 < 45	> 45 < 70	> 70 < 100	> 100 < 150	> 150 < 225	> 200 (blockbuster)

5 Experimental Design

As explained in Section 3, the dataset that we used in our experiments consists of 977 movies released as 'wide releases' between the years 2006 and 2011. Movies released between the years 2006 and 2010 are considered as training instances and movies released in the year 2011 are considered as test instances. This will ensure that we use information from the past to predict the gross for future movies. Features are constructed for the movies in the training set and test set for the relational setting and independent setting, as described in Section 4.1 and Section 4.2, respectively. Model parameters and number of graph features, in case of the relational setting are tuned using a validation set constructed using movies between the years 2006 and 2010.

5.1 Research Questions

Our experiments have been designed to address two main research questions:

- Which weighting schemes is better in terms of prediction accuracy?
- How does the relational setting compare with the independent setting?

5.2 Experiments

To answer the above questions, we have designed the following experiments:

1. **Experiment 1**: In this experiment, we test the performance of predictive models trained on features constructed using the independent setting (Section 4.2). This experiment will be referred to as exp_1 henceforth.
2. **Experiment 2**: We ran two variants of this experiment in which we test predictive models trained on graph features constructed using weighting scheme 1 (Section 4.1). In the first variant which will be called exp_2_0 henceforth, we use just the graph features to build the models. In the second variant which will be called exp_2_1, we add the movie independent features (*Budget, MPAA Rating, No. Theaters* and *Runtime*) to the features used in exp_2_0.

3. **Experiment 3**: In this experiment, we build models using graph features constructed using weighting scheme 2. Similar to experiment 2, we have exp_3_0 and exp_3_1.
4. **Experiment 4**: The graph features in this experiment are constructed using the weighting scheme 3. Similar to the above two experiments, we have exp_4_0 which uses just graph features and exp_4_1 which uses graph features along with movie independent features. The values for the coefficients in Equation 2 are: a1=0.55, a2=0.25, a3=0.2, a4=1, a5=1.

For all the experiments, we used Weka implementations of the Logistic Regression and Random Forest algorithms.

5.3 Evaluation Criteria:

To evaluate the results of the experiments outlined above, we have used the following metrics:

1. **Bingo Accuracy**: Also known as BINGO or simply accuracy, it is defined as the ratio between the number of instances correctly classified and the total number of instances.
2. **AUC**: Area under the ROC curve, or AUC, is one of the popular metrics used to evaluate prediction models. For each experiment, we report the weighted average AUC value, as output by Weka.

6 Results and Discussion

6.1 Comparison between Different Weighting Schemes

As mentioned earlier, experiments have been conducted to test which of the weighting schemes better capture the information from the features used in the dependency setting. As expected, weights generated using the custom weighting scheme produced better results for the evaluation metrics in most of the cases. This can be seen from Tables 6 and 7. Custom weighting scheme has better AUC and accuracy values compared to the AUC and accuracy values from the other two weighting schemes for the random forest classifier when using just the graph features (Table 6) or when the movie independent features are appended to the graph features (Table 7). For the logistic regression classifier, both AUC and accuracy values are better compared to the AUC and accuracy values from the other two weighting schemes when we append the graph features to the movie independent features. Hence, it is evident from Tables 6 and 7 that the classification accuracies of the predictive models built using the custom weighting scheme are better than those built using a constant weighting scheme or weights generated using an RBF kernel.

Table 6. AUC and accuracy values for logistic regression and random forest classifiers trained on graph features constructed using different weighting schemes. Best AUC and accuracy values, across all the experiments, for each classifier are highlighted.

Exp	Metrics	Logistic Regression	Random Forest
exp_2_0	AUC	0.538	0.536
	ACC	15.07	15.75
exp_3_0	AUC	**0.596**	0.559
	ACC	21.23	16.44
exp_4_0	AUC	0.585	**0.598**
	ACC	**24.66**	**21.92**

Table 7. Similar to Table 6, AUC and accuracy values for classifiers trained on movie independent features appended to the graph features. Best AUC and accuracy value across all the experiments, for each classifier are highlighted.

Exp	Metrics	Logistic Regression	Random Forest
exp_2_1	AUC	0.702	0.667
	ACC	26.03	20.55
exp_3_1	AUC	0.673	0.729
	ACC	21.92	27.4
exp_4_1	AUC	**0.735**	**0.732**
	ACC	**33.56**	**32.88**

6.2 Comparison between Independent and Relational Settings

Table 8 depicts the comparison between the results for the independent setting and the dependent setting for logistic regression and random forest classifiers. The proposed approach of using a dependency relation between the movies was able to improve the accuracy of the predictive models and is very close in-terms of the AUC metric compared to the independent setting. The reason for a lower AUC value in relational setting for the predictive models compared to those in independent setting might be because we have optimized the accuracy metric during the validation experiments to get the number of graph features to use and to tune the model parameters, ridge in case of logistic regression and number of trees for the random forest classifier.

Table 8. AUC and accuracy values for logistic regression and random forest classifiers in the independent and relational settings, using the three weighting schemes. Best AUC and accuracy values across all the experiments, for each classifier, are highlighted.

Exp	Metrics	Logistic Regression	Random Forest
exp_1	AUC	**0.748**	**0.745**
	ACC	29.45	26.71
exp_2_1	AUC	0.702	0.667
	ACC	26.03	20.55
exp_3_1	AUC	0.673	0.729
	ACC	21.92	27.4
exp_4_1	AUC	0.735	0.732
	ACC	**33.56**	**32.88**

As expected, better results for the accuracy are achieved using the custom weighting scheme and we hypothesize that tuning the coefficients a1, a2, a3, a4 and a5 from Equation 2 might improve the results further. It is also evident that the results shown in Table 8 are much better that the results from a random classifier which will have an accuracy of 11.11% (1/9).

7 Conclusions and Future Work

We analyzed the problem of predicting the revenue of a movie before its theatrical release and identified several factors influencing the same. We designed an approach to construct a dependency network for the movies in the dataset and worked on the application of a transductive algorithm to predict the missing labels for the nodes in the graph. We have designed three different weighting schemes to represent the network using the adjacency matrix and experiments are conducted using all the weighting schemes to determine which is effective. It is evident from Tables 6 and 7 that the custom way of generating weights produced better results compared to the other two weighting schemes. Our hypothesis that considering a dependency relation between movies helps improve the prediction accuracy is confirmed by the results in Table 8. The results also show that the AUC and accuracy values of the predictive models are improved when the movie independent features are appended to the graph features.

As an extension to the work described in this paper, we would like to study the influence of social media on a movie's success. Our intention is to capture the word-of-mouth effect or the demand for the movie using social media. There has been some work in the recent past along the lines of using social media data to predict movie ratings and box-office gains. Asur et al. [13] used 'Twitter' social media to collect tweets related to movies and analyzed their influence on a movie's revenue. They concluded that the rate of tweets on a movie has a positive influence on the movie's success. Moreover, sentiments extracted from tweets further improved the predictions. Wong et.al. [14] also used 'Twitter' data to predict the rating as well as box-office gains for a movie, by doing sentiment analysis on the tweets. Encouraged by the results published in [13] and [14], we plan to test how informative will the data from 'Twitter' be, for the movies in our dataset. To accomplish this, we developed a framework to query and retrieve tweets about movies from a popular search engine for *Twitter* called Topsy. The number of unique users and total tweets can be used to capture the percentage of audience interested in the gossip about the movie and the sentiment of a tweet can be used to know if a user likes or dislikes the movie. We hypothesize that the features constructed from the social media data will further improve the classification accuracy and be useful in answering the question: 'How predictive are the features from different data sources?'

Acknowledgements. We would like to thank Dr. Shenghuo Zhu for sharing the code of his algorithm.

References

1. Sharda, R., Delen, D.: Predicting box-office success of motion pictures with neural networks. Expert Systems with Applications 30, 243–254 (2006)
2. Zhang, L., Luo, J., Yang, S.: Forecasting box office revenue of movies with BP neural network. Expert Systems with Applications 36, 6580–6587 (2009)
3. Moon, S., Bergey, K.P., Lacobucci, D.: Dynamic effects among movie ratings, movie revenues, and viewer satisfaction. American Marketing Association (2010)
4. Zhang, W., Skiena, S.: Improving movie gross prediction through news analysis. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (2009)
5. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2007)
6. Zhou, D., Schölkopf, B., Hofmann, T.: Semi-supervised learning on directed graphs. In: Proceedings of Neural Information Processing Systems (2005)
7. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005 (2005)
8. Zhou, D., Bousquet, O., Navin, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Proceedings of Advances in Neural Information Processing Systems, vol. 16 (2004)
9. Shanklin, W.: What businesses can learn from the movies. Business Horizons 45(1), 23–28 (2002)
10. Geetor, L., Lu, Q.: Link-based Classification. In: Twelth International Conference on Machine Learning, ICML 2003, Washington DC (2003)
11. Neville, J., Jensen, D., Gallagher, B.: Simple Estimators for Relational Bayesian Classifiers. In: Proceedings of the Third IEEE International Conference on Data Mining, ICDM 2003 (2003)
12. Parimi, R., Caragea, D.: Predicting friendship links in social networks using a topic modeling approach. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 75–86. Springer, Heidelberg (2011)
13. Asur, S., Huberman, B.A.: Predicting the future with social media. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (2010)
14. Wong, F.M.F., Sen, S., Chiang, M.: Why watching movie tweets won't tell the whole story? In: Proceedings of the 2012 ACM Workshop on Online Social Networks, WOSN 2012, pp. 61–66. ACM, New York (2012)

Shifting Concepts to Their Associative Concepts via Bridges

Hongjie Zhai[1], Makoto Haraguchi[1], Yoshiaki Okubo[1],
Kiyota Hashimoto[2], and Sachio Hirokawa[3]

[1] Graduate School of Information Science and Technology, Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
{makoto,zhaihj,yoshiaki}@kb.ist.hokudai.ac.jp
[2] College of Sustainable System Sciences, Osaka Prefecture University
1-1 Gakuen-cho, Nakaku, Sakai, Osaka 599-8531, Japan
hash@kis.osakafu-u.ac.jp
[3] Research Institute for Information Technology, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan
hirokawa@cc.kyushu-u.ac.jp

Abstract. This paper presents a pair of formal concept search procedures to find associative connection of concepts via bridge concepts. A bridge is a generalization of a sub-concept of an initial concept. The initial concept is then shifted to other target concepts which are conditionally similar to the initial one within the extent of bridge. A procedure for mining target concepts under the conditional similarity with respect to the bridge is presented based on an object-feature incident relation. Such a bridge concept is constructed in the concept lattice of person-feature incident relation. The latter incident relation is defined by aggregating the former document-feature relation to have more condensed relation, while keeping the variation of possible candidate bridges. Some heuristic rule, named Mediator Heuristics, is furthermore introduced to reflect user's interests and intention. The pair of these two procedures provides an efficient method for shifting initial concepts to target ones via some bridges. We show their usefulness by applying them to Twitter data .

Keywords: associative search, bridge concept, mediator, conditional similarity.

1 Introduction

In the studies of Information Retrieval, a kind of technique, called "Associative Search" [1], is sometimes used to shift queries during the search processes for documents or messages. The purpose of shifting queries is to suggest feature terms and documents not being properly expressed by the initial query, reminding users of other features which may be related to user's interests or intention. This paper aims at formalizing those processes in terms of formal concepts [5] and to provide an efficient mining procedure to enumerate target concepts associative with the initial one, enjoying the structures of formal concept lattices.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 586–600, 2013.

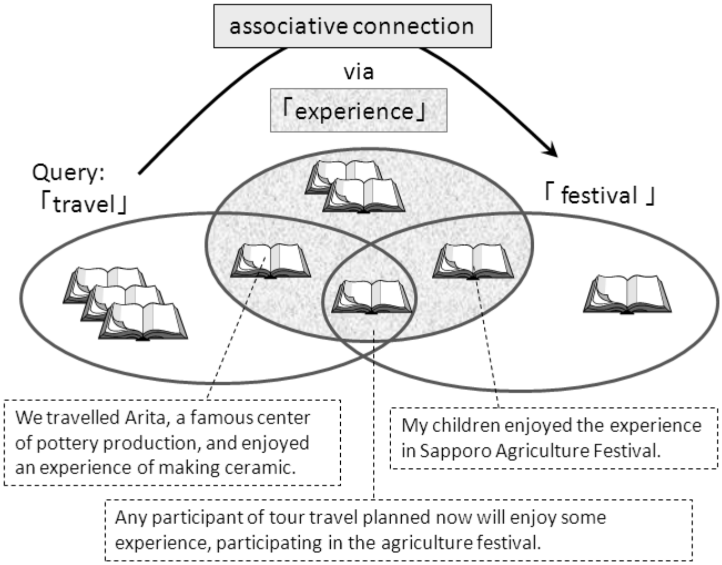

associative connection
via
「experience」
Query:
「travel」
「 festival 」
We travelled Arita, a famous center of pottery production, and enjoyed an experience of making ceramic.
My children enjoyed the experience in Sapporo Agriculture Festival.
Any participant of tour travel planned now will enjoy some experience, participating in the agriculture festival.

Fig. 1. Example of Associative Connection of Concepts

Fig. 1 illustrates an example of associative connection between two concepts, “travel” and “festival”, where the former is considered as the initial one, and ovals in the figure denote sets of documents with the feature terms as “travel”. At least some documents about “travel” involve another feature term, “experience” for instance, that may play a role of bridge for “travel” and “festival”. In other words, the associative connection is supported by such documents about “travel” with the term “experience” that form a sub-concept of “travel”. A bridge concept “experience” guiding us to “festival” can be constructed by dropping “travel” from the sub-concept of “travel” with “experience”. The dropping operation is performing generalization of concepts. As is well known, the framework of formal concepts is suited to perform such a construction based on Galois connection between feature terms as attributes and documents as objects. Particularly, specialization (generalization) to obtain sub-concepts (super-concepts) is naturally represented in concept lattices.

In the problem of Web navigation [13], a similar idea for navigating concept search in a concept lattice is presented, guiding related concepts along a lattice structure. Namely the basic operations are those for generalization, specialization and for taking sibling concepts in the lattice. On the other hand in this paper, under a bridge concept, which is also constructed by specialization and generalization but is much more constrained as we see later, target concepts reachable from the initial one are never our solution as long as they are not related to some bridge concept. Thus we make stronger structural constraint for the targets.

Major issues we have to solve in making bride concepts and associative connections based on them are listed as follows.

Selection of Candidate Bridges (SCB) :
There exist many sub-concepts of initial one. Even if we choose some of them, we have many ways to generalize those sub-concepts to get more general super-concepts as bridges.

Selection of Target Concepts (STC) :
Suppose some bridge concept is successfully constructed from the initial one. Even under such an assumption, several possible concepts associated with the initial one via the bridge are there. Some criterion or condition checking which target concepts are potentially associative under the presence of bridge will be needed.

The STC problem can be solved more directly than SCB by enumerating candidate concepts meeting a similarity constraint that they must be similar to the initial concept w.r.t. a given bridge. Thus we regard STC as a mining task instead of selection task. In the case of Fig 1, although “travel” and “festival” may not be similar so much, their extents (the document sets) would become closer when we restrict documents to those of “experience”. Thus we evaluate how object sets are overlapping each other within the extent of bridge. Our similarity is conditioned by bridge concepts. Several similarity measures for formal concepts are proposed in the literatures. For instance, in [12], similarity between concepts is defined by considering the overlappingness of their feature sets and object sets at the same time. However, we need only similarity showing how the object sets are overlapping. For this reason, we use bond [4], an extended Jaccard coefficient, to calculate the degree of overlappingness for object sets. In Section 6, we present a procedure to solve STC based on the bond measure conditioned by bridges.

On the other hand, SCB is more critical, since STC search completely depends on solutions of SCB. As is already discussed, bridges are constructed from document objects of some sub-concept of initial one. Some clustering methods, [9,10] for instances, may be applicable to have a family of object sets. From each object set, we would form an intent of bridge by examining and by selecting some of its characteristic feature terms. However, almost clustering methods disregard clusters not meeting their own clustering criterion. Consequently, possible candidate bridges are restricted.

To the contrary, in this paper, we regard the variation of possible clusters of document objects. As we have more candidate object sets from which bridges are constructed, users have more chances to become aware of them and to change their initial concepts based on them. It is however computationally hard to keep all of them because of a huge number of object subsets or sub-concepts of initial one.

In order to relax this kind of dilemma, we here propose to use other information about documents or messages, the information of writers or senders. With each object (document or message), we suppose just one person is associated.

Then we can build a person-feature relation from the object-feature relation by merging objects each person has. The variation of features possessed by objects is aggregated to those persons have. As the number of persons is much smaller than the number of document objects, we examine person sets, instead of document sets, from which some bridge concepts are drawn more efficiently.

The extraction of bridges from person sets is performed in the corresponding concept lattice for the person-feature incident relation. In order to make the process more efficient, we introduce the following heuristic:

Mediator Heuristics:

When we wonder which direction of associative connection is better to be developed, we normally ask some persons who talk about various areas of documents and feature terms. We call here such persons mediators. As more number of features or topics they are talking about, we regard them as better mediators.

Using a membership function of fuzzy k-means clustering [3], we define a probability distribution on features showing how a person is related to the features, and then calculate its entropy as the degree of mediator. Then, every person is ranked according to the entropy, and the relevance to the initial concept is tested by user. Some persons may be inadequate from the user's viewpoint, while another may be good conversely. We assign negative or positive signs to the former and the latter types of mediators, respectively. Then the target bridge concepts are required to cover positive mediators and not to cover negative ones. Consequently, the possible bridges are to be placed in a sub-lattice with the initial concept as its top and the least common generalization of positive mediators as its bottom. Using the search strategy presented in [2], we show in Section 6 a very efficient search procedure to find candidate bridge concepts in the sub-lattice.

The reminder of this paper is organized as follows. In the next section, we introduce some terminologies. Section 3 discusses a pair of incident relations as datasets we assume. Our user ranking method is presented in Section 4. In Section 5 and 6, we formalize our associative search and present its computation procedure, respectively. In our experimentation in Section 7, we apply our method to data of $500,000$ tweets, where features are compressed as 155 clusters of 912 original feature terms (nouns) to avoid the problem of very sparseness of the data. Such compressed features are also used to build message-feature incident relation. As an instance, we show an associative connections between "smartphone" and "digital book" via "mobile" actually extracted. The paper is concluded in Section 8 with a summary and some future directions.

2 Preliminaries

In this paper, we are concerned with a notion of *concepts* in *Formal Concept Analysis* (FCA) [5]. Let $\mathcal{O}$ be a set of *objects* (or individuals) and $\mathcal{A}$ a set of *attributes* (or features). For a binary relation $R \subseteq \mathcal{O} \times \mathcal{A}$, A triple $(\mathcal{O}, \mathcal{A}, R)$ is called a *formal context.* If $(x, y) \in R$, we say that the object x has the attribute y. Then, for a set of objects $X \subseteq O$, the set of attributes associated with X is denoted by X', that is, $X' = \{y \in \mathcal{A} \mid \forall x \in X, (x, y) \in R\}$, where "$'$" is called a *derivation operator*. Similarly, for a set of attributes $Y \subseteq A$, the set of objects sharing Y is denoted by Y', that is, $Y' = \{x \in \mathcal{O} \mid \forall y \in Y, (x, y) \in R\}$.

For a pair of $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{A}$, (X, Y) is called a *formal concept* (*FC*) under the formal context iff $X' = Y$ and $Y' = X$, where X and Y are called the *extent* and the *intent* of the concept, respectively. For concepts (X_i, Y_i) and (X_j, Y_j), if $X_i \subseteq X_j$ (or $Y_i \supseteq Y_j$), then we say (X_i, Y_i) is a *sub-concept* of (X_j, Y_j).

Since we are concerned with concepts in different formal contexts, the derivation operator in a context $\mathcal{C}$ is often explicitly denoted by $'^{(\mathcal{C})}$ or $''^{(\mathcal{C})}$.

3 Person-Term Relation and Document-Term Relation

In order to formalize our associative search, we assume a *person-term* relation and a *document-term* relation as formal contexts.

Let P be a set of *persons* and V a set of *terms* or *words* as a vocabulary. It is assumed that a *document* is written at the vocabulary V by a person in P. For a document d, the set of terms appeared in d is referred to as $terms(d)$ and the person by whom d is written as $person(d)$.

Given a set of documents D, we can define a document-term relation $R_D \subseteq D \times V$, where $(d, t) \in R_D$ if and only if $t \in terms(d)$. It is clearly represented as a formal context $\mathcal{D} = (D, V, R_D)$.

In addition, a person-term relation $R_P \subseteq P \times V$ can also be defined from D. For the document set D, the set of documents written by a person $p \in P$ is denoted by D_p, that is, $D_p = \{d \in D \mid person(d) = p\}$. Then, for a person p, the set of terms used by p in some document is given by $T_p = \cup_{d \in D_p} terms(d)$. Based on T_p, we can define $R_P \subseteq P \times V$ as $(p, t) \in R_P$ if and only if $t \in T_p$. Thus we have a corresponding formal context $\mathcal{P} = (P, V, R_P)$.

4 Ranking Persons Based on Mediator Level

Let P be a set of persons. We assume that each person in P is assigned a *mediator level.* Intuitively speaking, if a person is related to various kinds of topics, it seems possible for us to associatively access to several topics via the person. In this sense, it would be reasonable to consider that such a person is a good mediator for associative search and given a higher mediator level. On the other hand, if a person is concerned with some particular topic, he/she is assigned a lower mediator level.

4.1 Mediator Level

We consider a person to be a good mediator, if he/she is related to a more number of topics than the others. In order to find such a person, assuming a cluster of terms (words) to be a *topic*, we evaluate a user's mediator level based on how closely related to topics the user is.

For a person-term context $\mathcal{P} = (P, V, R_P)$ and a document-term context $\mathcal{D} = (D, V, R_D)$, the mediator level of a person in P is computed as follows:

1. The person context $\mathcal{P}$ is projected by preserving all terms with $TF\text{-}IDF$ values greater than $(\max + \min) * \alpha$, where *alpha* is a control parameter, and max and min are the maximum and minimum values of $TF\text{-}IDF$. The projected context is denoted by $\tilde{\mathcal{P}} = (P, \tilde{V}, \tilde{R_P})$.
2. Based on $\tilde{V} \subseteq V$, the document context is also projected. The projected context is denoted by $\tilde{\mathcal{D}}$ and defined as $\tilde{\mathcal{D}} = (D, \tilde{V}, \tilde{R_D} = (R_D \cap (D \times \tilde{V})))$. Then we consider its corresponding matrix $M_{\tilde{\mathcal{D}}} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{|\tilde{V}|})$, where $\boldsymbol{w_i}^T = (d_{i1}, \ldots, d_{i|\mathcal{D}|})$ and for each j, if $(j, i) \in \tilde{R_D}$, then $d_{ij} = 1$ and otherwise $d_{ij} = 0$.
3. The vectors $\boldsymbol{w_i}$ $(1 \leq i \leq |\tilde{V}|)$ are clustered into several groups of terms. Each cluster is regarded as a topic and is represented by its central vector.
4. For the (projected) person context $\tilde{\mathcal{P}}$, we consider its corresponding matrix $M_{\tilde{\mathcal{P}}} = (\boldsymbol{p}_1^T, \ldots, \boldsymbol{p}_{|P|}^T)^T$, where $\boldsymbol{p_i}^T = (d_{i1}, \ldots, d_{i|\mathcal{D}|})$ and for each j, if $(i, j) \in \tilde{R_P}$, then $d_{ij} = 1$ and otherwise $d_{ij} = 0$.
5. In order to transform each person vector $\boldsymbol{p_i}^T$ into those in the document space, we apply *Singular Value Decomposition* (SVD) to $M_{\tilde{\mathcal{D}}}$.
6. Degree of relatedness of a person p to a topic t is given by the distance measure used in fuzzy K-means in [3],

$$R(p, t) = \frac{1}{\sum_{t_i \in \mathcal{T}} \left(\frac{dist(p,t)}{dist(p,t_i)} \right)^{\frac{2}{m-1}}}$$

   where $\mathcal{T}$ is the set of topics, m is a control parameter for fuzzy level and $dist(p_i, t_j)$ is Euclidean Distance in the document space between a person p_i and a topic t_j.
7. Regarding the vector of $R(p, t_i)$-value for each topic t_i as a probability distribution, the mediator level of the person p is given as the entropy of the distribution.

For each cluster i and j, we believe that $< c_i, c_j > \to 0$ by using the center of similar words. In other words, each topic has no relation with others. If one user has a more general interest, he/she will get involved into a more number of separated topics. Entropy will help us to judge the variance of a user's interest.

By sorting the persons in P in descending order of their mediator levels, we can define a ranked list of persons in P. For each person $p \in P$, the mediator rank and the mediator level of p are referred to as $rank(p)$ and $level(p)$, respectively.

4.2 Clustering Terms

In our clustering step, we use *Laplacian Eigenmaps* described in [6] to get closely related terms into one cluster. Moreover, for each topic, the document distribution is assumed to be Gaussian and the Gaussian kernel is used to make this points linear. Our clustering algorithm is performed as follows:

1. Calculate the matrix A as

$$A_{ij} = \begin{cases} exp(-disi,j/\sigma^2), & \text{if } r \neq j \\ 0, & \text{otherwise} \end{cases}$$

2. Calculate the matrix D as $D_{ij} = \sum_j (A_{ij})$, where D is a diagonal matrix.
3. L is defined as $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.
4. Find x_1,x_2,...,x_k, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalue), and form the matrix $X = [x_1 x_2 ... x_k] \in \Re^{n \times k}$ by stacking the eigenvectors in columns.
5. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
6. Treating each row of Y as a point in $\Re^K$, cluster them into k clusters via an extended k-means.
7. Finally, assign the original point s_j to cluster j if and only if row i of the matrix Y was assigned to cluster j.

In this procedure, σ^2 is a parameter for how rapidly the affinity A_{ij} falls off with the distance between s_i and s_j. In fact, however, a fixed parameter may lead to problems when topics' documents number and variance various, because if we have topics A and B, where topic A has a larger variance and more documents than B, then we get a document d which talks about A and B at the same time and assume that topics A and B in d have the same importance. If creating a new cluster is not allowed, we may prefer clustering d into A because it has a larger variance and it seems containing more property. So we introduce a auto-adaptive parameter selection.

To make the parameter σ adaptive, we use the following method to select a σ parameter for point i and j, which is similar to the work in [8].

The affinity between points s_i and s_j is defined as

$$A_{ij} = exp\left(-\frac{|s_i - s_j|^2}{\sigma_i \sigma_j}\right),$$

where the σ_i and σ_j are in a local scale $\sigma_i = \sqrt{\frac{1}{n}\sum_n |s_i - s_j|^2}$. This method allows σ to be automatically adjusted according to local variance, that is, no parameter setting is required. Our preliminary experimental results show effectiveness of this parameter selection method.

4.3 Extended *k*-Means Algorithm

For twitter data, it is usually hard to set a reasonable cluster number, and at most time, adjusting this parameter is a hard work for us. In actual experiment, we usually tend to select a lesser cluster number to avoid the classes which only contain one item. So we would like to adapt a simple clustering method which is able to adjust the number of clusters automatically.

As mentioned above, we use an extended k-means which allows to make new clusters in our clustering process by adjusting inner variances of clusters under a threshold so that we can obtain a good result with smaller distortion. The algorithm is summarized as follows:

1. Given k as an initial cluster number, k centers are randomly selected.
2. We calculate the squared errors with
$$J = \sum_{C_k \in \mathcal{CLS}} \sum_{\forall x_i \in C_k} (|c_k - x_i|^2),$$
where $\mathcal{CLS}$ is the family of clusters, c_k is the center of cluster C_k.
3. For each point, a cluster with the smallest distance to the point is identified.
4. The variance of the cluster with this point added is calculated. If the variance is larger than a threshold, a (singleton) cluster with the point is newly created. If otherwise, the point is merged into to the cluster.
5. When all of the points are processed, all empty clusters are deleted.
6. For each cluster C_i, its center vector (point) c_i is re-calculated by $c_i = \frac{1}{n} \sum_{x \in C_i} x$, where n is the number of points in C_i.
7. If the change of squared error is smaller than a given threshold, the clusters are output and the process is terminated. If otherwise, return to 2.

In order to verify the ability of extended k-means, we have tested it on a dataset with several centers and noise. Figure 2 shows our preliminary experimental results. It is clear that a traditional k-means cannot handle this situation as is shown in Fig. 2(a). On the other hand, our algorithm results in a good separation (Fig. 2(b)). In the method of k-means++ [7], although a set of good initial k seeds can be identified to get a better performance, we cannot change the initial k value which is inadequately given. However, in our method, it is no matter what the preset k is. Especially, if we provide a k smaller than the actual number of clusters, we can expect a relatively good result, which can allow us to select the cluster number more easily.

5 Method of Associative Search by Mediators

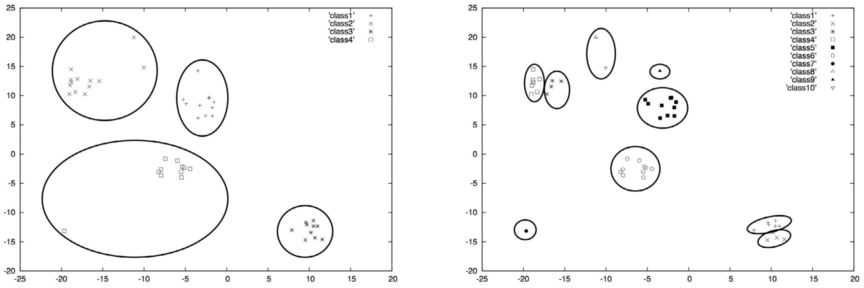
In this section, we formalize our method of associative search.

We are given a pair of relations, a person-term relation and a document-term relation, each of which is represented as a formal context, $\mathcal{P} = (P, V, R_P)$ and $\mathcal{D} = (D, V, R_D)$, respectively. As has been discussed in the previous section, it is assumed that the set of persons P is sorted (ranked) in descending order of their mediator levels.

Our associative search is performed by the following five steps.

1) Identifying Query Concept:

Given a set of keywords $K \subseteq V$ as a query, a query concept $Q_{\mathcal{P}} = (P_Q = K'^{(\mathcal{P})}, T_Q = K''^{(\mathcal{P})})$ in $\mathcal{P}$ is computed.



(a) Traditional k-Means

(b) Extended k-Means

Fig. 2. Clustering Results by Traditional k-Means and Extended k-Means

2) Getting User Interest:
For each person $p \in P_Q$ with Top-N higher mediator level, the user is asked to assign a mark "+", "−" or "∗" according to his/her interest.

3) Extracting Maximal Concepts Consistent with User Interest:
We try to extract maximal sub-concepts of $Q_{\mathcal{P}}$, $C_B = (P_B, T_B)$, which is consistent with the user interest.

4) Identifying User Aspects as Concepts:
Based on C_B in $\mathcal{P}$, a user aspect is identified as a concept $C_A = (A'^{(\mathcal{D})}, A)$ in $\mathcal{D}$, where $A = T_B \setminus T_Q$.

5) Extracting δ-Similar Concepts w.r.t. User Aspect:
Regarding C_A as a bridge concept, we try to extract concepts in $\mathcal{D}$, C_{target}, each of which is δ-similar to the query concept $Q_{\mathcal{D}} = (K'^{(\mathcal{D})}, K''^{(\mathcal{D})})$ w.r.t. A.

5.1 Identifying Query Concepts

Let us assume that a query Q a user interested in is given as a set of keywords (terms) $K \subseteq V$. We can consider a *query concept* in each context. They are defined as $C_Q^{\mathcal{P}} = (K'^{(\mathcal{P})}, K''^{(\mathcal{P})})$ in $\mathcal{P}$ and $C_Q^{\mathcal{D}} = (K'^{(\mathcal{D})}, K''^{(\mathcal{D})})$ in $\mathcal{D}$, respectively. $K'^{(\mathcal{P})}$ is a set of persons related to the query and $K'^{(\mathcal{D})}$ a set of documents matching the query.

5.2 Getting User Interest

We try to associatively search concepts of documents which are similar to $C_Q^{\mathcal{D}}$ with respect to some user's interest in persons. In order to get such a user's interest, we interactively ask the user to express his/her preference on the persons related to the query, that is, the persons in $K'^{(\mathcal{P})}$. Since the number of persons in $K'^{(\mathcal{P})}$ is large, the burden of such a user interaction would be a bit heavy for the user. Therefore, the user is asked his/her preference only on the Top-N ranked persons with higher mediator levels.

More precisely speaking, let $M \subseteq P$ be the list of Top-N ranked mediators in P, that is, $M = \{p \in P \mid rank(p) \text{ is in the top-}N\}$. For the query concept in $\mathcal{P}$, $Q_{\mathcal{P}} = (P_Q, T_Q)$, according to the user interest, for each person $p \in (P_Q \cap M)$, the user assigns a sign $+$, $-$ or $*$ to p for "*favorite*", "*dislike*", and "*don't care*", respectively, where the sign given to p is referred to as $sign(p)$.

Assuming $*$ as default sign to each $p \in P_Q \setminus M$, we can divide P_Q into three groups, POS, NEG and DC defined as $POS = \{p \in P_Q \mid sing(p) = +\}$, $NEG = \{p \in P_Q \mid sing(p) = -\}$ and $DC = \{p \in P_Q \mid sing(p) = *\}$.

5.3 Finding Concepts Consistent with User Interest

Based on the user interest, we try to find a concept in $\mathcal{P}$ which is *consistent* with the user interest. The consistency of concept is formally defined as follows.

Definition 1. (Consistent Concept)
Let $Q_{\mathcal{P}} = (P_Q, T_Q)$ be a query concept in $\mathcal{P}$, where P_Q is divided into POS, NEG and DC based on the user interest. Then, a concept $B = (P_B, T_B)$ in $\mathcal{P}$ is said to be *consistent* with the user interest if and only if $P_B \subseteq P_Q$, $P_B \supseteq POS$ and $P_B \cap NEG = \emptyset$. ∎

A consistent concept B is a sub-concept of the query concept whose extent must subsume POS and whose intent must completely exclude NEG. Since there are in general several concepts satisfying the constraints, we try to extract maximally general ones among them.

5.4 Identifying User Aspect as Concept

Roughly speaking, an associative search of concepts can be realized by finding a concept C which is similar to a query concept Q with respect to some aspect user interested in. If we can observe a concept which corresponds to the user aspect and bridges Q and C in some sense, it would be reasonable to accept similarity between Q and C under the aspect. In order to realize this kind of associative search, we here formalize a user aspect as a concept reflecting the user interest.

Let $B = (P_B, T_B)$ be a (maximal) concept in $\mathcal{P}$ which is consistent with the user interest. From the definition, since B is a sub-concept of $Q_{\mathcal{P}} = (P_Q, T_Q)$, $T_Q \subseteq T_B$ holds. Therefore, $A = T_B \setminus T_Q$ can be viewed as the set of attributes (terms) which can implicitly characterize the user interest. In other words, A can be considered to represent a preferable aspect of the user.

In order to find similarity of concepts in $\mathcal{D}$ under the aspect reflecting the user interest in persons, we consider a concept in $\mathcal{D}$ which is defined based on A. Formally speaking, if A is a closure in $\mathcal{D}$, that is, $(A'^{(\mathcal{D})}, A''^{(\mathcal{D})} = A)$ is a concept in $\mathcal{D}$, then we regard the concept $C_A = (A'^{(\mathcal{D})}, A)$ as an aspect reflecting the user interest in persons.

5.5 Extracting Conditionally Similarity Concepts w.r.t. User Aspect

Intuitively speaking, our associative search is realized by finding a concept C which is similar to a query concept Q with respect to some user aspect. Before presenting such a conditional similarity between concepts, we define a similarity between concepts without any conditioning. Our similarity can be defined based on the notion of *bond* [4] which is regarded as an extension of *Jaccard Coefficient* and can easily be calculated in our concept search.

Definition 2. (Concept Similarity Based on Bond Measure)
Let $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be a pair of concepts. Then a similarity between C_1 and C_2, denoted by $sim(C_1, C_2)$, is defined as

$$sim(C_1, C_2) = bond(Y_1 \cup Y_2) = |X_1 \cap X_2| / |X_1 \cup X_2|. \qquad \blacksquare$$

The measure is extended for *conditional similarity* between concepts.

Definition 3. (Concept Similarity Based on Conditional Bond)
Let $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be a pair of concepts. For a set of attributes R as some aspect, a similarity between C_1 and C_2 with respect to the aspect R, denoted by $sim(C_1, C_2|R)$, is defined as

$$sim(C_1, C_2|R) = bond(Y_1 \cup Y_2|R) = |\bigcap_{y \in Y_1 \cup Y_2} (R' \cap y')| / |\bigcup_{y \in Y_1 \cup Y_2} (R' \cap y')|. \qquad \blacksquare$$

Let δ be a given threshold for the minimum bond value. For a pair of concepts C_1 and C_2, if $sim(C_1, C_2) \geq \delta$ holds, then C_1 and C_2 are said to be *δ-similar*. Moreover, $sim(C_1, C_2|R) \geq \delta$ for an aspect R, C_1 and C_2 are said to be *conditionally δ-similar* with respect to R.

For a query concept Q and the user aspect C_A, if C_A bridges Q and a concept C in some sense, then it would be natural to find similarity between Q and C under the aspect. Here, C_A is an essential concept which corresponds to the user aspect and can work as the basis of the similarity between Q and C. We call such a C_A a *bridge concept* for Q and C.

Definition 4. (δ-Bridge Concept)
Let $C_R = (X_R, Y_R)$ and $C_L = (X_L, Y_L)$ be a pair of concepts. For a given similarity threshold δ, if a concept $C_W = (X_W, Y_W)$ satisfies the following conditions, then C_W is called a *δ-bridge concept* between C_R and C_L w.r.t. the *aspect* Y_W.

Structural Constraint: $X_R \cap X_W \neq \emptyset$ and $X_L \cap X_W \neq \emptyset$.
Conditional Similarity Constraint: C_R and C_L are conditionally δ-similar with respect to Y_W, that is, $sim(C_R, C_L|Y_W) \geq \delta$. $\blacksquare$

The structural constraint requires that the concept C_W properly bridges (overlaps) with both C_R and C_L at their extents. Moreover, by the conditional similarity constraint, C_R and C_L must be δ-similar under the conditioning with the intent of C_W, Y_W. It should be emphasized here that C_R and C_L are not always δ-similar in the original context (without conditioning).

Regarding the user aspect $C_A = (A'^{(\mathcal{D})}, A)$ as a bridge concept, for the query concept $Q_{\mathcal{D}} = (K'^{(\mathcal{D})}, K''^{(\mathcal{D})})$ in $\mathcal{D}$, we try to find a target concept C_{target} which is conditionally δ-similar to $Q_{\mathcal{D}}$ with respect to A.

6 Extracting Target Concepts for Associative Search

As has been discussed in the previous section, our computation procedure of associative search consists of five steps. In the procedure, the primary tasks are "finding maximal consistent concepts" and "finding conditionally δ-similar concepts". For each of the tasks, we can design a simple depth-first algorithm.

6.1 Extracting Concepts Consistent with User Interest

Our task here is to find every maximal concept C_B in $\mathcal{P}$ which is a sub-concept of the query concept $Q_{\mathcal{P}} = (P_Q, T_Q)$ and whose extent must subsume POS and exclude NEG.

Assume P_Q can be divided into POS, NEG and DC. In order to find C_B satisfying the constraints, we can basically expand POS by adding a person in DC step by step in a depth-first manner. More precisely, for the closure of person set X_i such that $(POS \cup DC) \supseteq X_i \supseteq POS$, we expand X_i by adding a person $xin(DC \setminus X_i)$ and compute the closure $X_{i+1} = (X \cup \{x\})''^{(\mathcal{P})}$. Then we check whether $X_{i+1} \subseteq (POS \cup DC)$ or not. If yes, X_{i+1} is tried to further expand by adding a person in $DC \setminus X_{i+1}$. If otherwise, X_{i+1} is discarded and X_i is expanded with another person by backtrack because X_{i+1} includes some person in NEG or $(X'^{(\mathcal{P})}_{i+1}, X_{i+1})$ is not a sub-concept of $Q_{\mathcal{P}}$. If a closure X_i cannot be expanded, then X_i is stored into a list of maximal concepts as a tentative candidate. Such an expansion process recursively iterated until no closure remains to be expanded. Details of the algorithm can be found in [2].

6.2 Extracting Conditionally Similar Concepts w.r.t. User Aspect

The task here is to find concepts each of which is conditionally δ-similar to the query concept and can be connected with the query concept by a concept corresponding to the user aspect.

Let $Q_{\mathcal{D}} = (K'^{(\mathcal{D})}, K''^{(\mathcal{D})})$ be a query concept and $C_A = (A'^{(\mathcal{D})}, A)$ a user aspect (as a concept). A target we try to find is a concept $C_{Target} = (T''^{(\mathcal{D})}, T)$ such that (1) $T \cap A = \emptyset$, (2) $K''^{(\mathcal{D})} \cap T = \emptyset$, (3) $K'^{(\mathcal{D})} \cap T'^{(\mathcal{D})} \neq \emptyset$ and (4) $bond(K''^{(\mathcal{D})} \cup T|A) \geq \delta$. Particularly, we try to extract maximal ones satisfying the constraints. It should be noted here that the constraints (1) and (2) are not included in our original definition of δ-bridge concepts. They are assumed to obtain more interesting associations of terms. Moreover, from a computational view point, they can restrict our search space for efficient computation.

In order to find such a concept, we try to recursively expand a closure of terms in depth-first manner. Let $X_i \subseteq V$ be the closure of a set of terms such that $X_i \cap (K''^{(\mathcal{D})} \cup A) = \emptyset$. For a term $x \in V \setminus (K''^{(\mathcal{D})} \cup A \cup X_i)$, we check whether $X_{i+1} = (X_i \cup \{x\})''^{(\mathcal{D})}$ as T satisfies all of the four constraints. If X_{i+1} does not satisfy (1) or (2), X_{i+1} can be discarded because any expansion of X_{i+1} can never satisfy the constraints. If X_{i+1} does not satisfy (3), that is, $K'^{(\mathcal{D})} \cap X'^{(\mathcal{D})}_{i+1} = \emptyset$, then any expansion of X_{i+1} also violates the constraint. Therefore we can stop expanding X_{i+1}. If the constraint (4) cannot be satisfied for X_{i+1}, any expansion of X_{i+1} can be pruned. This is because the bond measure is monotonically decreasing as a set of attributes (terms) becomes larger. Therefore, (4) cannot be satisfied for any expansion of X_{i+1}. If X_{i+1} is discarded, X_i is tried to expand with another term in $V \setminus (K''^{(\mathcal{D})} \cup A \cup X_i)$ by backtrack.

On the other hand, all of the constraints are satisfied for X_{i+1}, then $(X'^{(\mathcal{D})}_{i+1}, X_{i+1})$ becomes a candidate of our target. Then X_{i+1} is further tried to expand with a term in $V \setminus (K''^{(\mathcal{D})} \cup A \cup X_{i+1})$. Such an expansion process is recursively iterated until no closure remains to be examined.

7 Experimental Results

In this section, we present our experimental results. Our system coded in C has been executed on a PC with Intel® Core$^{\text{TM}}$-i3 M380 (2.53GHz) CPU .

7.1 Person-Topic Relation and Tweet-Topic Relation

We have created our datasets from a set of tweets gathered with *Twitter* search API[1]. We have collected Japanese tweets including a keyword "*Soccer*"[2] in November, 2012. The number of collected tweets is about 600,000.

As a preprocess, we have first applied *Morphological Analysis*, and then extracted nouns as feature terms. Then, too frequent and too infrequent nouns have been removed. Since each tweet consists of at most 140 characters, a number of tweets have become empty after the preprocess. We have exclude those empty ones and obtained 538,355 tweets. The number of terms in the remaining tweets is 51,927 and the number of persons by whom those are tweeted 238.

In our mediator ranking process, the tweets are first concatenated into a single document for each people. Then, all terms with lower *TF-IDF* values have been further removed. With our extended k-means, the remaining 912 terms have been clustered into 155 topics. The persons have been assigned mediator levels based on the distances to the topics and ranked according to the levels.

It is noted here that after the ranking, the original tweets have been compressed into ones represented in corresponding topics. This is because each of the original tweets consists of a small number of nouns and it is hence difficult

[1] `https://dev.twitter.com`

[2] In practice, each keyword is in Japanese.

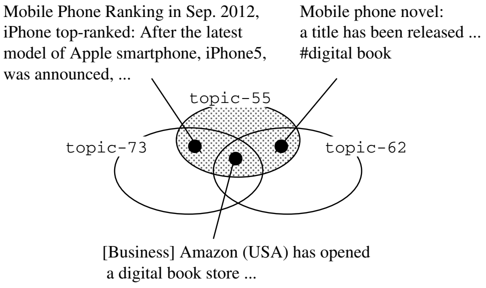

Mobile Phone Ranking in Sep. 2012, iPhone top-ranked: After the latest model of Apple smartphone, iPhone5, was announced, ...
Mobile phone novel: a title has been released ... #digital book
topic-55
topic-73
topic-62
[Business] Amazon (USA) has opened a digital book store ...

Fig. 3. Example of Associative Connection of Concepts for "Smartphone"

to extract concepts without the compression. For example, if a tweet consists of a set of (original) terms a, b, c belonging to topics A, B and C, respectively, then it is compressed into one with A, B and C. By the compression, our chance to obtain concepts can be enhanced because several distinct terms are identified as the same topic. Thus, sharing 155 topics, we have prepared a person-topic relation with 238 persons and a tweet-topic relation with $538,355$ tweets.

7.2 Example of Target Concept by Associative Search

We show here an example we have actually extracted from the above relations.

Given a keyword "smartphone" as a query, we first convert it into its corresponding topic, topic-73. The extent of the query concept in the person-topic relation consists of 200 persons. For a user interest setting the highest ranked person as NEG, we have obtained 21 maximal consistent concepts. Then, those concepts provides 21 candidate aspects. Some of them correspond to concepts in the tweet-topic relation and can work as δ-bridge concepts each of which connects the query concept and a target concept. An example of associative connection of concepts is presented in Figure 3.

In the figure, topic-55 of the bridge concept includes "mobile" and topic-62 of the target concept "digital book". This means that we can associatively access to "digital book" from "smartphone" via "mobile". More concretely speaking, the following tweet which is an instance of the bridge concept can connect "smartphone" and "digital book":

> *Amazon (USA) has opened a digital book store "Kindle Store" in Japan. Although Kindle Paperwhite will be shipped in November, users can access to the store with their smartphones (multi-function mobile phones).*

Needless to say, such a flexible access cannot be obtained by standard search engines. Thus, our associative search provides us a chance to get various useful information which might be interesting for us.

The total computation time has been 5.0 seconds, including 0.24 sec. for extracting 21 maximal consistent concepts. Thus, our associative search can be performed efficiently.

8 Concluding Remarks

In this paper, we have discussed a framework of associative search. A remarkable point of the framework is that our search is dependent on user interests in good mediators. We have presented a computational procedure for our associative search and designed depth-first algorithms for extracting our target concepts. Some experimental results have showed effectiveness of our search method.

As has been mentioned, it would be difficult to extract useful concepts from too sparse datasets like twitter data without compressing or abstracting original terms. The result of our associative search is strongly affected by how the terms are compressed (clustered) into topics. Therefore, it would be worth further investigating a method for detecting adequate topics. Some additional knowledge such as a thesaurus would be helpful in improving quality of topic detection. Our framework, however, does not necessarily require such a term compression if we are concerned with datasets with denser contents like News article, Blog articles, Web documents and so forth. Some experimentations are also currently in progress for such denser datasets.

References

1. Takano, A., Niwa, Y., Nishioka, S., Iwayama, M., Hisamitsu, T., Imaichi, O., Sakurai, H.: Information Access Based on Associative Calculation. In: Jeffery, K., Hlaváč, V., Wiedermann, J. (eds.) SOFSEM 2000. LNCS, vol. 1963, pp. 187–201. Springer, Heidelberg (2000)
2. Li, A., Haraguchi, M., Okubo, Y.: Implicit Groups of Web Pages as Constrained Top N Concepts. In: Proc. of the 2008 IEEE/WIC/ACM Int'l Conf. on WI-IAT Workshops, pp. 190–194 (2008)
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers (1981)
4. Omiecinski, E.R.: Alternative Interest Measures for Mining Associations in Databases. IEEE Trans. on KDE 15(1), 57–69 (2003)
5. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer (1999)
6. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. In: Proc. of NIPS 2001, pp. 849–856 (2001)
7. Arthur, D., Vassilvitskii, S.: k-means++: The Advantages of Careful Seeding. In: Proc. of SODA 2007, pp. 1027–1035 (2007)
8. Zelnik-Manor, L., Perona, P.: Self-Tuning Spectral Clustering. In: Proc. of NIPS 2004, pp. 1601–1608 (2004)
9. Gan, G., Ma, C., Wu, J.: Data Clustering - Theory, Algorithms, and Applications. SIAM (2007)
10. Carpineto, C., Osiński, S., Romano, G., Weiss, D.: Survey of Web Clustering Engines. ACM Computing Surveys 41(3), Article 17 (2009)
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
12. Alqadah, F., Bhatnagar, R.: Similarity Measures in Formal Concept Analysis. Annals of Mathematics and Artificial Intelligence 61(3), 245–256 (2011)
13. Dau, F., Ducrou, J., Eklund, P.: Concept Similarity and Related Categories in SearchSleuth. In: Proc. of ICCS 2008, pp. 255–268 (2008)

Estimating and Forecasting Network Traffic Performance Based on Statistical Patterns Observed in SNMP Data

Kejia Hu[1,2], Alex Sim[1], Demetris Antoniades[3], and Constantine Dovrolis[3]

[1] Lawrence Berkeley National Laboratory, USA
[2] University of California at Davis, USA
{kjhu,asim}@lbl.gov
[3] Georgia Institute of Technology, USA
{danton,dovrolis}@gatech.edu

Abstract. With scientific data growing to unprecedented volumes and the needs to share such massive amounts of data by increasing numbers of geographically distributed collaborators, the best possible network performance is required for efficient data access. Estimating the network traffic performance for a given time window with a probabilistic tolerance enables better data routing and transfers that is particularly important for large scientific data movements, which can be found in almost every scientific domain. In this paper, we develop a network performance estimation model based on statistical time series approach, to improve the efficiency of network resource utilization and data transfer scheduling and management over networks. Seasonal adjustment procedures are developed for identification of the cycling period and patterns, seasonal adjustment and diagnostics. Compared to the traditional time series models, we show a better forecast performance in our seasonal adjustment model with narrow confidence intervals.

Keywords: Time series, Seasonal Adjustment, Network Traffic Forecast, STL, X12-ARIMA.

1 Introduction

The analysis of network traffic is getting more and more important today to efficiently utilize the limited resources offered by network infrastructure and wisely plan the large data transfers. Estimating the network traffic for a given time window with a given probabilistic tolerance error enables better data routing and transfers, which is particularly important for large scientific data movements. Short-term prediction of network traffic guides the several immediate scientific data placement. Long-term forecast of network traffic evaluates the performance of network and enables the capacity planning of the network infrastructure up to the future needs.

The problem of analysis of network traffic has received attention over the years. Previous researches about network traffic can be distinguished in two categories: frequency-domain methods including spectral analysis and wavelet analysis

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 601–615, 2013.

[23] and time-domain methods including auto-correlation and cross-correlation analysis [5]. Besides the main stream models such as time-domain models ARIMA, FARIMA or frequency-domain methods wavelet analysis, there are also learning approach methods [1,18].

However, one very important feature in the time series is the periodicity or seasonality [10]. The seasonal variation is a component of a time series and occurs as a repetitive and predictable movement around the trend line in one time cycle. Organizations such as manufacturing industry with quarterly assignment adjust their performance relative to normal seasonal variation. In the census data released by U.S. Census Bureau, many indexes are seasonally adjusted to monitor the general trend without the influence of periodical changes. For example, the unemployment rate is expected to increase in every June because of the recent graduates entering the job market. However, the overall unemployment rate should be evaluated after removing the expected seasonality for this June effect. Seasonal movements are often large enough to mask other characteristics of the data such as current trends. For example, if each month has a different seasonal tendency toward high or low values, it could be difficult to detect the general direction of a recent monthly movement in the time series (increase, decrease, turning point, no change, consistency with another economic indicator, etc.). With seasonal adjustment removing seasonal component from the original series, seasonally adjusted series would reveal the recent trend without obscuration from the seasonality, and its relationship with other different series can be easily measured. However, the periodicity in the network traffic has never been explored in our knowledge, and the modeling based on periodicity has never been applied to the analysis and forecast of network traffic. From our analysis, the network traffic measurement data, SNMP shows significant periodical behavior. In Figure 1, the periodicity of network traffic within a day is shown, based on the seasonality of the 1-year SNMP data, and the trend component shows the general change in the original series while the irregular component shows the collection of the random behavior in network usage.

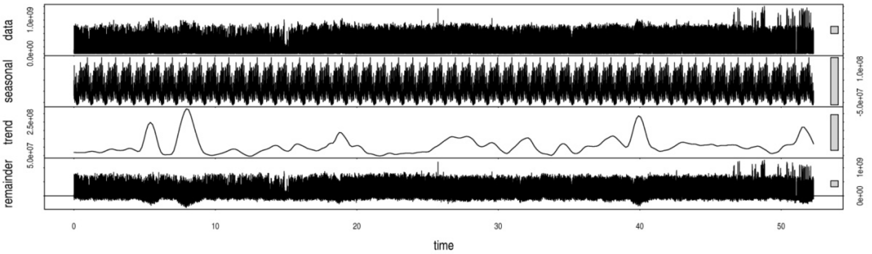


Fig. 1. Periodicity in network traffic data. Original series, seasonal component, trend component and irregular/remainder component from the top to the bottom.

Figure 1 is the result of seasonal adjustment, which decomposes the original series into three components: S (Seasonal Component), I (Irregular Component), and T (Trend Component). The seasonal component is the second plot in Figure 1 which models the undergoing specific variations at certain moments during one cycling period. It usually combines features of regular behavior of network usage and routine data transfer. The third plot is trend component which shows the long-term change from general phenomena, and it fits our interest in estimating the current situation and predicting future condition. We could find the illustrated series has a general trend that involves high volume network traffic at the beginning and the end of the observation range. The last plot is irregular component which models the unexpected behavior from the statistical errors or from the nonrecurring accidental or fortuitous events. It is usually assumed to follow a normal distribution and outliers can be detected by p-values.

In this paper, we will address two challenges in the network performance modeling with STL and seasonal adjustment: 1. Seasonality in the network measurement data has not been addressed before in our knowledge, and we will show our findings from the evaluation results from three criterion indexes and diagnostic results supporting the existence of seasonality in the network measurement data. 2. The periodicity in the network measurement data is unknown. STL and seasonal adjustment methods cannot be used without the periodicity of the time series. We studied three criterions to select the best periodicity to generate the prediction with the least forecast error and the full extraction of the seasonal/periodical pattern in the measurement data.

In this paper, we focus on finding and modeling the periodical patterns in the network traffic, and discuss the procedure of seasonal adjustment methods on network traffic measurement data. Unlike general social data, the cycling period is unknown in the network traffic data. In section 2, we discuss the identification of the significant cycling period of the network traffic measurement data, two seasonal adjustment methods, X12-ARIMA and STL, and the validation methods with seasonal adjustment. In section 3, we discuss the results of analysis, and evaluate the performance of the prediction model. In section 4, we conclude our results with comparison of our models to the results from two methods, ARIMA and wavelet-based methods.

2 Periodicity and Seasonal Adjustment

2.1 Criterion to Identify Periodicity

Most social data shows seasonal periodical patterns. In agricultural industry, we can find a seasonality of sowing and harvesting in a yearly cycle. In banking industry, we can find a seasonality of savings amount increasing at the beginning and decreasing at the end in a monthly cycle. In highway transportation traffic, you can find a seasonality of rush hours in a daily period. However, periodicity of network traffic is unknown, and network measurement data is collected more frequently, compared to other social data, in about every 30 seconds for SNMP measurements. In order to apply seasonal adjustment, the cycling period needs to be determined.

We evaluate the data based on different cycling period according to three criterions:

1. Identified seasonality: A cycling period is determined within the period that can be identified with significant seasonality. This feature in our model is used to provide better prediction.
2. Residual seasonality: The seasonally adjusted data series is the residual data that seasonal components are subtracted from the original data series. Residual seasonality is not expected in the seasonally adjusted data series.
3. Log transformation: Log transformation was applied to provide stationary data over time, for old data, recent historical data as well as newly acquired data.

Seasonality detection methods were used for the first two criterions. Log transformation was determined by how the data is stationary.

Seasonality Detection

The seasonality detection methods are divided into two groups: graphical techniques and statistics based on Seasonal Index measures. The graphical techniques include run sequence plot, seasonal subseries plot, multiple box plots and autocorrelation plot. The significance of seasonality is determined based on the plot by the human eyes, and it is often subjective. For an automated procedure of the seasonal adjustment, the statistical method based on seasonal index measures is selected in our model. Our analysis used two diagnostic methods, the F-test for the presence of seasonality [12] and M7 for X11-ARIMA [17].

Log-Transformation

The log transformation based on the log-likelihood test is to stabilize the variance, so that the data can be modeled with the Box-Jenkins methodology [2]. The model is derived with the log-transformed data and the original data to obtain the maximum likelihood. Log transformation is applied when the model has larger likelihood based on the transformed data.

After evaluating periodicity for the network traffic measurement data, the network traffic data is organized into a time frame with the determined cycle.

2.2 Seasonal Adjustment

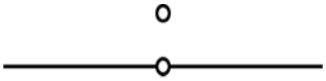
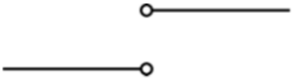
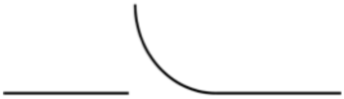
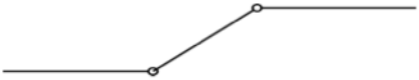
Missing Value Treatment

Before the seasonal adjustment methods are applied, any missing values and identifying outliers in network traffic measurement data are treated. Some of the network traffic data are lost due to failures on the collection device. The rate of the missing values in our data is around 0.7%, but missing values may cause an increase in the forecast error, especially when recent records are missing. The recent activities of nearby data points and the feature of its cycling period are considered for the replacement methods for missing values, and they are estimated by a weighted average of the recent data points and the points falling into the same cyclic spots within every period.

$$\widehat{x_{tk+p}} = \sum_{i=tk}^{tk+2p} \alpha_i x_i + \sum_{j=1}^{C} \beta_j x_{tj+p}$$

where $\sum_{i=tk}^{tk+2p} \alpha_i + \sum_{j=1}^{C} \beta_j = 1$, t is the length of every cycling period, k represents the kth cycle where the missing value is found, p represents the pth spot where the missing value is found, and C is the total number of cycles in the period. $\sum_{i=tk}^{tk+2p} \alpha_i x_i$ represents the influence of the recent activity to the missing point, and $\sum_{j=1}^{C} \beta_j x_{tj+p}$ represents the periodicity occurred in the data series. The weighted average coefficients α_i and β_j would be selected as the closer activities are more related to the missing point. The missing value x_{tk+p} occurred in *p*th spot in kth cycle would be replaced by its estimated weighted average $\widehat{x_{tk+p}}$.

Table 1. Illustration of Four Types of Regressors

AO	LS
	
TC	RP
	

Outlier Detection

The outliers are recorded, rather than being replaced, by its value, time and type. The four types of records are AO (Additive outliers), LS (Level Shift), TC (Temporary Change) and RP (Ramps) as shown in Table 1. The outlier data series is used in the RegARIMA for X12-ARIMA method.

$$\log(Y_t) = \beta' X_t + Z_t$$

where the log transformation is based on the stable variation, Z_t is the ARIMA process, Y_t is the original series, and X_t is the regressor of outlier data series.

$$\text{AO: if } t_0 \text{ is an addtive outlier, then } X_{AO_t} = \begin{cases} 1 & t = t_0 \\ 0 & t \neq t_0 \end{cases}$$

$$\text{LS: if series shift suddenly at } t_0 \text{ and continues on new level, } X_{LS_t} = \begin{cases} -1 & t < t_0 \\ 0 & t \geq t_0 \end{cases}$$

TC: if series shift suddenly at t_0, then slowly declines to original level.

$$X_{TC_t} = \begin{cases} 0 & t < t_0 \\ \alpha^{t-t_0} & t \geq t_0 \end{cases} \text{ where } \alpha \text{ is a decreasing rate estimated from data}$$

RP: if series slowly change to a new level start from t_0 and end by t_1,

$$X_{RP_t} = \begin{cases} -1 & t < t_0 \\ \dfrac{t - t_0}{t_1 - t_0} - 1 & t_0 \leq t < t_1 \\ 0 & t \geq t_1 \end{cases}$$

For a seasonal-trend decomposition procedure based on Loess (STL) [3], the extreme values are modified to be robust in the iteration of STL algorithm. The distortion of extreme values would be limited in the modeling and forecasting, but its influence would remain in the data series for hidden information.

Seasonal Adjustment Methods

Two seasonal adjustment methods are applied to the network traffic measurement data, a seasonal-trend decomposition procedure based on Loess (STL) and the X12-ARIMA [21,22].

X12-ARIMA includes a group of calendar effects that the network traffic measurement data does not show, and the regressors that only fit the characteristics of network traffic must be selected. The network traffic measurement data fluctuates frequently, compared to the social data, due to the dynamic network data transfer features, and the outliers that are represented by large data transfers are likely to occur in random. The special characteristics of network traffic should be considered in the X12-ARIMA regression model for the frequent changing patterns and more aberrant events.

After the seasonal adjustment on the network traffic measurement data, the data will be decomposed into three components: trend, seasonal and irregular components. The performance of the seasonal adjustment model will be evaluated, and assessed with two diagnostic methods.

2.3 Seasonal Adjustment Diagnostics

The diagnostics of seasonal adjustment examine the stability of adjusted data series with a persistent model for new data feeds, which enables better prediction of network traffic performance. New measurement is collected for the network traffic data, and the model must not be changed frequently causing the computational costs and time. To assess the stability, two diagnostics results are considered from the revisions history diagnostics and the sliding spans diagnostics.

Revision History Diagnostics

The revision history diagnostics [19] create many seasonal adjustments on a sequence of increasing data spans, at a new time point each time. The assessment of stability is based on the evaluation of the magnitude of revisions over time, which parameterizes the model characteristics such as transformation type, performance parameters such as forecast errors, and model evaluation values such as AIC and log-likelihood. Take the adjusted series $\mathrm{A_t}$ as an example, we illustrate how we measure the revision over a period lag.

For a given series $\mathrm{y_t}$ where *t=1,...,T*, we define $\mathrm{A_{t|n}}$ to be the seasonal adjustment of $\mathrm{y_t}$ calculated from the series $\mathrm{y_1}$, $\mathrm{y_2}$,..., $\mathrm{y_n}$, where $\mathrm{t \le n \le T}$. The concurrent seasonal adjustment of observation t is $\mathrm{A_{t|t}}$ and final adjustment is $\mathrm{A_{t|T}}$. The concurrent target captures the lagged revision history where the target is assumed to be the concurrent estimate. Concurrent target $= \frac{\mathrm{A}_{t|t+lag}-\mathrm{A}_{t|t}}{\mathrm{A}_{t|t}}$. If the revision (absolute value of concurrent target) is very large, we consider the model is unstable when we add more data. The final target concurrent gives the lagged revision history where the target is assumed to be the final estimate. Final target $= \frac{\mathrm{A}_{t|T}-\mathrm{A}_{t|t+lag}}{\mathrm{A}_{t|t+lag}}$. If the revision (the absolute value of final target) is too large, we consider the model is unstable when we go back into different time points in history.

Sliding Spans Diagnostics

The sliding spans diagnostics [9] compare the adjusted results to the overlapping sub spans of the time series, for the stability of the seasonal adjustment. Each span with a length H starts one cycling period after the previous span, where H depends on the choice of seasonal adjustment filters. Seasonal adjustment is applied on each span, and their adjusted results are compared. When the adjusted results would be changed too much across spans, the seasonal adjustment must not have a stable model.

Let A_t denote its seasonally adjusted value obtained from the complete series, and let A_t^j denote the adjusted value obtained from the *j*-th span. Then, the seasonal adjustment A_t is called unacceptably unstable, if

$$\frac{\max_{j} \mathrm{A}_t^j - \min_{j} \mathrm{A}_t^j}{\min_{j} \mathrm{A}_t^j} > 0.03$$

3 Seasonal Adjustment on SNMP Data

The SNMP data provides aggregated link usage data, collected every 30 seconds, on the network connections. We have access to publicly available ESnet SNMP data [15], and the time span of these SNMP data is from May 6 18:49:00 PDT 2011 to the present time. In our current studies, we retrieved data up to Jul 25 15:56:30 PDT 2012.

3.1 Periodicity of SNMP Data

The potential periodicity for SNMP data can be from as small as one minute to a few months. Figure 2 shows the decomposed series based on periodicity of a week, a day, an hour and a minute. In each plot, four subplots show, in order, original series, seasonal component, trend component and irregular component from the top to the bottom. With a longer periodicity, clearer pattern of seasonality is observed, and the trend component shows smooth curves. The irregular component shows very large variations, and we suspect that many routine events cannot be counted as seasonal component with the large rough cycling period. With a shorter periodicity, the seasonal pattern is not clear, and the trend component changes frequently. However, the irregular component shows a small portion. Seasonal adjustment with a longer periodicity such as a week does not identify a clear seasonality, and a shorter periodicity

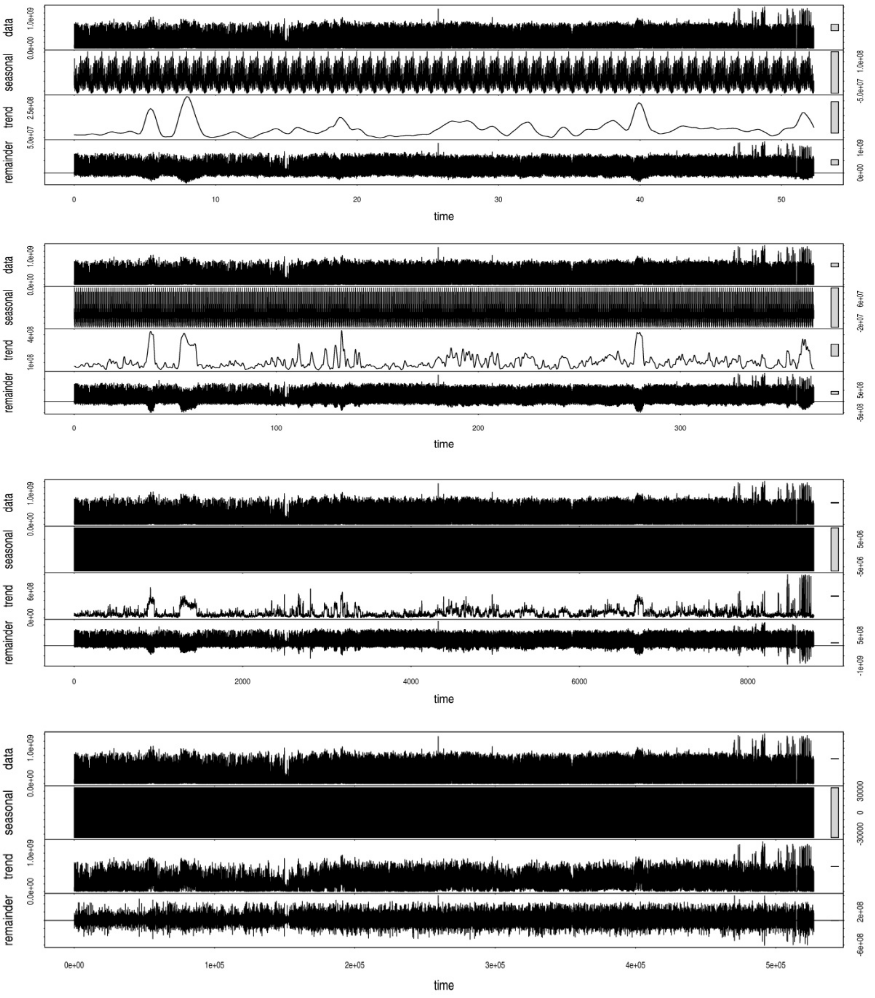


Fig. 2. Seasonal adjustment based on the periodicity of a week, a day, an hour and a minute, from the top to the bottom. Each plot shows four subplots of original series, seasonal component, trend component and irregular component.

such as a minute causes the seasonality distorted easily by random events. Table 2 shows the portion of seasonally adjusted components, compared to the original data series.

Table 2 show that the seasonal and irregular components decrease with a shorter periodicity, and the trend component increases with a shorter periodicity. The seasonal and trend components have the deterministic model for forecasting, and the irregular component has the distribution information on the forecast with a certain level of error, which can be modeled as a normal distribution with a zero mean and a variance. The non-deterministic behavior of irregular component causes the uncertainty in the forecast. For more deterministic information and less randomness, portions of seasonal and trend components should be higher, and the portion of irregular component should be lower. As a trade-off during the periodicity selection, a significant seasonality should be identified while controlling the irregular portion in the model. From Table 2, the best periodicity would be observed between a day and an hour.

Table 2. Portion of seasonally adjusted components

Periodicity	Seasonal	Trend	Irregular
Week	43.2%	44.6%	78.3%
Day	28.1%	69.1%	53.0%
Hour	5.7%	88.5%	29.8%
Minute	0.1%	98.5%	6.8%

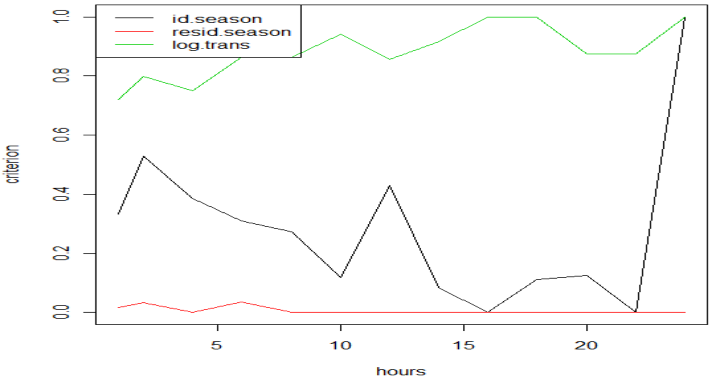


Fig. 3. Seasonal adjustment criterions for different periodicity

Figure 3 shows the seasonal adjustment criterion defined in section 2.1 to evaluate the performance of model with different periodicity. The significant seasonality is observed when the index of the identified seasonality is close to 1. The residual seasonality is close to 0 when the seasonality is fully extracted from the original series and modeled. The log transformation shows stability when the value is close to either 0 or 1. From Figure 3, the optimal period would be based on a day, since the identified seasonality is exactly equal to 1 and the residual seasonality is equal to 0. The model identifies and extracts a significant seasonality effectively. The stability of the seasonal adjustment based on the daily period is strong with the index of “log transformation” as 1. Figure 3 also shows that 12 hours of periodicity indicates low residual seasonality with the relatively higher identified seasonality at 0.4. The log transformation does not indicate the stability as high as the daily periodicity, but the residual seasonality is 0 indicating the model can fully extract the seasonality. This may corresponds to the network activity based on the daily working schedule. Another significant seasonality is shown in Figure 3 at 2 hours of the periodicity with the identified value at 0.5 and the remaining seasonality close to 0, although the log transformation is valued at 0.8 indicating the model is not stable. From these observations, the optimal cycling period would be determined as a day with significant identified seasonality, zero remaining seasonality in the residuals and stable transformation all the time. When we test the cycling period for different series, the daily period holds best cycling period for 90% series.

3.2 Seasonal Adjustment

Based on the periodicity of a day, the seasonal adjustment is applied on the organized time frame of SNMP data.

Figure 4 shows the results from seasonal adjustment based on STL and X12-ARIMA respectively with plots of original series, seasonal, trend and irregular components from the top to the bottom.

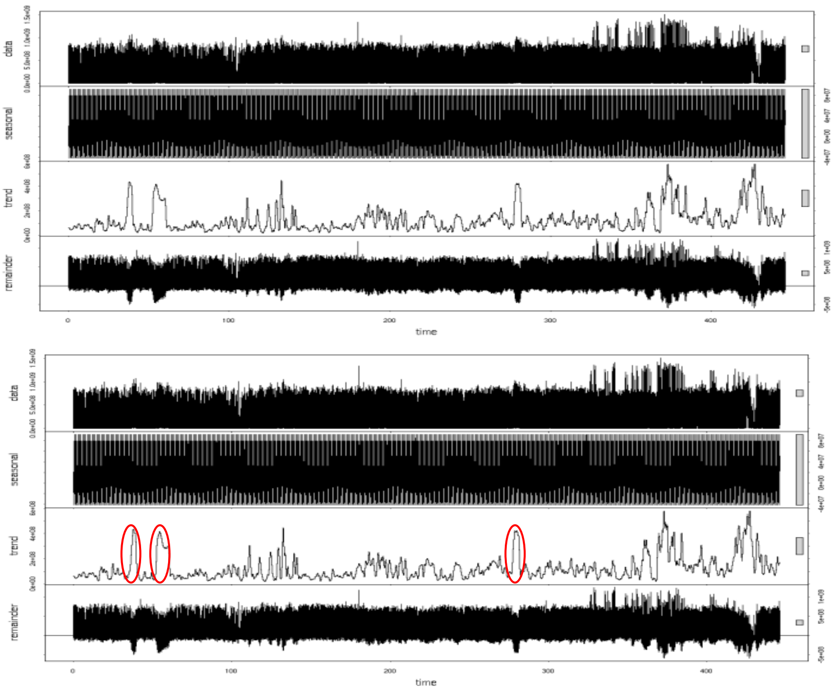


Fig. 4. Seasonally adjusted series based on STL and X12-ARIMA

The STL model derived with 15 outer loops. The span used for "s" (seasonal), "t" (trend) and "l" (loess) is respectively 12841361,4321 and 2881. The weights for observation fall into 1st quantile at 0.7703182 and 3rd quantile at 0.9887374 with median to be 0.9452160. Extreme value in the original series is detected with weight equal to 0. In the final decomposition, the IQR (interquartile range) of seasonal component accounts for 40.5% of the IQR of original series and IQR of trend component is 75.9% of the IQR of original series.

The X12-ARIMA model choice is ARIMA (3 0 0)(0 1 1) with log transformation with regression on identified outliers and an intercept as 0.02. F-test for seasonality has significant F-value 30.383 and suggests seasonality present at 0.1 percent level. The residual seasonality is tested to be no presence at the 1 percent level. The IQR of seasonal component decomposed in X12-ARIMA is 42.5% of the IQR of original series and IQR of trend component is 73.2% of the IQR of original series.

The results obtained from X12-ARIMA and STL show similar decomposed components. The trend component indicates that there are more level shifting occurred at the end of the time series. At the end of time series, more additive outliers are identified. A few temporary changes are observed in the series, and 3 significant ones are identified which is circled red in Figure 4 with a sudden large jump as steep cliff and a single peak last for a while and then a sudden drop back to normal level.

3.3 Diagnostics

The diagnostic tests would validate and examine stability of the model based on the seasonal adjustment. The revision histories diagnostics test the model performance parameters for different time spans, as shown in Figure 5. Plots are shown from Q statistics q/q2, M statistics, log-likelihood and AIC. The dashed lines in each plot indicate the boundary of 20% threshold, as the changes of either performance parameters or forecast error would be limited within 20% of changing interval. Almost all values across time spans stay within the boundary, indicating the stability of the model. Only 1 out of 20 time spans has a value of log-likelihood out of the boundary, and 2 out of 20 time spans have a value of m statistics touching the limit. The diagnostics tests show the stable seasonal adjustment in the model performance evaluation parameters.

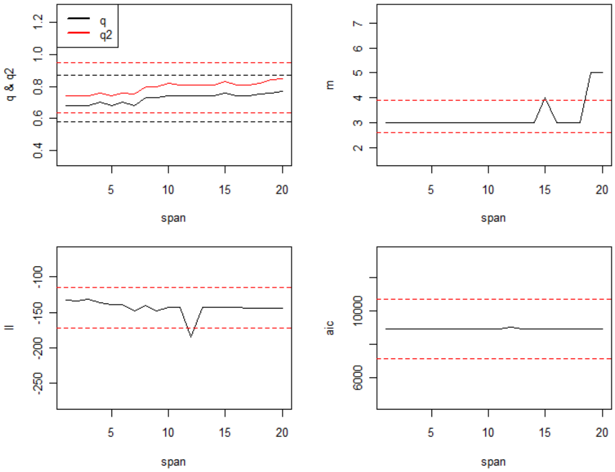


Fig. 5. Diagnostics: model performance

4 Forecast Errors

The forecast error can be explored in the model based on X12-ARIMA by estimating the last data cycle from the model based on the data without the last cycle. Similarly, the last two cycles can be estimated by the model based on the data without the last two cycles.

$$\text{Forecast Error} = \frac{\text{Forecast} - \text{True Traffic}}{\text{True Traffic}} * 100\%$$

Figure 6 shows that the forecast errors are less than 20% in most cases, and around 10% in some cases. The forecast error is smaller in the next day prediction, and increases over time because of the uncertainty into the future with less information. The extreme events may cause a spike in the forecast errors, but the forecast errors are still within a good acceptance range.

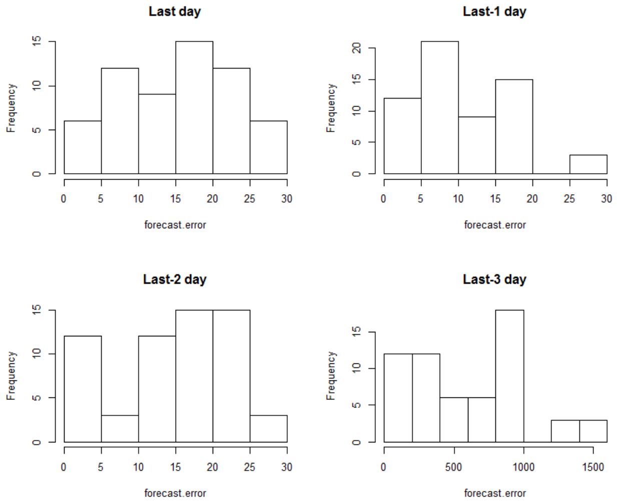
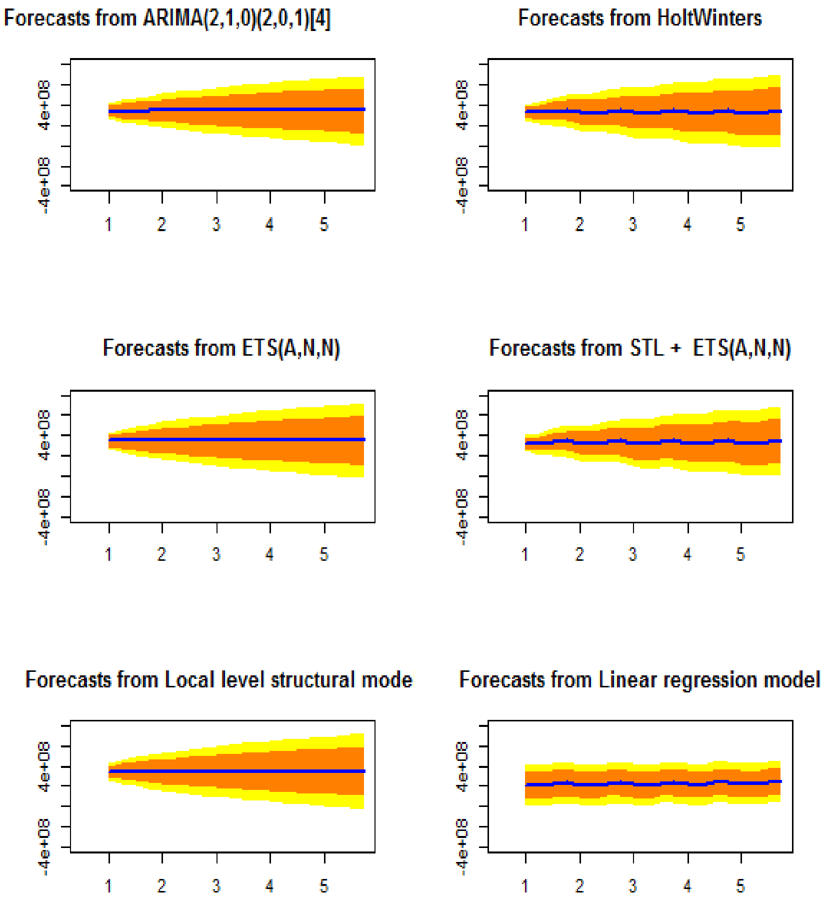


Fig. 6. Forecast error based on X12-ARIMA (%)

In comparison of the forecast performance among 6 different methods: ARIMA model, ETS model, Holt-Winters method, STL model, linear regression model and local level structural model, time series fluctuation is considered in three methods; Holtwinters by minimizing the squared prediction error using Holt-Winters Filtering [13,20], STL and linear regression fit model with explanatory variables as trend and seasonality components. Other methods include ARIMA based on the AIC (Akaike information criterion), Exponential smoothing state space model (ETS) [4,16] based on AIC determine its triplet (E,T,S) which denote additive or multiplicative model for error and trend and the presence of seasonality, and local level structural methods with state-space models [11] fitted by maximum likelihood.

Figure 7 shows the prediction with point estimator and two confidence interval forecasts. The orange shades indicate the 90% confidence interval, and the yellow shared represents the 95% of confidence interval. The forecast by the seasonal adjustment method based on STL offers predictions for seasonal variation, while the traditional ARIMA, ETS and local structure models only give estimators being same all the time without considering fluctuation. The ARIMA, ETS and local structure models do not provide forecasts without using full periodic information in the original data series, when the data show a periodicity. The method based on STL provides the smallest confidence interval among the 6 methods, and the accuracy of the prediction with the seasonal adjustment is better than the rest of the traditional methods. The improvement in the forecast error is mainly because of the new variables in the time series model to explain the original series. The newly added variables include three seasonal components, seasonality, trend and residual, as well as variables of outliers and missing values. These new additions capture more features of the data and generate better prediction for the time series.


Forecasts from ARIMA(2,1,0)(2,0,1)[4]

Forecasts from HoltWinters

Forecasts from ETS(A,N,N)

Forecasts from STL + ETS(A,N,N)

Forecasts from Local level structural mode

Forecasts from Linear regression model

Fig. 7. Compare with other models

5 Conclusion and Future Work

In this paper, we presented statistical models for estimating and forecasting network traffic based on the statistical patterns found in the network measurement SNMP data. The three steps Seasonal Adjustment procedure will enable us to analyze the seasonality existed in scientific data such as network traffic. We first determine the cycling period with significant seasonality is daily and with application X12-ARIMA and STL we decompose the original series into three components: seasonal component, trend component and irregular component. The diagnostics test is adopted to assess the credibility of our model. The prediction error derived from our model is on average within 20% and it shows superior results in terms of narrower confidence interval with less prediction interval when compared with 6 traditional time series models on network traffic. Our ongoing work includes further explored the usage of the three components resulted from Seasonal Adjustment procedure. To list a few, we will use trend component to trace the data flow over the whole network map and use seasonal component to plan routine data transfer. With combination of all three components, we can plan the future data transfer based on the prediction of network traffic condition. Long-term prediction of future network traffic development could also enable us to wisely allocate the infrastructure and links within the network.

Acknowledgments. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

1. Alarcon-Aquino, V., Barria, J.A.: Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 36(2), 208–220 (2006)
2. Box, G., Jenkins, G.: Time series analysis: Forecasting and control. Wiley Series in Probability and Statistics (1970)
3. Cleveland, W.S.: STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics 6(1), 3–73 (1990)
4. Durbin, J., Koopman, S.J.: Time Series Analysis by State Space Methods. Oxford University Press (2001)
5. Feng, H., Shu, Y.: Study on Network Traffic Prediction Techniques. In: Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 1041–1044 (2005)
6. Findley, D.F., Hood, C.H.: Seasonal Adjustment Procedures. In: Experiences and Perspectives, Istituto Nazionale di Statistica, Rome, pp. 231–251 (2000)
7. Findley, D.F.: Some Recent Developments and Directions in Seasonal Adjustment. Journal of Official Statistics 21(2) (2005)
8. Findley, Monsell, Bell, Otto, Chen: New Capabilities and Methods of the X-12-ARIMA Seasonal Adjustment Program. Preprint version of JBES (1998)

9. Findley, M., Shulman, P.: Sliding Spans Diagnostics for Seasonal and Related Adjustments. Journal of the American Statistical Association 85, 345–355 (1990)
10. Granger, C.W.J.: Seasonality: Causation, Interpretation, and Implications. In: Zellner, A. (ed.) NBER Book Seasonal Analysis of Economic Time Series, pp. 33–56 (1979)
11. Harvey, A.C.: Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press (1989)
12. Higginson, J.: F-test for the Presence of Moving Seasonality. Statistics Canada, Seasonal Adjustment Methods (1975)
13. Holt, C.C.: Forecasting trends and seasonals by exponentially weighted moving averages. International Journal of Forecasting 20(1), 5–10 (1957)
14. Hood, C.H., Monsell, B.C.: Getting Started with X-12-ARIMA Input Files on Your PC (2002), http://www.census.gov/ts/papers/gettingstartedx12.pdf
15. ESnet, http://stats.es.net
16. Hyndman, R.J., Akram, M.D., Archibald, B.: The admissible parameter space for exponential smoothing models. Annals of Statistical Mathematics 60(2), 407–426 (2008)
17. Lothian, J., Morry, M.: A Test of Quality Control Statistics for the X-11-ARIMA Sea-sonal Adjustment Program. Research Paper. Seasonal Adjustment and Time Series Staff. Statistics Canada (1978)
18. Qian, Y., Xia, J., Fu, K., Zhang, R.: Network Traffic Forecasting by Support Vector Machines Based on Empirical Mode Decomposition Denoising. In: 2012 2nd International Conference on Consumer Electronics, Communications and Networks, CECNet (2012)
19. Seasonal Adjustment Diagnostics and Supporting Document A (Checklists). Census Bureau, https://www.census.gov/ts/papers/G18-0_v1.1_Seasonal_Adjustment.pdf
20. Winters, P.R.: Forecasting sales by exponentially weighted moving averages. Management Science 6, 324–342 (1960)
21. X-12-ARIMA Seasonal Adjustment Program, http://www.census.gov/srd/www/x12a/
22. X-12-ARIMA Version 0.3 Reference Manual (2006), http://www.census.gov/ts/x12a/v03/x12adocV03.pdf
23. Zhao, H., Ansari, N.: Wavelet Transform-based Network Traffic Prediction: A Fast On-line Approach. Journal of Computing and Information Technology 20(1), 15–25 (2012)

A Lightweight Combinatorial Approach for Inferring the Ground Truth from Multiple Annotators

Xiang Liu[1], Liyun Li[2], and Nasir Memon[1]

[1] Polytechnic Institute of New York University
2 Metrotech Center, Brooklyn, NY, 11201
xliu15@students.poly.edu, memon@poly.edu
[2] Linkedin Corporation,
2029 Stierling Ct, Mountain View, 94043
liyli@linkedin.com

Abstract. With the increasing importance of producing large-scale labeled datasets for training, testing and validation, services such as Amazon Mechanical Turk (MTurk) are becoming more and more popular to replace the tedious task of manual labeling finished by hand. However, annotators in these crowdsourcing services are known to exhibit different levels of skills, consistencies and even biases, making it difficult to estimate the ground truth class label from the imperfect labels provided by these annotators. To solve this problem, we present a discriminative approach to infer the ground truth class labels by mapping both annotators and the tasks into a low-dimensional space. Our proposed model is inherently combinatorial and therefore does not require any prior knowledge about the annotators or the examples, thereby providing more simplicity and computational efficiency than the state-of-the-art Bayesian methods. We also show that our lightweight approach is, experimentally on real datasets, more accurate than either majority voting or weighted majority voting.

Keywords: Crowdsouring, Annotator-Task Model, Weighted Majority Voting, Combinatorial Model, Social Computing.

1 Introduction

In many machine learning application areas, it is vital to obtain large-scale labeled datasets for training, testing and validation. However, consider labeling tasks such as annotating millions of documents for natural language processing [1] or describing thousands of pictures for computer vision research [2]. The tediousness of these kinds of tasks make it difficult for researchers to obtain large-scale labeled datasets, which is becoming more and more a bottleneck that impedes the application of new learning algorithms.

To tackle the scarcity of such large-scale labeled datasets, crowdsourcing services such as Amazon Mechanical Turk(MTurk) [3] has been introduced. Crowdsourcing has now greatly changed the way with which the large-scale datasets

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 616–628, 2013.

are being created. Rather than being produced "by hand", it is now possible to distribute the task of labeling large-scale datasets into the crowd. Using these crowdsouring services, one can expect labelers in the crowd to annotate a large number of examples in short time, usually with a relatively small financial cost. However, such collected wisdom does not necessarily reveal the underlying ground-truth class labels. And the quality of these labels varies as the labelers in the crowd are heterogeneous in terms of skill, expertise and even consistency. Therefore, rather than solving the problem, crowdsourcing alleviates the scarcity of large-scale labeled datasets, where the problem of inferring ground-truth labels from the crowd still remains.

One common strategy for obtaining an estimate of the ground truth is majority voting. However, as majority voting assumes that each annotator is equally good, the quality of the majority label is effected by individual annotator's biases. In extreme cases, if there is only one expert who gives the correct label while all the others give incorrect labels, then the majority voting method would always prefer the incorrect label. To address these problems, we assume that each labeling task corresponds to a point x in some low dimensional space, and that each annotator labels point x according to a linear threshold rule. By mapping the tasks and annotators onto a low dimensional space, we aim to minimize the number of errors introduced in each step.

In this paper, we propose a discriminative approach to learn the underlying ground truth labels using only information of labels provided by multiple annotators in the crowd. Our algorithm handles the fact that the labels from these crowdsourcing annotators are noisy and provides a local optimal estimate of the underlying golden ground-truth. More specifically, our contributions are:

1. Instead of using a generative model to restrict the annotation process, we formalize the problem as a combinatorial problem and solve it using a discriminative approach. As our approach is non-parametric and does not require any prior knowledge or tuning any parameters as a priori, it can serve as a complimentary approach to other Bayesian generative approaches, which are usually more complicated by attempting to model the entire subjective process of labeling an object.
2. This approach incurs no training complexity and the estimated ground truth can be obtained almost directly. Thus, it provides more simplicity and computational efficiency than state-of-the-art Baysian methods.
3. Even though our algorithm is lightweight, simple and efficient, we show that, by experiments on multiple datasets, it performs consistently and significantly better than majority voting and simple weighted majority voting. Therefore, our algorithm could be an competing alternative approach to majority voting, as well as a complimentary lightweight method for other more sophisticated Bayesian algorithms.

The rest of this paper is organized as follows. In Section 2 we discuss related work on modeling heterogenous annotators and annotation process. Then we describe our approach in details in Section 3 and Section 4 and present the experimental results in Section 5. We conclude in Section 6 and discuss the limitations and future work.

2 Related Work

Recent research interests in learning from crowdsourcing arise from areas like natural language processing [1] and computer vision [2]. Crowdsourcing is used to collect labels for machine learning [11] [12] [13].Previous work has mainly focused on obtaining more reliable labels. The measures for evaluating these obtained labels rely on either comparing with the gold ground-truth (if they exist) or optimizing certain utility metrics for these labels [4] [5] [6]. The most common method for using crowdsourcing to label data is to obtain multiple labels for each object from labelers and label the object as the majority label [7].

In [8], the authors propose to model heterogeneous annotators by their sensitivity and specificity. The labeling process is modeled as linear threshold decisions with different levels of biases specified by each annotator's sensitivity and specificity. [9] and [10] proposed Bayesian probabilistic models for the annotation process. [10]'s approach also models the annotation process as a linear threshold decision process, where the model is built on a number of parametric distributions: Mixture of Gaussians for the images, Gaussian on the weight vectors for the annotators, and Gaussian on the thresholds. The authors use an iterative process to find the optimal parameters by MAP estimation. Given that these models achieve decent performances, to accurately estimate the posterior distributions, all these Bayesian generative models require certain prior knowledge (on either the annotators or the tasks). Also the complexities for training and prediction are usually non-trivial. The work of Dawid and Skene [14] assumes that each worker is associated with a probabilistic confusion matrix that generates his labels. Each entry of the matrix indicates the probability that items in one class are labeled as another. Given the observed labels, the true labels for each item and the confusion matrices for each worker can be jointly estimated by a maximum likelihood method. Zhou's paper [15] proposed a minimax entropy principle to jointly estimate the distributions and the ground truth given the observed labels by workers.

Rather than adopting any probabilistic model, by drawing inspiration from [16], who mapped the viewing region of each feature onto a circle in 2D space, we propose an approach to model the annotators and labeling tasks onto 1D and 2D euclidian spaces. We use a two-phase algorithm to search for the "optimal" annotator-task arrangement. Both steps work in greedy fashion. Our approach achieves better performance on real datasets than majority voting and weighted majority voting and provides a complimentary method to estimate the underlying labels without priori knowledge.

3 Modeling Annotators and Tasks

In each binary labeling task, an annotator j, looks at task i and assigns it label l_{ij} (0 or 1) according to the his expertise to the specific task. The labels collected from m tasks and n labelers form a sign matrix with each row representing a task and each column representing an annotator. We assume that each annotator labels a task according to a linear threshold rule in some low dimensional space (1D, 2D and 3D) and each task is represented by a single point in that space. Suppose the sign matrix is the only information we have, without knowing any prior knowledge or defining any parameter, in the situation of $d \in \{1, 2, 3\}$, we can model the annotators and tasks as follows.

Let $F = \{f_1, ..., f_n\}$ be the collection of half-spaces for annotators given by a normal direction and threshold; $f_i = \{h_i, t_i\}$, where $h_i \in \{x = (x_1, ..., x_d) : \|x\| = 1\}$ and $t_i \in \mathbb{R}$. Let $V = \{v_1, \ldots, v_m\}$ be a finite set of points on the space. If $d = 1$, $v_i \in \mathbb{R}$ while in the cases of $d = 2, 3$, $v_i = \{(x_1, ..., x_d : \|x\| = 1)\} \in \mathbb{R}^d$. We are interested in finding a sign matrix M that satisfies:

$$M_{ij} = \{ {}^{1 \quad if\ v_i^T h_j > t_j,}_{0 \quad if\ v_i^T h_j < t_j} .$$

Any sign matrix arising from a collection of points and half-spaces in this way will be called a "corrected" matrix. The Fundamental problems we consider in this work are: Given a sign matrix M, can we find a set of half-spaces F and a set of defining points V leads to the closest "corrected" matrix M^* ? With the "corrected" matrix M^*, how to label each row in M^* ?

The problem of deciding if M is a "corrected" matrix in three-dimensions is NP-hard [16]. Thus we only provide efficient algorithms for $d \in \{1, 2\}$.

3.1 1D Mapping Method

In the case of 1D mapping algorithm, by the definition of V, the tasks are mapped to points in 1D space(a line more specifically), where each row is a sign vector representing the set of labels given to a particular task. The input of the algorithm is an $m \times n$ sign matrix M, we would like to:

1. Find a set of linear functions $F = \{f_1, ..., f_n\}$ that intersect the line and divide the line into $n + 1$ cells $C = \{C_1, ..., C_{n+1}\}$ such that by introducing this arrangement, the accumulated error is minimized. Here error is the number of rows in M that do not satisfy the definition of "corrected" matrix by any chance of placement. Cell $C_i = (p_1, ..., p_n), p_j = 1$ if C_i is in the positive half space defined by f_j ; $p_i = 0$ otherwise. For example, if annotators i and j intersect the line and divide it into three cells with patterns 11, 10, and 01. Then task $t = (0, 0)$ could not be placed, thus an error arises.
2. Arrange the set of points $V = \{V_1, ..., V_m\}$ into the cells that we have defined. V_i is placed into cell C_t if and only if C_t has the least number of contradiction with the i-th row of the sign matrix $(M_{i1}, M_{i2}, ..., M_{in})$ among all cells in C.
3. Now we have the half-spaces F that lead to the closest "corrected" matrix of the original sign matrix.

3.2 2D Mapping Method

Now we consider the case of 2D mapping arrangement. For simplicity, by the definition of V, the tasks are mapped to points on a circle. Recall the definition of annotators, $f = (h_i, t_i)$, where $h_i \in S^{d-1}$, and $t_i \in \mathbb{R}$. If $t_i = 0$ the half-space f_i is called central arrangement. We assume that the half-spaces $F = \{f_1, f_2, ..., f_n\}$ are of non-central arrangements, which means the linear functions in F are not necessarily cross the center of the circle. The input of the algorithm is an $m \times n$ sign matrix M, we would like to:

1. Find a set of linear functions $F = \{f_1, ..., f_n\}$ that intersect the circle and divide the circle into $t(4 \leq t \leq 2n)$ cells $C = \{C_1, ..., C_t\}$ such that in this arrangement, the accumulated error is minimized. Here error is defined in the same way as in 1D mapping method, which the number of rows in M that do not satisfy the definition of "corrected" matrix by any chance of cell placement.
2. Arrange the set of points $V = \{V_1, ..., V_m\}$ into the cells that we have defined. V_i falls into the cell C_t if and only if C_t has the least number of contradiction with $(M_{i1}, M_{i2}, ..., M_{in})$ among all cells in C.
3. Finally we have the points V and half-spaces F that leads to the closest "corrected" matrix of the original sign matrix.

The difference between 1D and 2D mapping approaches lies in that in each step of annotators mapping, the number of cells are increased by 1 and 2 respectively, as shown in Fig 1.

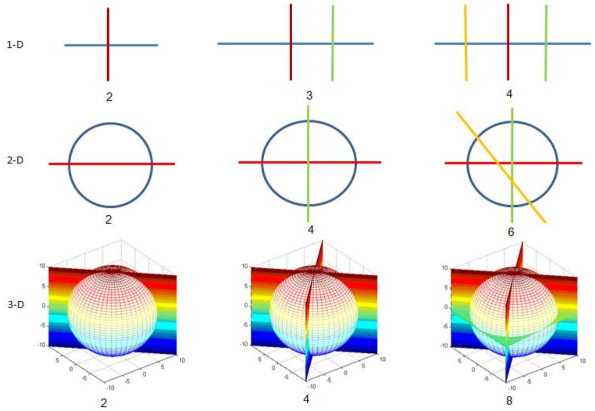


Fig. 1. Examples of half spaces separate the 1D, 2D and 3D spaces according to some linear threshold functions. It is showed that with an increasing number of half spaces which correspond to annotators, the space is divided into $\{2, 3, 4\}$, $\{2, 4, 6\}$, and $\{2, 4, 8\}$ different cells in 1D, 2D and 2D arrangements respectively. However, there should be 2^n possible combinations given the binary labels provided by n annotators. The missing number of cells introduces errors into the mapping arrangements.

4 Mapping Algorithms and Label Estimation

We will introduce in this section the two-phase mapping algorithms to obtain the annotator-task "corrected" sign matrix. We will introduce the details of 2D mapping algorithm and omit the process for 1D mapping. Both phases work in greedy fashion.

4.1 Mapping Annotators

As the errors introduced by intersecting lines(annotators) into the circle increase exponentially, the order of annotators to map is vital. We first construct a graph G where each node represents a column of the original sign matrix. There is an edge (i, j) between node i and node j if and only if column i and column j cross each other. Two columns must cross exactly when we see all four possible sign patterns on columns i and j of M. Let $S = \{s_1, ..., s_u\}$ be the connected components of G, $|s_1| > |s_2| > \ldots > |s_u|$. There is an ordering of the columns in s_i such that each column must cross a column that appears before it. Then list all the columns in such order: $s_1, \ldots, s_u$.

The input of the algorithm is the sign matrix M and the ordering of the columns. We would like to obtain a cyclic order A that captures the relative ordering of intersection points of F with the circle. Initially, we insert the 1st column and the 2nd column which cross each other obviously, after the insertion, A = $\{\{l_1\}\{l_2\}\{l_{1'}\}\{l_{2'}\}\}$ or $\{\{l_1\}\{l_{2'}\}\{l_{1'}\}\{l_2\}\}$. Then from the third to the n^{th} column, we greedy search for all the possibilities(the possibilities that line f_i's two ends intersect with two different cells are: f_i's two ends intersect with one cell; f_i's two ends intersect with two existing intersection points; f_i's one end intersects with a cell, and the other end intersects with an existing intersection point. And f_i, $f_{i'}$ can switch their orders.) that line f_i could intersects the circle, and choose the optimized arrangement that introduces the least number of rows not satisfying the definition of "corrected" matrix for the first i columns in S, $F_i = \{f_1, \ldots, f_i\}$.

4.2 Mapping Tasks

This phase still works in a greedy search. For each task item i, we place V_i into cell C_t if and only if the placement introduces the least number of contradiction with the i-th row of original sign matrix, (M_{i1}, M_{i2},..., M_{in}), among all cells in C. If there are more than one cell's placement obtain the least error number, just randomly choose one cell among them.

4.3 Label Estimation

As shown in the first figure in Fig 2, we have the closest "corrected" matrix to the original sign matrix by using annotator-task model. Then we propose two methods to infer the ground truth for each task. One simple idea is to apply majority voting on the "corrected" matrix and receive the estimated labels.

Another approach is to estimate the labels by modeling annotators' behavior based on the annotator-task placement.

As shown in the second figure in Fig 2, each annotator corresponds to a line that separates the circle into two cells and intersects the circle with two points. Since each annotator has different levels of annotation skills, consistency or bias, we want to assign each annotator a weight that measures how reliable he is. We provides the weight for annotators by first grouping the adjacent intersected points of lines and the circle into one group if they have the same normal vector direction, h_i. We assume that each point within the same group has same weight, which is in proportion to the number of points in that group. The weight of each annotator is the sum of its two intersected points with the circle. Then the label of task v is obtained by:

$$w = \sum_{f_i \cdot v > t_i} w_i - \sum_{f_j \cdot v < t_j} w_j .$$

If $w \geq 0$, we label it as positive, otherwise, label it as negative. For example, if we have a cyclic order $A = \{1, 3, 5', 4', 2, 3', 5, 1', 4, 2'\}$, then we group the intersected points into 8 groups: $\{1, 3\}, \{5', 4'\}, \{2\}, \{3'\}, \{5\}, \{1'\}, \{4\}, \{2'\}$. The weights for each group are $0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1$.

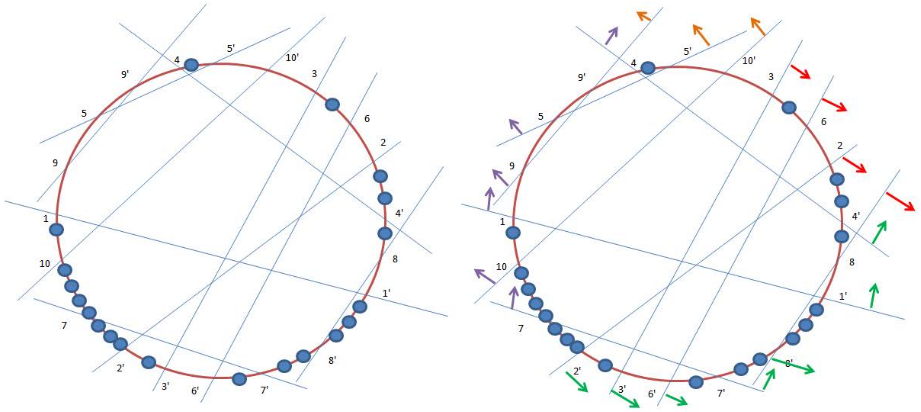


Fig. 2. How 2D mapping method works on a specific sign matrix in the joy dataset. The arrangements of annotators and tasks are shown in the left figure. The annotators' behavior can be modeled according to the right figure.

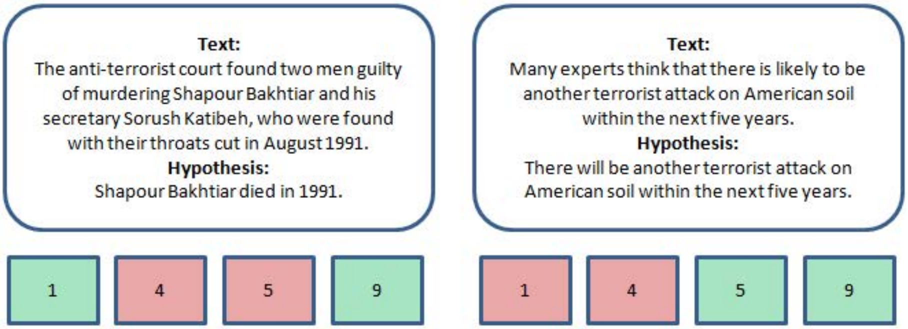
5 Experiment

In this section, we present experimental results, and compare our proposed method with the baselines, majority voting method and weighted majority voting method.

5.1 Datasets

SNOW datasets [1] is one of popular emotion datasets including seven task-annotator matrices, including ANGER, DISGUST, FEAR, JOY, SADNESS, Temporal Event Recognition (RTE) and Temporal Event Recognition (TEMP). The first five matrices are numeric matrices but not sign matrices. For simplicity, we applied Weka [18] unsupervised NumerictoBinary filter for attributes to transform them into sign matrices.

RTE and TEMP are sign matrices shown in Fig 3. Each RTE instance in the dataset is a pair of sentences. Annotators are asked to label whether the second



Text:
The anti-terrorist court found two men guilty of murdering Shapour Bakhtiar and his secretary Sorush Katibeh, who were found with their throats cut in August 1991.
Hypothesis:
Shapour Bakhtiar died in 1991.

Text:
Many experts think that there is likely to be another terrorist attack on American soil within the next five years.
Hypothesis:
There will be another terrorist attack on American soil within the next five years.

Fig. 3. Examples of binary labels of RTE and TEMP obtained from Amazon Mechanical Turk. The boxes show the binary labels provided by four annotators 1, 4, 5, 9. Green box indicates the second sentence (the Hypothesis) can be implied from the first sentence (the Text), i.e., the Hypothesis can be determined to be true given that the Text is true. Red box means that the second sentence can not be inferred from the first sentence. It is assumed that annotators do not have any prior knowledge before the Text is shown.

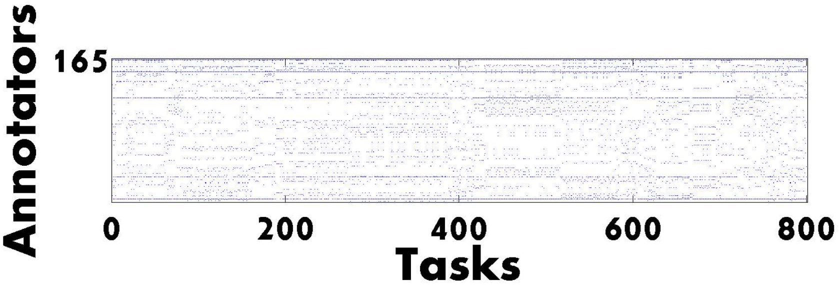


Fig. 4. RTE label matrix. The x-axis is tasks ID from 1 to 800. The y-axis is annotators ID from 1 to 165. Blue cells indicate the according task is labeled by the corresponding annotator. White cells denote tasks with missing values. It is shown that the matrix is sparse with 94% values missing in total.

sentence can be inferred from the first sentense. RTE has 800 instances with 165 annotators. Each task is labeled by 10 annotators and every group of 20 tasks are shared among the same group of annotators, resulting in 94% of items' value missing in the matrix, as shown in Fig 4. Each TEMP instance is a short article including two events. The annotators need to judge which of the two events happen first. The original data set has 462 instances and 76 annotators. Each task in TEMP is labeled by 10 annotators and every 10 tasks are shared among the same group of annotators.

Table 1 shows the summary of the seven parts of the dataset.

Table 1. Summary of the SNOW datasets collected from Amazon Mechanical Turk showing the number of tasks per dataset, the number of labels per task and the total number of annotators that provided labels

Dataset	Tasks	Assignments	Annotators
RTE	800	10	165
TEMP	462	10	76
Anger	100	10	38
Disgust	100	10	30
Fear	100	10	32
Sadness	100	10	38
Joy	100	10	38

5.2 Experimental Results and Discussions

As introduced in section 3 and 4, we implemented our algorithms according to the two phases and estimated the labels using the simple method(perform majority voting on the "corrected" matrix). With the ground truth, we compared the prediction accuracy of our proposed method with the other two baselines, majority voting method and weighted majority voting method.

RTE Dataset. Although the RTE dataset contains 800 tasks and 165 annotators, each task is only labeled by 10 annotators, and every 20 tasks are shared by the same annotator group. Thus we first divided the whole sign matrix into forty 20-by-20 sign matrices.

As shown in Fig 6, our model on 2D space performs better estimation of the ground truth than majority voting while the majority voting beats the 1D mapping method. By the default scenario of majority voting method, if half or more than half of the annotators label a task as positive(5 or more annotators in our experiment), then the task is labeled as positive. If a strategy is considered as "t voting strategy" when labeling a task as positive if t or more than t annotators label it as positive, then we can conclude from Fig 6 that if $t \geq 5$, both "t voting strategy" on "corrected" matrices processed by 1D and 2D mappings achieve better performance than "t voting strategy" itself.

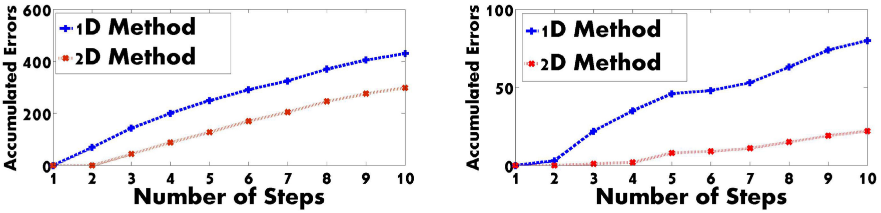


Fig. 5. The left figure shows the accumulated errors for 1D and 2D arrangements on RTE dataset. The right figure shows the accumulated errors for 1D and 2D arrangements on TEMP dataset. The x-axis corresponds to the number of linear separators mapped into the space.

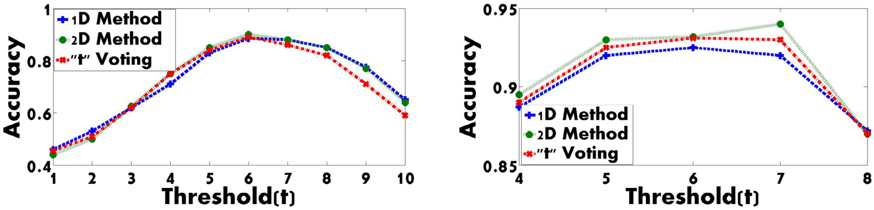


Fig. 6. The performance of 1D approach, 2D approach and majority voting on both RTE dataset and TEMP dataset

TEMP Dataset. TEMP contains 462 tasks and 76 annotators. Each task is only labeled by 10 annotators and every 10 tasks are shared by the same annotator group. Similar to RTE, we first divide the whole sign matrix into forty-six 10-by-10 sign matrices.

The TEMP experiment(right picture in Fig 6) shows that our model on 2D space performs slightly better than the majority voting method while the 1D approach performs no better than majority voting. In this experiment, both "t voting strategy" on "corrected" matrices achieved by 1D and 2D methods perform nearly the same with "t voting strategy". This can be explained by the fact that the number of accumulated errors introduced to TEMP by the algorithm is far less than the one in RTE shown in Fig 5, which implies that the original sign matrix of TEMP dataset is close to the "corrected" matrix.

The Other Datasets. For the other 5 datasets, we obtain similar result as in RTE and TEMP datasets that the mapping method on 2D space performs better than majority voting method while majority voting method performs better than 1D approach.

If more "priori information"(the number of positive labels in a given sign matrix) is introduced, we can obtain further improvement on estimation for both 1D and 2D approaches compared with majority voting method shown in Fig 7.

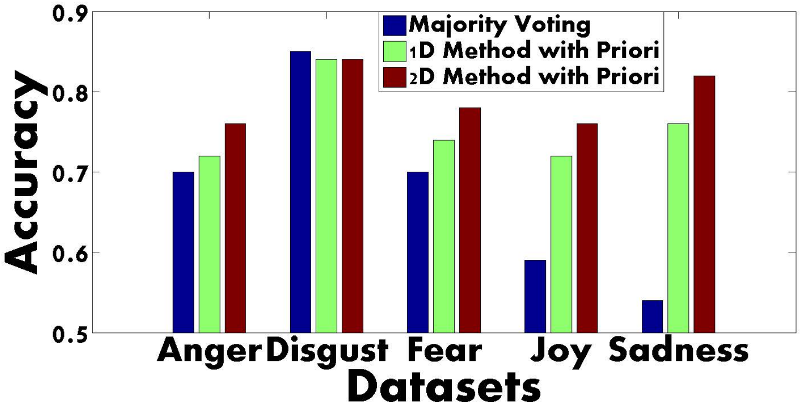


Fig. 7. The overall performance of majority voting, 1D approach with priori knowledge and 2D approach with priori knowledge for the five datasets.

Weighted Majority Voting. To further prove the effectiveness of our methods, we also compared our proposed method with the other baseline, weighted majority voting [17], which the weights for annotators are learned iteratively with a penalty parameter beta. Experimental results on precisions are shown in the bar graph in Fig 8.

From the graph, it could be seen that our 2D Weight-based Estimation on "corrected" matrix performs significantly better than the other baselines on RTE dataset while it performs slightly better than the other three baselines on TEMP dataset.

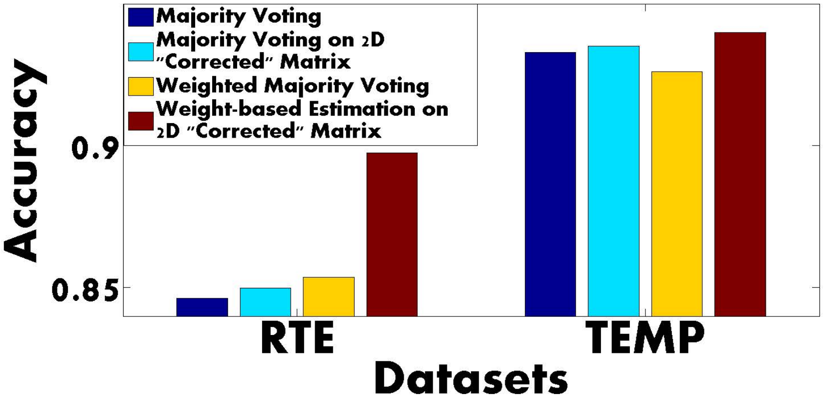


Fig. 8. The performance of majority voting, 2D simple approach and 2D weight-based estimation on corrected matrix for the RTE and TEMP datasets

Issue with Errors. As shown in Fig 1, the number of cells increases by 1 and 2 for 1D and 2D mapping approaches in each step while all the possibilities for annotator-label increases exponentially. Such mismatch produce errors. The experimental result shows how errors increase, which is demonstrated in Fig 5. It can be inferred from the figures that the 2D arrangement introduces half less errors than the 1D arrangements, which intuitively makes sense as we discussed.

6 Conclusions

In this paper, we propose a model to map an annotator-task sign matrix M onto a line/circle in 1D/2D spaces respectively. Each annotator corresponds to a line that crosses the line/circle in 1D/2D space and each task corresponds to a point that fall onto the line/circle. Given the original matrix M, the closest "corrected" matrix to M and is computed and the binary label for each task is learned. Then we compare it with the majority voting and weighted majority voting on the initial sign matrix to see the prediction accuracy for the two approaches.

Our method achieves better result than the majority voting and weighted majority voting in most of the datasets that we tested. The proposed algorithm can handle noisy data and provide a method to model the behavior of each annotator. Our findings reveal that annotators can be clustered into different groups. And for certain tasks, there are expert annotators that give "better" labels according to their skills, consistencies and biases. However, our model can only deal with binary labels. Applying our model to continuous annotator labels will be explored in the future.

References

1. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and Fast - But is it Good? Evaluating NonExpert Annotations for Natural Language Tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 254–263. ACL (2008)
2. Sorokin, A., Forsyth, D.: Utility data annotation with Amazon Mechanical Turk. In: Proc. of CVPR 2008, pp. 1–8 (2008)
3. Amazon Mechanical Turk, `https://www.mturk.com/mturk/welcome`
4. Raykar, V.C., Yu, S., Zhao, L.H., Jerebko, A., Florin, C., Valadez, G.H., Bogoni, L., Moy, L.: Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In: Proc. of ICML 2009, pp. 889–896 (2009)
5. Smyth, P., Fayyad, U., Burl, M., Perona, P., Baldi, P.: Inferring Ground Truth from Subjective Labelling of Venus Images. Advances in Neural Information Processing Systems 7, 1085–1092 (1995)
6. Yan, Y., Rosales, R., Fung, G., Schmidt, M.W., Valadez, G.H., Bogoni, L., Moy, L., Dy, J.G.: Modeling annotator expertise: Learning when everybody knows a bit of something. Journal of Machine Learning Research - Proceedings Track (JMLR) 9, 932–939 (2010)
7. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: Proceeding of the 14th International Conference on Knowledge Discovery and Data Mining (KDD), pp. 614–622 (2008)

8. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L.: Linda Moy: Learning From Crowds. Journal of Machine Learning Research 11, 1297–1322 (2010)
9. Whitehill, J., Ruvolo, P., Wu, T.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: NIPS (2009)
10. Welinder, P., Branson, S., Belongie, S., Perona, P.: The multidimensional wisdom of crowds. In: Advances in Neural Information Processing Systems, NIPS (2010)
11. Ertekin, S., Hirsh, H., Rudin, C.: Approximating the wisdom of the crowd. In: Proceedings of the Workshop on Computational Social Science and the Wisdom of Crowds (2011)
12. Kamar, E., Hacker, S., Horvitz, E.: Combining human and machine intelligence in largescale crowdsourcing. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, pp. 467–474 (2012)
13. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in Neural Information Processing Systems, vol. 24, pp. 1953–1961 (2011)
14. Dawid, A.P., Skene, A.M.: Maximum likeihood estimation of observer error-rates using the EM algorithm. Journal of the Royal Statistical Society 28(1), 20–28 (1979)
15. Zhou, D., Platt, J., Basu, S., Mao, Y.: Learning from the Wisdom of Crowds by Minimax Entropy. In: NIPS (2012)
16. Basri, R., Felzenszwalb, P.F., Girshick, R.B., Jacobs, D.W., Klivans, C.J.: Visibility constraints on features of 3D objects. In: CVPR, pp. 1231–1238 (2009)
17. Blum, A.: Empirical support for winnow and weighted-Majority algorithms: results on a calendar scheduling domain. Machine Learning 26, 5–23 (1997)
18. Frank, Eibe, et al.: Weka. Data Mining and Knowledge Discovery Handbook, 1305–1314 (2005)

Large Scale Visual Classification with Many Classes

Thanh-Nghi Doan[1], Thanh-Nghi Do[3], and François Poulet[1,2]

[1] IRISA,
[2] Université de Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex, France
{thanh-nghi.doan,francois.poulet}@irisa.fr
[3] Institut Telecom, Telecom Bretagne UMR CNRS 6285 Lab-STICC, Brest, France
Université européenne de Bretagne, France, Can Tho University, Vietnam
tn.do@telecom-bretagne.eu

Abstract. The usual frameworks for visual classification involve three steps: extracting features, building codebook and encoding features, and training classifiers. The current release of ImageNet dataset [1] with more than 14M images and 21K classes makes the problem of visual classification become more difficult to deal with. One of the most difficult tasks is to train a fast and accurate classifier. In this paper, we address this challenge by extending the state-of-the-art large scale classifier Power Mean SVM (PmSVM) proposed by Jianxin Wu [2] in two ways: (1) The first one is to build the balanced bagging classifiers with under-sampling strategy. Our algorithm avoids training on full data and the training process of PmSVM rapidly converges to the optimal solution, (2) The second one is to parallelize the training process of all classifiers with multi-core computers. We have developed the parallel versions of PmSVM based on high performance computing models. The evaluation on 1000 classes of ImageNet (ILSVRC 1000 [3]) shows that our approach is 90 times faster than the original implementation of PmSVM and 240 times faster than the state-of-the-art linear classifier (LIBLINEAR [4]).

Keywords: Large Scale Visual Classification, High Performance Computing, Sampling Strategy, Parallel Support Vector Machines.

1 Introduction

Visual classification is one of the important research topics in the area of computer vision and machine learning. Low-level local features and bag-of-words model (BoW) are the core of state-of-the-art visual classification systems. The usual frameworks for visual classification involve three steps: 1) extracting features, 2) building codebook and encoding features, and 3) training classifiers. All these frameworks were evaluated on small datasets, e.g. Caltech 101 [5], Caltech 256 [6], and PASCAL VOC [7] that can fit into desktop memory. In step 3, most researchers choose either linear or non-linear SVM classifiers that can be trained in a few minutes.

However, ImageNet with very large number of classes poses more challenges in training classifiers. ImageNet is much larger in scale and diversity than the other benchmark datasets. The current release ImageNet has grown a big step in terms of the number of images and the number of classes, as shown in Fig. 1 - it has 21,841 classes with more than 1000 images for each class on average.

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 629–643, 2013.

With millions of images, training an accurate classifier may take weeks or even years [8], [9]. The recent works in large scale learning classifiers converge on building linear SVM classifiers. We are able to train linear classifiers (e.g. LIBLINEAR) in order of seconds, even with millions of training examples. However, in the context of visual classification, linear classifier is inferior in terms of accuracy, compared to non-linear classifiers [10], [11], [2]. Wu [2] proposes Power Mean SVM classifier that outperforms LIBLINEAR and other additive kernel classifiers in terms of training time and classification accuracy. Nevertheless, the current version of PmSVM does not take into account the benefits of high performance computing (HPC). On ILSVRC 1000, it takes very long time to train all binary classifiers. Therefore, it motivates us to study how to speed-up PmSVM for large scale visual classification. In this paper, we have developed the extended versions of PmSVM in two ways:

1. Propose a balanced bagging algorithm for training binary classifiers. Our algorithm avoids training on full data and the training process of PmSVM rapidly converges to the optimal solution.

2. Parallelize the training process of all binary classifiers based on HPC models. In the training step of classifiers, we apply our balanced bagging algorithm to achieve the best performance.

Our approach is evaluated on the 10 and 100 largest classes of ImageNet and ILSVRC 1000. The result shows that our approach is 90 times faster than the original implementation of PmSVM and 240 times faster than LIBLINEAR without (or very few) compromising classification accuracy.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work on large scale visual classification. Section 3 introduces Power Mean SVM. In section 4, we present its improvement for large number of classes and describe how to speed-up the training process of PmSVM by using our balanced bagging algorithm and take into account the benefits of HPC. Section 5 presents numerical results before the conclusion and future work.

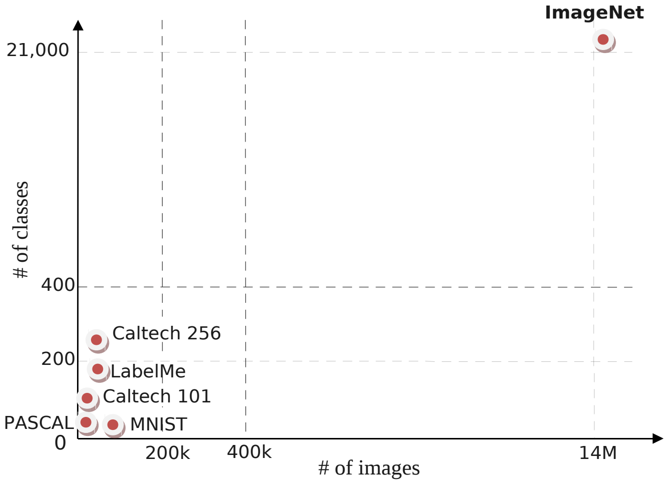


Fig. 1. A comparison of ImageNet with other benchmark datasets

2 Related Work

Many previous works on image classification rely on bag-of-words model (BoW) [12], local feature quantization and support vector machines. These models may be enhanced by multi-scale spatial pyramids [13] on BoWs or histogram of oriented gradient [14] features. Some recent works consider exploiting the hierarchical structure of dataset for image recognition and achieve impressive improvements in accuracy and efficiency [15]. Related to classification is the problem of detection, often treated as repeated one-versus-all classification in sliding windows [7], [16]. In many cases, such localization of objects might be useful for improving classification accuracy performance. However, in the context of large scale visual classification with hundreds or thousands of classes, these common approaches become computationally intractable.

To address this problem, Fergus *et al.* [17] study semi-supervised learning on 126 hand labeled Tiny Images categories, Wang *et al.* [18] show classification experiments on a maximum of 315 categories. Li *et al.* [19] do research with landmark classification on a collection of 500 landmarks and 2 million images. On a small subset of 10 classes, they have improved BoW classification by increasing the visual vocabulary up to 80K visual words. Furthermore, the current released ImageNet makes the complexity of large scale visual classification become a big challenge. To tackle this challenge, many researchers are beginning to study strategies on how to improve the accuracy performance and avoid using high cost non-linear kernel SVMs for training classifiers. The recent prominent works for these strategies are proposed in [8] [9], [20], [21] where the data is first transformed by a nonlinear mapping induced by a particular kernel and then efficient linear classifiers are trained in the resulting space. They argue that the classification accuracy of linear classifiers with high dimensional image representations is similar to low dimensional BoW with non-linear kernel classifiers. In [9], each local descriptor is coded either using Local Coordinate Coding [22] or Supper-vector Coding [23], after performing spatial pyramid pooling the resulting image representation is a vector in approximately 262K dimensions. To train classifiers, they propose a parallel averaging stochastic gradient descent (ASGD) algorithm. With 1000 classes from ILSVRC 1000, it takes 4 days to train 1000 binary SVM classifiers (one-versus-all) for one feature channel on three 8-core computers. Sánchez and Perronin [21] study the impact of high dimensional Fisher vectors on large dataset. They show that the larger the training dataset, the higher the impact of the dimensionality on the classification accuracy. To get the state-of-the-art result on ILSVRC 1000, they use the spatial pyramids to increase the dimensionality of their Fisher vectors to approximately 524K dimensions and then exploit Product Quantizier [24] to compress the data before training classifiers. With this approach, training 1000 SGD SVM classifiers (one-versus-all) for one feature channel takes 1.5 days on a 16-core computer.

In contrast with the approaches using the efficient linear SVMs, some recent works show that in the context of training classifiers for computer vision tasks, linear SVMs are inferior in terms of accuracy, compared to the kernel versions of SVM. In many cases of visual classification, learning SVMs with additive kernels give significantly higher rate in accuracy performance than dot product kernel [10] [11]. The main drawback of these approaches is the high cost of training non-linear kernel classifiers. It may be thousands times higher than linear classifiers. However, some recent solutions are proposed to solve this limitation [10], [11]. Additive kernel SVMs now use only few times more training time, compared to the state-of-the-art linear SVM classifiers. To design a fast and accurate non-linear kernel classifier for large scale dataset, Wu [2] proposes an efficient algorithm for PmSVM. They show empirically that PmSVM outperforms LIBLINEAR and the state-of-the-art additive kernel classifiers in terms of both training time and classification accuracy. For instance, with 1000 classes from ILSVRC 1000, PmSVM is 3 times faster than LIBLINEAR and 2 times faster than the state-of-the-art additive kernel implementations while getting a significant improvement in classification accuracy from +4.6% to +7.1%. However, the current version of PmSVM does not take the benefits of the modern chip manufacturing. On one core of our computer, it takes more than 18 hours to train 1000 binary classifiers on ILSVRC 1000.

In the multi-core era, computers with multi-cores or multiprocessors are becoming more and more popular and affordable. So it motivates us to investigate parallel solutions and demonstrate how PmSVM can benefit from modern platforms. Furthermore, in the case of large number of classes, we show that our balanced bagging algorithm is very useful to speed-up the training process of classifiers without (or very few) compromising classification accuracy. Our experiments show very good results and confirm that the balanced bagging algorithm and parallel solutions are very essential for large scale visual classification in terms of training time.

3 Power Mean Support Vector Machines

Let us consider a binary linear classification task with m datapoints in a n-dimensional input space $x_1, x_2, \ldots, x_m$ having corresponding labels $y_i = \pm 1$. SVM classification algorithm [25] aims to find the best separating surface as being furthest from both classes. It can simultaneously maximize the margin between the support planes for each class and minimize the error. This can be accomplished through the quadratic program (1).

$$min_{\alpha}(1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K\langle x_i, x_j\rangle - \sum_{i=1}^{m}\alpha_i$$
$$s.t. \begin{cases} \sum_{i=1}^{m} y_i \alpha_i = 0 \\ 0 \le \alpha_i \le C \quad \forall i = 1,2,...,m \end{cases} \tag{1}$$

where $K\langle x_i, x_j\rangle$ is a kernel function of x_i and x_j, C is a positive constant used to tune the margin and the error.

The support vectors (for which $\alpha_i > 0$) are given by the solution of the quadratic program (1), and then, the separating surface and the scalar b are determined by the support vectors. The classification of a new data point x is based on:

$$sign(\sum_{i=1}^{\#SV} y_i \alpha_i K\langle x, x_i\rangle - b) \tag{2}$$

Variations on SVM algorithms use different classification functions. No algorithmic changes are required from the usual kernel function K as a linear inner product other than the modification of the kernel evaluation, including a polynomial function of degree d, a RBF (Radial Basis Function) or a sigmoid function. We can get different support vector classification models.

PmSVM proposed by Wu [2] replaces the kernel function $K\langle x_i, x_j\rangle$ in (1) and (2) with the power mean kernel $M\langle x_i, x_j\rangle$ (x_i and $x_j \in R^n_+$), which is well-known as a general form of many additive kernels (e.g. χ^2 kernel, histogram intersection kernel or Hellinger's kernel).

$$M_p\langle x_i, x_j\rangle = \sum_{d=1}^{n} (x_{id}^p + x_{jd}^p)^{\frac{1}{p}} \tag{3}$$

where $p \in R$ is a constant.

PmSVM uses the coordinate descent method [26] for dealing with training tasks. Furthermore, the gradient computation step of the coordinate descent algorithm can be estimated approximately with the polynomial regression with a very low cost [2]. Therefore, PmSVM is very efficient in both training and testing tasks, compared to LIBLINEAR and other additive kernel SVMs.

4 Extensions of PmSVM to Large Number of Classes

Most SVM algorithms are only able to deal with a two-class problem. There are several extensions of a binary classification SVM solver to multi-class (k classes, $k \geq 3$) classification tasks. The state-of-the-art multi-class SVMs are categorized into two types of approaches. The first one is to consider the multi-class case in an optimization problem [27], [28]. The second one is to decompose multi-class into a series of binary SVMs, including one-versus-all [25], one-versus-one [29] and Decision Directed Acyclic Graph [30]. Recently, hierarchical methods for multi-class SVM [31], [32] start from the whole data set, hierarchically divide the data into two subsets until every subset consists of only one class.

In practice, one-versus-all, one-versus-one are the most popular methods due to their simplicity. Let us consider k classes ($k > 2$). The one-versus-all strategy builds k different classifiers where the i^{th} classifier separates the i^{th} class from the rest. The one-versus-one strategy constructs $k(k-1)/2$ classifiers, using all the binary pairwise combinations of the k classes. The class is then predicted with a majority vote.

When dealing with very large number of classes, e.g. hundreds of classes, the one-versus-one strategy is too expensive because it needs to train many thousands classifiers. Therefore, the one-versus-all strategy becomes popular in this case. PmSVM algorithm also uses the one-versus-all approach to train independently k binary classifiers. However, the current PmSVM takes very long time to classify very large number of classes.

Due to this problem, we propose two ways for speed-up learning tasks of PmSVM. The first one is to build the balanced bagging classifiers with sampling strategy. The second one is to parallelize the training task of all classifiers with multi-core computers.

4.1 Balanced Bagging PmSVM

In the one-versus-all approach, the learning task of PmSVM is to try to separate the i^{th} class (positive class) from the $k-1$ others classes (negative class). For very large number of classes, e.g. 1000 classes, this leads to the extreme imbalance between the positive class and the negative class. The problem is well-known as the class imbalance. As summarized by the review papers of [33], [34], [35] and the very comprehensive papers of [36], [37], solutions to the class imbalance problems were proposed both at the data and algorithmic level. At the data level, these algorithms change the class distribution, including over-sampling the minority class [38] or under-sampling the majority class [39], [40]. At the algorithmic level, the solution is to re-balance the error rate by weighting each type of error with the corresponding cost. Our balanced bagging PmSVM belongs to the first approach (forms of re-sampling). Furthermore, the class prior probabilities in this context are highly unequal (e.g. the distribution of the positive class is 0.1% in the 1000 classes classification problem), and over-sampling the minority class is very expensive. We propose the balanced bagging PmSVM using under-sampling the majority class (negative class).

For separating the i^{th} class (positive class) from the rest (negative class), the balanced bagging PmSVM trains T models as shown in algorithm 1.

Algorithm 1. Balanced bagging PmSVM

```
input :
        $D_p$ the training data of the positive class
        $D_n$ the training data of the negative class
        $T$ the number of base learners
output:
        $PmSVMmodel$

Learn:
for $k \leftarrow 1$ to $T$ do
    1. The subset $D'_n$ is created by sampling without replacement $|D'_n|$ negative
       datapoints from $D_n$ (with $|D'_n| = |D_p|$)
    2. Build a PmSVM model using the training set (including $D_p$ and $D'_n$)
end
combine $T$ models (averaging) into the aggregated $PmSVMmodel$
```

We remark that the margin can be seen as the minimum distance between two convex hulls, H_p of the positive class and H_n of the negative class (the farthest distance between the two classes). Under-sampling the negative class (D'_n) done by balanced bagging provides the reduced convex hull of H_n, called H'_n. And then, the minimum distance between H_p and H'_n is larger than H_p and H_n (full dataset). It is easier to achieve the largest margin than learning on the full dataset. Therefore, the training task of PmSVM is fast to converge to the solution. According to our experiments, by setting $T = \sqrt{\frac{|D_n|}{|D_p|}}$, the balanced bagging PmSVM achieves good results in very fast training speed.

4.2 Parallel PmSVM Training

Although PmSVM and balanced bagging PmSVM deal with very large dataset with high speed, they do not take into account the benefits of HPC, e.g. multi-core computers or grids. Furthermore, both PmSVM and balanced bagging PmSVM train independently k binary classifiers for k classes problems. This is a nice property for parallel learning. Our investigation aims at speed-up training tasks of multi-class PmSVM, balanced bagging PmSVM with multi-processor computers or grids. The idea is to learn k binary classifiers in parallel.

Algorithm 2. Hybrid MPI/OpenMP parallel PmSVM

```
input :
        $D$ the training dataset with $k$ classes
        $T$ the number of MPI processes
output:
        $PmSVMmodel$

Learn:
$MPI - PROC_1$
#pragma omp parallel for
for $i_1 \leftarrow 1$ to $k_1$ do                              /* class $i_1$ */
    Build a binary PmSVM model using the training set $D$ to separate the positive
    class $i_1$ from the rest.
end
:
$MPI - PROC_T$
#pragma omp parallel for
for $i_T \leftarrow 1$ to $k_T$ do                              /* class $i_T$ */
    Build a binary PmSVM model using the training set $D$ to separate the positive
    class $i_T$ from the rest.
end
```

The parallel programming is currently based on two major models, Message Passing Interface (MPI) [41] and Open Multiprocessing (OpenMP) [42]. MPI is a standardized and portable message-passing mechanism for distributed memory systems. MPI remains the dominant model (high performance, scalability, and portability) used in high-performance computing today. However, a MPI process loads the whole dataset ($\sim$ 25GB) into memory during learning tasks, making it intractable. The simplest development of parallel PmSVM algorithms is based on the shared memory multiprocessing programming model OpenMP. However OpenMP is not guaranteed to make the most efficient computing. Finally, we present a hybrid approach that combines the benefits from both OpenMP and MPI models. The parallel PmSVM algorithm is described in algorithm 2. The number of MPI processes depends on the memory capacity of high performance computing systems.

5 Experiments

In this section we compare the extended versions of PmSVM with the original implementation and LIBLINEAR in terms of training time and classification accuracy. Our experiments are run on machine Intel(R) Xeon(R), CPU X5650, 2.67GHz, 24 cores, and 144GB main memory.

We have implemented four parallel versions of PmSVM: 1) OpenMP version of PmSVM (omp-PmSVM), 2) balanced bagging version of omp-PmSVM (omp-iPmSVM), 3) hybrid MPI/OpenMP version of PmSVM (mpi-omp-PmSVM), and 4) balanced bagging version of mpi-omp-PmSVM (mpi-omp-iPmSVM).

PmSVM. We set the parameters of PmSVM as follow: $p = -1$ (equivalent to χ^2 kernel) and $C = 0.01$.

LIBLINEAR. This is the linear SVM from [4] with default parameters ($C = 1$).

iPmSVM. This is the balanced bagging PmSVM with the same SVM parameters as PmSVM.

5.1 Dataset

The parallel versions of PmSVM are designed for large scale datasets, so we have evaluated the performance of our approach on the three following datasets.

ImageNet 10. This dataset contains the 10 largest classes from ImageNet (24,807 images with data size 2.4GB). There are more than 2000 diversified images per class. In each class, we sample 90% images for training and 10% images for testing. First, we construct bag-of-words histogram of every image by using dense SIFT descriptor (extracting SIFT on a dense grid of locations at a fixed scale and orientation), and 5000 codewords. Then, we use feature mapping from [10] to get the high-dimensional image representation in 15,000 dimensions. This feature mapping has been proven to give a good image classification performance [10]. We end up with 2.6GB of training data.

ImageNet 100. This dataset contains the 100 largest classes from ImageNet (183,116 images with data size 23.6GB). In each class, we sample 50% images for training and 50% images for testing. We also construct bag-of-words histogram of every image by using dense SIFT descriptor and 5000 codewords. For feature mapping, we use the same method as we do with ImageNet 10. The final size of training data is 8GB.

ILSVRC 1000. This dataset contains 1000 classes from ImageNet with 1.2M images (126GB) for training, 50K images (5.3GB) for validation and 150K images (16GB) for testing. To compare with the results reported in [2], we use the same method to encode every image as a vector in 21,000 dimensions. We also take ≤ 900 images per class for training dataset. Therefore, the total number of training images is 887,816 and the size of training data is 12.5GB. All testing samples are used to test SVM models.

5.2 Training Time

We have only evaluated the training time of SVM classifiers excluding the time needed to load data from disk. As shown in Fig. 2 and 3, on small and medium datasets as ImageNet 10, ImageNet 100, our four parallel versions show a very good speed-up in training process, compared to the original implementation of PmSVM and LIBLINEAR (Table 1 and 2).

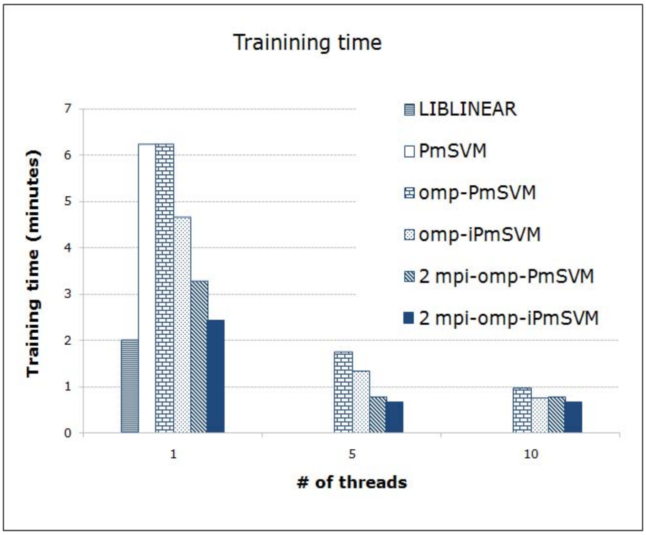


Fig. 2. SVMs training time with respect to the number of threads for ImageNet 10

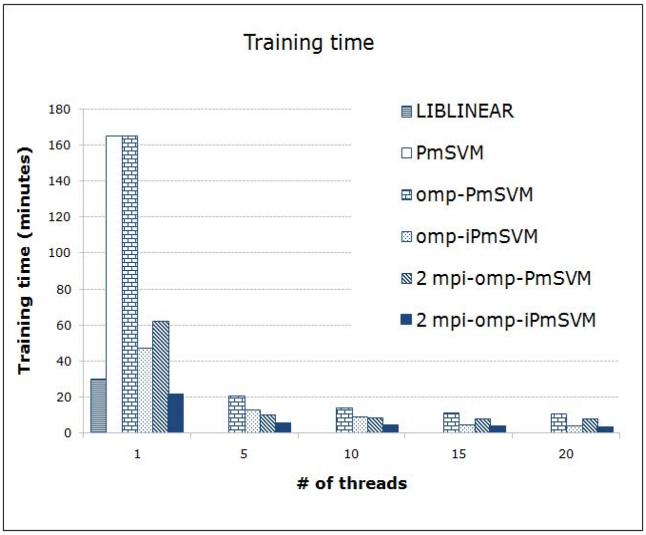


Fig. 3. SVMs training time with respect to the number of threads for ImageNet 100

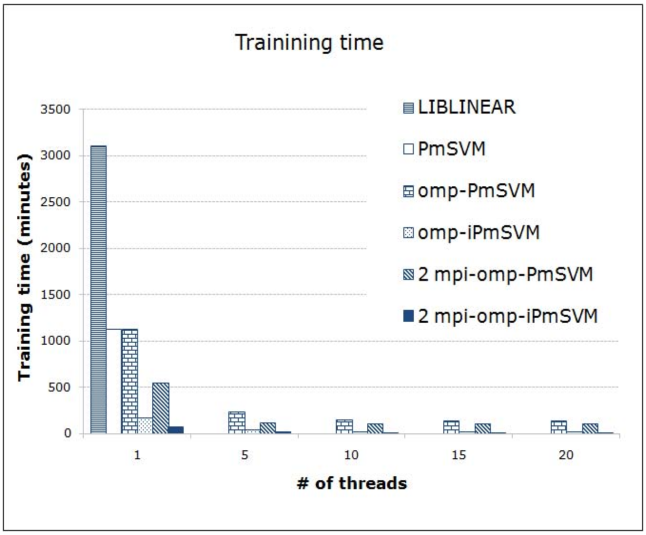


Fig. 4. SVMs training time with respect to the number of threads for ILSVRC 1000

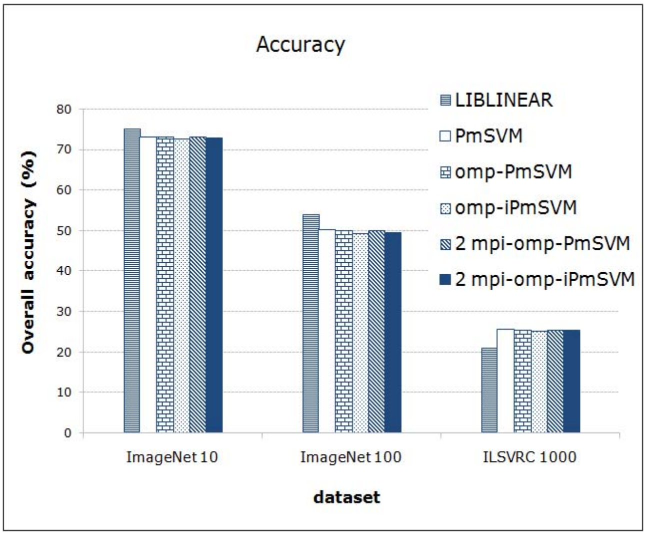


Fig. 5. Overall accuracy of SVM classifiers (%)

Table 1. SVMs training time (minutes) on ImageNet 10

# OpenMP threads	1	5	10
LIBLINEAR	2.02		
PmSVM	6.23		
omp-PmSVM	6.23	1.75	0.98
omp-iPmSVM	4.66	1.34	0.77
2 mpi-omp-PmSVM	3.29	0.78	0.76
2 mpi-omp-iPmSVM	2.44	0.67	**0.65**

Table 2. SVMs training time (minutes) on ImageNet 100

# OpenMP threads	1	5	10	15	20
LIBLINEAR	30.41				
PmSVM	165.45				
omp-PmSVM	165.45	21.12	14.52	11.67	10.92
omp-iPmSVM	47.67	12.97	9.09	5.05	4.44
2 mpi-omp-PmSVM	62.18	10.63	8.85	8.25	8.06
2 mpi-omp-iPmSVM	21.56	5.68	4.16	3.79	**3.58**

Table 3. SVMs training time (minutes) on ILSVRC 1000

# OpenMP threads	1	5	10	15	20
LIBLINEAR	3106.48				
PmSVM	1132.03				
omp-PmSVM	1132.03	231.00	152.87	140.72	135.63
omp-iPmSVM	173.26	39.55	23.63	19.43	18.12
2 mpi-omp-PmSVM	550.04	119.17	102.81	103.62	103.12
2 mpi-omp-iPmSVM	72.07	16.69	**13.08**	13.37	13.25

Table 4. SVMs overall classification accuracy (%)

dataset	ImageNet 10	ImageNet 100	ILSVRC 1000
LIBLINEAR	75.09	54.07	21.11
PmSVM	73.16	50.17	25.64
omp-PmSVM	73.16	50.17	25.64
omp-iPmSVM	72.79	49.42	25.35
2 mpi-omp-PmSVM	73.16	50.17	25.64
2 mpi-omp-iPmSVM	72.79	49.42	25.35

ILSVRC 1000. Our implementations achieve a significant speed-up in training process when performing on large dataset ILSVRC 1000.

Balanced Bagging PmSVM. As shown in Fig. 4, the balanced bagging version of PmSVM (omp-iPmSVM running with 1 thread) has a very fast convergence speed in training process, it is more than 6 times faster than the original implementation of PmSVM (Table 3).

OpenMP PmSVM. On a multi-core machine, OpenMP version of PmSVM (omp-PmSVM) achieves a significant speed-up in training process with 20 OpenMP threads. As shown in Fig. 4, our implementation is 8 times faster than the original PmSVM and 23 times faster than LIBLINEAR (Table 3). Due to the restriction of our computer (24 cores), we set the maximum number of OpenMP threads to 20. We can set more than 20 OpenMP threads, but according to our observation there is very few significant speed-up in training process because there is no more available core.

OpenMP and Balanced Bagging PmSVM. With balanced bagging algorithm applied to OpenMP version of PmSVM (omp-iPmSVM), we significantly speed-up the training process on this training data. For instance, with the number of OpenMP threads set to 20, omp-iPmSVM is 62 times faster than the original PmSVM and 171 times faster than LIBLINEAR.

MPI/OpenMP PmSVM. As show in Fig. 4, our hybrid MPI/OpenMP version of PmSVM (mpi-omp-PmSVM) achieves a significant speed-up in training process with 2 MPI processes and 10 OpenMP threads. Our implementation is 11 times faster than the original PmSVM and 30 times faster than LIBLINEAR (Table 3). Due to the large size of this training data, each MPI process of PmSVM needs to use $\sim$ 25GB main memory to train classifiers. With the memory restrictions of our computer, we can only evaluate mpi-omp-PmSVM by setting the number of MPI processes to 2. In this case, we vary the number of OpenMP threads running in each MPI process. Again, with 24 cores of our computer we set the maximum number of OpenMP threads to 10.

MPI/OpenMP and Balanced Bagging PmSVM. The most significant parallelization performance of PmSVM we achieve is the combination of MPI/OpenMP and balanced bagging PmSVM (mpi-omp-iPmSVM). As shown in Fig. 4, our implementation achieves a significant performance in training process with 2 MPI processes and 10 OpenMP threads. It is 90 times faster than the original PmSVM and 240 times faster than LIBLINEAR. On ILSVRC 1000, we need only 13 minutes to finish training 1000 binary classifiers, compared to the original PmSVM ($\sim$ 19 hours) and LIBLINEAR ($\sim$ 2 days and 4 hours), as shown in Table 3. This result confirms that our approach has a great ability to scaleup to full ImageNet dataset with more than 21,000 classes.

5.3 Classification Accuracy

As shown in Fig. 5, on the small datasets like ImageNet 10 and ImageNet 100, LIBLINEAR outperforms PmSVM and iPmSVM in terms of classification accuracy.

However, when we perform classification on the dataset with very large number of classes like ILSVRC 1000, the picture of accuracy performance is quite different.

PmSVM and iPmSVM achieve better results than LIBLINEAR (from +4.24% to +4.53%, ie. a relative increase of 20.1%).

Note that iPmSVM runs much faster than PmSVM without (or very few) compromising classification accuracy (Table 4).

6 Conclusion and Future Work

We have developed the extended versions of PmSVM to efficiently deal with large scale datasets with very large number of classes like ImageNet. To speed-up the training process of the binary classifiers, we have extended PmSVM in two ways. The first one is to build the balanced bagging classifiers with under-sampling strategy. Our algorithm avoids training on full training data, so the training process of PmSVM rapidly converges to the optimal solution. The second one is to parallelize the training process of all classifiers with multi-core computers. We have developed the parallel versions of PmSVM based on HPC models (OpenMP, MPI, and hybrid MPI/OpenMP). In each parallel version of PmSVM we apply our balanced bagging algorithm to obtain the best performance in the training process.

We have evaluated the performance in terms of training time on the large scale dataset like ImageNet. On ILSVRC 1000, our implementation is 90 times faster than the original implementation of PmSVM and 240 times faster than LIBLINEAR. To obtain this performance we set the number of threads to 20 on our computer. Therefore, with our approach we can get higher performances by using more resources (CPU cores, computer, etc.). Furthermore, with the balanced bagging approach we significantly speed-up the training process of the classifiers without (or very few) compromising the overall classification accuracy. We need only 13 minutes to train 1000 binary classifiers on ILSVRC 1000. Obviously, this is a roadmap towards very large scale visual classification. However, when the training data is larger, PmSVM requires a large amount of main memory. In the near future, we may study the approach that avoids loading the whole training data to main memory as in [43]. Another possibility could be to compress the training data and handle the compressed data on the fly, as in [21].

Acknowledgements. This work was partially funded by Region Bretagne (France).

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: Imagenet: A large-scale hierarchical image database. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
2. Wu, J.: Power mean svm for large scale visual classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2344–2351 (2012)
3. Berg, A., Deng, J., Li, F.F.: Large scale visual recognition challenge 2010. Technical report (2010)
4. Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear svm. In: International Conference on Machine Learning, pp. 408–415 (2008)

5. Li, F.F., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding 106(1), 59–70 (2007)
6. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. Technical Report CNS-TR-2007-001, California Institute of Technology (2007)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88(2), 303–338 (2010)
8. Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 71–84. Springer, Heidelberg (2010)
9. Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L., Huang, T.S.: Large-scale image classification: Fast feature extraction and svm training. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1689–1696 (2011)
10. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(3), 480–492 (2012)
11. Wu, J.: A fast dual method for HIK SVM learning. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 552–565. Springer, Heidelberg (2010)
12. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: In Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
13. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2169–2178 (2006)
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886–893. IEEE Computer Society (2005)
15. Griffin, G., Perona, D.: Learning and using taxonomies for fast visual categorization. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society (2008)
16. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: IEEE 12th International Conference on Computer Vision, pp. 606–613. IEEE (2009)
17. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: Advances in Neural Information Processing Systems, pp. 522–530 (2009)
18. Wang, C., Yan, S., Zhang, H.J.: Large scale natural image classification by sparsity exploration. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3709–3712. IEEE (2009)
19. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections. In: IEEE 12th International Conference on Computer Vision, pp. 1957–1964. IEEE (2009)
20. Perronnin, F., Sánchez, J., Liu, Y.: Large-scale image categorization with explicit data embedding. In: CVPR, pp. 2297–2304 (2010)
21. Sánchez, J., Perronnin, F.: High-dimensional signature compression for large-scale image classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1665–1672 (2011)
22. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Advances in Neural Information Processing Systems, pp. 2223–2231 (2009)
23. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: European Conference on Computer Vision, pp. 141–154 (2010)
24. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(1), 117–128 (2011)

25. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
26. Yuan, G.X., Ho, C.H., Lin, C.J.: Recent advances of large-scale linear classification. Proceedings of the IEEE 100(9), 2584–2603 (2012)
27. Weston, J., Watkins, C.: Support vector machines for multi-class pattern recognition. In: Proceedings of the Seventh European Symposium on Artificial Neural Networks, pp. 219–224 (1999)
28. Guermeur, Y.: Svm multiclasses, théorie et applications (2007)
29. Krebel, U.: Pairwise classification and support vector machines. Advances in Kernel Methods: Support Vector Learning, 255–268 (1999)
30. Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin dags for multiclass classification. Advances in Neural Information Processing Systems 12, 547–553 (2000)
31. Vural, V., Dy, J.: A hierarchical method for multi-class support vector machines. In: Proceedings of the Twenty-frst International Conference on Machine Learning, pp. 831–838 (2004)
32. Benabdeslem, K., Bennani, Y.: Dendogram-based svm for multi-class classification. Journal of Computing and Information Technology 14(4), 283–289 (2006)
33. Japkowicz, N. (ed.): AAAI'Workshop on Learning from Imbalanced Data Sets. Number WS-00-05 in AAAI Tech. report (2000)
34. Weiss, G.M., Provost, F.: Learning when training data are costly: The effect of class distribution on tree induction. Journal of Artificial Intelligence Research 19, 315–354 (2003)
35. Visa, S., Ralescu, A.: Issues in mining imbalanced data sets - A review paper. In: Midwest Artificial Intelligence and Cognitive Science Conf., Dayton, USA, pp. 67–73 (2005)
36. Lenca, P., Lallich, S., Do, T.-N., Pham, N.-K.: A comparison of different off-centered entropies to deal with class imbalance for decision trees. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 634–643. Springer, Heidelberg (2008)
37. Pham, N.K., Do, T.N., Lenca, P., Lallich, S.: Using local node information in decision trees: coupling a local decision rule with an off-centered. In: International Conference on Data Mining, pp. 117–123. CSREA Press, Las Vegas (2008)
38. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
39. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B 39(2), 539–550 (2009)
40. Ricamato, M.T., Marrocco, C., Tortorella, F.: Mcs-based balancing techniques for skewed classes: An empirical comparison. In: ICPR, pp. 1–4 (2008)
41. MPI-Forum.: Mpi: A message-passing interface standard
42. OpenMP Architecture Review Board: OpenMP application program interface version 3.0 (2008)
43. Do, T.N., Nguyen, V.H., Poulet, F.: Gpu-based parallel svm algorithm. Journal of Frontiers of Computer Science and Technology 3(4), 368–377 (2009)

Area under the Distance Threshold Curve as an Evaluation Measure for Probabilistic Classifiers

Sydney Williams[1], Michael Harris[2], Jacob Furst[3], and Daniela Raicu[3]

[1] Illinois Institute of Technology, Biomedical Engineering, Chicago, Illinois
sydney@hawk.iit.edu
[2] Sonoma State University, Computer Science, Rohnert Park, California
harrismi@seawolf.sonoma.edu
[3] DePaul University, Computing and Digital Media, Chicago, Illinois
{jfurst,draicu}@cdm.depaul.edu

Abstract. Evaluation for probabilistic multiclass systems has predominately been done by converting data into binary classes. While effective in quantifying the classifier performance, binary evaluation causes a loss in ability to distinguish between individual classes. We report that the evaluation of multiclass probabilistic classifiers can be quantified by using the area under the distance threshold curve for multiple distance metrics. We construct our classifiers for evaluation with data from the National Cancer Institute (NCI) Lung Image Database Consortium (LIDC) for the semantic characteristic of malignancy. We conclude that the area under the distance threshold curve can provide a measure of the classifier performance when the classifier has more than two classes and probabilistic predictions.

Keywords: Machine learning, medical informatics, probabilistic classifier, ROC curve, K-Nearest neighbor.

1 Introduction

In the advent of classifiers as key diagnostic tools within medical imaging, rigorous exploration of evaluation measures has begun in order to assess their performance. In particular, accuracy and the receiver operator characteristic curve (ROC) have been well-accepted within the machine learning community [1]. In more recent analyses, cost curves have also been reviewed because of the ability to assess different severities of misclassifications [2]. Nevertheless, previous discussions have been limited to binary labeling systems, where classifiers identify the probability of either a positive or negative label only.

While a binary label is often the preferred output of a computer aided diagnosis system (e.g., telling a doctor “healthy” or “sick”), sometimes an output that has more details is required. Multiple label classifiers can produce an output that gives a range of information to a physician, such as a rating on a 1-5 scale, or the structure of the tissue sample as a type (fat, fluid, soft tissue, air, etc). In contrast to ordinal regression where the relationship between labels 1-5 is significant, multiclass classifiers focus on class distinction, regardless of the label given. These types of classifiers are

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 644–657, 2013.

also useful outside the medical field for purposes such as image labeling, identifying genre's in music, recommender systems, and any other problem where the desired output is to identify which of several types the input belongs to.

By using a multiclass probabilistic input, a classifier will learn using a range of information. For example, in this paper we see that multiple radiologists annotate characteristics lung nodules uniquely, and a multiclass input accounts for these differences in training. Furthermore, by combining probability with a multiclass output, one can assess how similar an object is to more than one class. For example, in showing the probabilities for the most likely cases a doctor would have more information than if simply provided a single class, further increasing the data available when compared to binary classification. Because classifiers are not perfect, having an effective way to show the most likely possibilities for the object in question is useful: a medical expert may agree with the classifiers second-most probable choice instead of the first choice.

In order to evaluate a probabilistic and multi-class classifier, a new evaluation measure has been introduced by Zinovev, Furst, and Raicu in 2011, the Area Under the Distance Threshold Curve (AUCdt) using relative differences measured by the Jeffrey Divergence [3]. Expanding upon the various evaluation functions described by Amor, Benferhat, and Elouedi in 2006 [4] the distance threshold curve helps visualize and quantify multiclass probabilistic performance. In this paper, we will expand upon the previously used AUCdt by incorporating additional distance measures as well as attempting to weight the difference across multiple class labels in regards to their relative "costs" for our application. The evaluation difference metrics that we are investigating are the City Block Difference (CB), the Jeffrey Divergence (JD), and Earth Movers Distance (EMD). Our goal is to show that these new quantities will help evaluate multiclass probabilistic classifiers just as has been done previously in the binary class case with area under the ROC curve (AUC).

2 Background

There are several challenges in evaluating probabilistic multiclass classifiers. One of these is that there is not always a correct answer, as a sample may belong to multiple classes, or there is disagreement between experts as to which class the sample belongs to. Many of the evaluation methods that are used for binary classifiers simply do not work very well when applied to multi-class output, or do not provide an accurate picture of how close the predicted output is to the expected output.

Accuracy is simply defined as the correct number of classifications over the total number of instances, or also 1-ERROR [1]:

$$Accuracy = \frac{TP+TN}{CP+CN} \tag{1}$$

where TP and TN indicate the true (correctly-classified) positives and negatives, and CP and CN , the total number of positives and negatives. While this measure is readily-available with most classifiers, its simplistic nature cannot distinguish between false positives or negatives nor generate a visual graph to define classifier performance over varying conditions, such as a threshold. Consequentially, it has been suggested that the ROC curve will allow for a more percipient yet still-consistent evaluation [5].

Discussed by Provost and Fawcett in 1997 for machine learning, the ROC curve has been a major evaluation tool for classifiers [6]. The ROC curve is constructed by plotting sensitivity vs. 1-specificity as defined using terminology from the confusion matrix seen in Table 1:

Table 1. Confusion matrix for constructing ROC curve, where labeled values come from known labels and predicted values from the classifier

Labeled **Predicted**	**+**	**–**
+	True Positive (TP)	False Positive (FP
–	False Negative (FN)	True Negative (TN)

With sensitivity being equivalent to the true positive rate:

$$Sensitivity = TPR = \frac{TP}{TP + FN} \tag{2}$$

and specificity, 1 minus the false positive rate [7]:

$$Specificity = 1 - FPR = \frac{TN}{TN+FP}. \tag{3}$$

A range of false positive probability thresholds is then created from 0 to 1. For each new threshold, if the probability of a positive label is equal or above the threshold, it is added to the value of the true positive rate and plotted as the ROC curve. Figure 1 shows an example of an ROC curve generated using artificial data with varying percentages of perfect agreement and random labels. Along with the benefit of providing an important visualization to a classifier's performance, the ROC curve also can easily be compared to "random" classifications (diagonal line with slope of 1 through curve) and is not susceptible to the bias of skewed distributions as is accuracy [8].

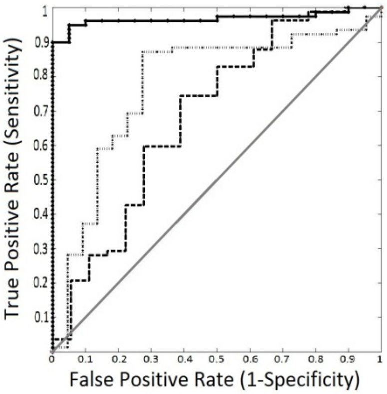


Fig. 1. Example of ROC curve— solid: artificial data with 90% agreement between predicted and labeled classes, dotted: data with 50% agreement, dashed: data with 10% agreement, diagonal: completely random performance (no classification)

Due to the benefits noted by the ROC curve, the Area Under the ROC Curve was explored by Bradley in 1997. It is calculated by taking the area enclosed beneath the ROC plot which has a range of 0 to 1. An area of 0 indicates no classification, 0.5 random classification, and 1, perfect classification [1]. The more similarity or agreement between predicted and labeled classes, the more area enclosed by the AUC. The AUC provides a quantifiable measure in order to differentiate and rank the effectiveness of different classifiers [8].

Despite the strength that ROC and AUC have given to the machine learning community, there are still several cases in which they fail as an evaluation measure. One example is cost, as explained by Drummond and Holte in 2006. The ROC curve does not take into effect the relative "costs" of misclassifications (i.e., the difference between a false positive and a false negative for the application). A cost curve evaluates the normalized cost of misclassification for a particular classifier using a varying range of percent positive example. Positive slopes on the cost curve graph are equivalent to when false positive rate is less than the false negative rate, and negative slopes, when false positive rate is greater. Depending on the application and whether or not false positives or false negatives "cost" more, one can assess the effect of misclassifications for a given classifier. Likewise, the convex hull beneath the cost curve can be made minimal to minimize cost [2]. In recent research, the Brier Curve has been of particular interest as a cost curve that works explicitly with probabilistic labels and has an area underneath, the Brier Score, analogous to AUC [9]. Despite the possibility of incorporating cost curves with ROC to further develop understanding of a classifiers performance, another key issue that arises is that these measures are only limited to a binary class system [2]. Furthermore, a previous expansion of the ROC to multiclass systems was done by Hand and Till in 2001 [10]. Nevertheless this was done in pairwise comparison, with only one class compared to all others.

3 Methodology

3.1 Probabilistic Distributions

Objects classified by a group of people may end up with some disagreement in the class or ranking assigned. This is most pronounced when the experts don't have access to what the others decided, as for example, getting a second opinion from another doctor, or a panel of Olympic judges each providing a different score for a gymnast, based on their own internal scoring metric. Because each opinion is valid and useful, even though different, by assigning probabilities based on the number of experts in agreement, we can produce a probability distribution showing the most likely true case. The goal of a classifier is then to attempt to match this varied response, so as to simulate a panel of experts, and to show multiple possibilities in addition to the highest one. An example of two distinct, multiclass probabilistic distributions can be seen in Figure 2 below. The distance threshold curve that we are proposing in this paper will allow us to evaluate and quantify the differences between such distributions.

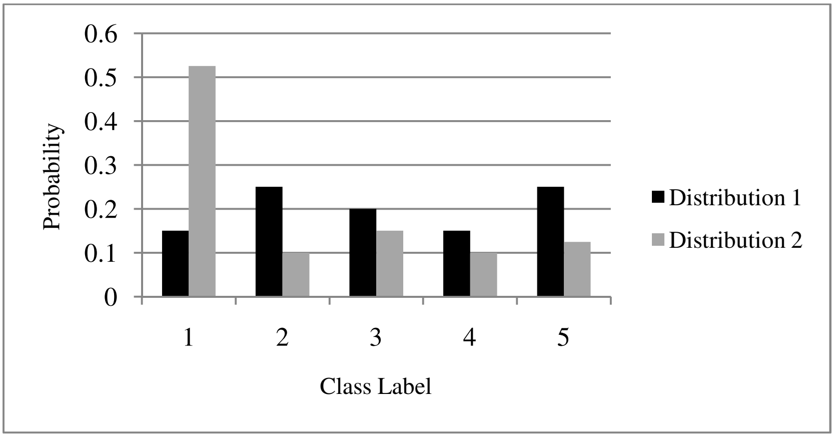


Fig. 2. Example of two distinct probabilistic distributions

3.2 Data

The Lung Image Database Consortium (LIDC) dataset consists of lung nodules rated by up to four radiologists for a variety of semantic characteristics; 64 image features were extracted plus the z-position from the nodule slice images [3]. We then used the data for each image slice, and grouped them by nodule. We filtered by selecting nodules that had been labeled by four radiologists, and then selected two slices, that were located at roughly 33% and 67% through the nodule based on the Z-position value. The slice selection was done programmatically, so as to be able to select multiple images evenly distributed throughout the nodule. We tried different values for the number of slices, and found that two provided a good mix of accuracy and speed. By concatenating the features from the second selected slice to the first, we doubled the number of features from 65 to 130. Two additional features were added based on the entire set of slices for the nodule, Height, and Volume, where height is the distance from the smallest Z-value to the largest, and Volume is the area of each slice multiplied by the thickness, and then summed across all slices. Using this, we were able to produce a data set of 830 out of 2660 total nodules, each with 132 features and characteristic labels from all four radiologists. For our paper, we focused on the semantic characteristic of malignancy, which is scaled on an ordinal range of 1-5, with 1 being the “most-likely benign” and 5 being the “most-likely malignant”.

3.3 Classifiers

This data set was then run through a K-Nearest Neighbor classifier, using different values of K. We used a 90%-10% split, where every 10th nodule was set aside for testing purposes, and the other nine were used for training. We varied our number of nearest neighbors from K=2 to K=9 and found that using 5 nearest neighbors was optimum by using the City Block Distance to compare the output results to the radiologist labels and find the difference. From there, we proceeded to use more sophisticated analysis methods to study the results. The classifier took in multiple labels for each nodule, and produced multiple probabilities as the output. An example is found in Table 2:

Table 2. Sample distribution of radiologist and classifier-predicted probabilistic distributions. In this case, the nodule had been labeled "2" by one radiologist, "3" by two radiologists, and "4" by one radiologist.

Label	**1**	**2**	**3**	**4**	**5**
Radiologists Values (A_i)	0%	25%	50%	25%	0%
Predicted Values (B_i)	0%	21.4%	50.0%	25.0%	3.6%

The K-nearest neighbor algorithm sums the Radiologists' Values, $A_{i,}$ of the K-nearest neighbors (found using the K smallest values of the L_1 norm), and divides by K, to produce the Predicted Values, B_i. The results of the training set are artificially higher than expected due to the fact that for a nodule in the training set, there is a guarantee that one of the nearest neighbors will be the exact nodule being examined, so the results from the testing set are much more useful for determining how well the classifier performed.

In a binary K-NN classifier, the last step is to select the class that is the most frequent, and declare the test case to be of that type. In our case, we take the average probabilities for each type, thus resulting in a distribution similar to B_i in Table 2. A case for probabilistic inputs for multiclass K-NN Classifiers has previously been made in 2009 by Jain and Kapoor [11].

3.4 Evaluation Measures

Accuracy. In order to determine the accuracy of each slice prediction, each probabilistic distribution of values needed to be converted into a single, discrete label. To do so, each distribution by radiologists and predictions were viewed separately and the largest probability was found. The class label index at this case (i.e. 1,2,3,4 or 5), was then assigned as the single label for the slice. In the case that there were two or more probabilities that were equally high, the higher index value was chosen as the label. This was done to err on the side of caution for our malignancy classifier, where a label of "5" is considered the "most-likely malignant".

ROC and AUC. To perform the ROC analysis of the data, a separate conversion of the probabilistic distributions was required from a multiclass to binary array. In order to do so, the five labels had to be isolated into positive and negative classes. Labels 1 and 2 were assigned to the negative class and labels 4 and 5 to the positive. Label 3 is defined as "unknown" malignancy by the LIDC, so was subjected to three conditions:

3 in the positive class (AUC 3+), 3 in the negative class (AUC 3-), or 3 removed from classification evaluation (AUC 3out). It is important to acknowledge that it is conservative to place 3 in the positive class label due to clinical relevancy of the data: in terms of malignancy, it is more appropriate to overestimate than underestimate.

Next positive and negative class probabilities were summed into a binary array of distributions for both radiologists' and predicted label. In both cases, the larger of the two probabilities received the label for that particular slice. The positive class was selected in the case that the two probabilities were 0.5. From there, a simple ROC curve was constructed using the radiologists' labels and the positive probabilities of the predicted labels with a threshold range of 0 to 1 in 0.05 increments. The area under the curve was then extracted using a trapezoidal approximation of the integral.

AUC$_{dt}$. The Area Under the Distance Threshold Curve was constructed using a variety of distance metrics, including the City Block, Jeffrey Divergence, and Earth Mover's Distance. For any distance measure (d), the AUC$_{dt}$ for instances (S) is given by [3]:

$$AUC_{dt} = \int_0^1 \frac{\sum_{j=1}^{|S|} d(j) \leq x}{|S|} \, dx \tag{4}$$

City Block Distance. The City Block Distance, a member of the Minkowski Distances for p=1 [12]:

$$d_{CB}(A,B) = \sum_{i=1}^{n} (|A_i - B_i|^P)^{\frac{1}{P}} \tag{5}$$

was calculated between each radiologist's probability (A$_i$) and predicted probability (B$_i$) for a given label (i=1:5), summed across all labels, and then stored into a vector (d$_{CB}$) of distances for all nodules.

Jeffrey Divergence. The Jeffrey Divergence was calculated and stored into a similar vector of distances (d$_{JD}$) with the following algorithm [12]:

$$d_{JD}(A,B) = \sum_{i=1}^{n} [A_i \log\left(\frac{A_i}{\frac{A_i+B_i}{2}}\right) + B_i \log\left(\frac{B_i}{\frac{A_i+B_i}{2}}\right)] \tag{6}$$

Earth Mover's Distance. Earth mover's distance is a measure of the amount of change required to move between two different sets of data, assuming that there is a value to each location as well as the value stored in that location [3]. With the previous distance metrics, there is no difference between a nodule that should be a 5 being incorrectly rated as a 1, and being incorrectly as a 4. In many cases, where the classes are part of a scale, such as a five star rating for restaurants, or the severity of a tumor, there is a large difference between those two types of errors. This method takes that into account, so that a rating of 1 is seen as much worse than a rating of 4, in the case that the real value is 5.

$$EMD_0 = 0 \tag{7}$$

$$EMD_{i+1} = (B_i + EMD_i) - A_i \tag{8}$$

$$d_{EMD}(A,B) = \sum_{i=0}^{N+1} |EMD_i| \tag{9}$$

Weighted Distance. In terms of the clinical application of our classifier, the relative cost between labels was explored through the idea of "weighting" each label. It has already been mentioned that as the label for a nodule's malignancy increases, so does the clinical severity of that label. Therefore, weighted distances for the City Block and Jeffrey Divergence distance metrics were introduced as well. The weighting used was to multiply the difference for each rating by the value of the rating. In other words, for formulas 5 and 6, each probability (A_i and B_i) was multiplied by the label (*i*). This algorithm puts additional cost weight on the higher malignancy labels, with each label increasing in severity by a factor of one. Nevertheless, this is only a modest weighting assumption as such levels of clinical severity are difficult to quantify.

Construction of the Distance Threshold Curve. After all distance distributions vectors were created, their values were normalized between 0 to 1 for that particular distance metric. For each distance measure, a cumulative histogram was created of the varying distance values for threshold values of 0 to 1 with 0.05 increments and reported the percent frequency of instances for each threshold value (Figure 3). After the curve was plotted, the area beneath the curve was calculated using the trapezoidal integral approximation.

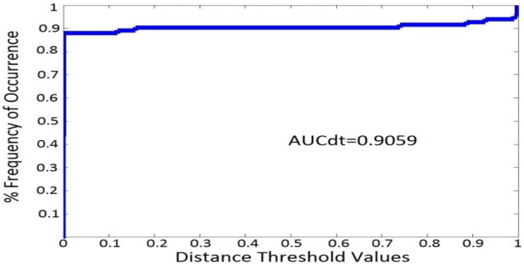


Fig. 3. Example of distance threshold curve constructed with data having 90% agreement. The AUC_{dt} is 0.9059.

In the ideal case of perfect classification the two distributions would be identical and have no difference, producing an AUC_{dt} with 100% frequency of cases for all distance thresholds and an area of 1, similar to the perfect classification instance of ROC curve for the binary case. Similarly, the worst classification gives an area of 0.

3.5 Random Classification of Evaluation Measures

One of the strengths in using accuracy and AUC for ROC evaluation measures is that they have known theoretical behaviors for random classification—that is to say, when a classifier arbitrarily assigns information. In our 5 label multiclass case, accuracy should be roughly 20% for random performance. Likewise, the binary positive/negative case induced with traditional ROC produces a diagonal line of slope 1 and has an AUC of 0.5 with random classification. However, in the new AUC$_{dt}$ for the multiclass probabilistic case, the random performance was unknown and unique for the various distance metrics.

In order to quantify AUC$_{dt}$ for the arbitrary case, random data was created to test this new measure as well as AUC and accuracy. First, we generated a set of 5000 random probability distributions for Labels 1-5 with a total sum across each distribution of 1. Then we simulated random radiologists for 5000 distributions. The probabilities available in the radiologists' distributions were limited to 0, 0.25, 0.5, 0.75, and 1 (as they would when given deterministic labels). The values of accuracy, AUC, and AUC$_{dt}$ mean and standard deviation of the 50 random radiologist sets with the random predictions are show in Table 3.

Table 3. Mean and standard deviation of evaluation measures with sets of 4 random radiologists and random predictions for 5000 values

Evaluation Measure	Mean	Standard Deviation
Accuracy	0.1999	0.0059
AUC (3+)	0.5080	0.0063
AUC (3-)	0.5055	0.0055
AUC (3out)	0.5054	0.0049
AUC$_{dt}$ CB	0.5391	0.0023
AUC$_{dt}$ JD	0.6891	0.0022
AUC$_{dt}$ EMD	0.7948	0.0015
AUC$_{dt}$ Wtd. CB	0.8374	0.0019
AUC$_{dt}$ Wtd. JD	0.6891	0.0022

It is important to notice that the values for AUC$_{dt}$ with random performance are not only different for a given distance metric (CB, JD, EMD, Wtd. CB, or Wtd. JD), they are all higher than 0.5 in the binary case with AUC for ROC. These values are important to have in evaluating with AUC$_{dt}$ in order to ensure that the classifier is performing better than random in the probabilistic multiclass case.

Next, in order to better understand how to quantify AUC$_{dt}$, the random classification data was iteratively changed to have varying amounts of agreement between the predictions and the actual radiologists' labels. As there were 5000 random predictions labels and only 747 actual nodules in the training set that were labeled, 4 samples of the 5000 random predictions were taken for each agreement level and the average values were reported. The agreement was increased by 10% from 0 to 90% and then also calculated at 95% and 99%. The evaluation metric values are shown in Table 4.

The information provided regarding random performance in Table 4 allows a certain value of AUC$_{dt}$ to be assessed in terms of similarity between the classified predictions and the actual label values. It is a baseline for which our classifier's performance can be assessed. For example, if our classifier returns an AUC$_{dt}$ value of 0.93 using Jeffrey Divergence, we know that our classifier would be classifying with between 80% and 90% agreement to the radiologists' labels.

Table 4. Performance of AUC_{dt} for all distance metrics comparing actual radiologists' label distributions and random predictions. The random prediction distributions were incrementally changed to have various percentages of agreement with the actual radiologists' labels.

% Agreement	AUC_{dt} CB	AUC_{dt} JD	AUC_{dt} EMD	AUC_{dt} Wtd. CB	AUC_{dt} Wtd. JD
0	0.5225	0.6725	0.7789	0.8220	0.6759
10	0.4769	0.6251	0.7530	0.8130	0.6290
20	0.5484	0.6750	0.7856	0.8410	0.6794
30	0.5805	0.6974	0.7967	0.8443	0.7012
40	0.6336	0.7379	0.8256	0.8683	0.7422
50	0.7350	0.8111	0.8745	0.9061	0.8139
60	0.7563	0.8256	0.8843	0.9087	0.8270
70	0.8269	0.8737	0.9172	0.9383	0.8783
80	0.8684	0.9031	0.9357	0.9497	0.9035
90	0.9433	0.9592	0.9713	0.9771	0.9601
95	0.9699	0.9793	0.9866	0.9909	0.9788
99	0.9946	0.9957	0.9976	0.9984	0.9962

4 Results

The classifiers developed for the semantic characteristic of malignancy were initially evaluated with binary evaluation metrics: accuracy and AUC (3 positive). The same classifier was also evaluated using the newly-introduced metrics for the area under the distance threshold curve: City Block Difference (simple and weighted), Jeffrey Divergence (simple and weighted), and Earth Mover's Distance. For all evaluation measures, a value as close to 1 is desirable. The results of the binary and multiclass measures for the best K-Nearest Neighbor classifier (K=5) are show in Table 5. Figures 4-5 provide the visual representation of all binary and multiclass evaluation metrics.

Table 5. Accuracy, AUC, and Area under the distance threshold curve (AUCdt) for three distance metrics: City Block Distance, Jeffrey Divergence, and Earth Mover's Distance, unweighted and weighted; 5NN Classifier of Malignancy.

Classifier	Accuracy	AUC (3+)	AUC_{dt} CB	AUC_{dt} JD	AUC_{dt} EMD	AUC_{dt} Wtd. CB	AUC_{dt} Wtd. JD
5-NN Training	0.6760	0.9438	0.7321	0.8620	0.8974	0.9187	0.8625
5-NN Testing	0.5542	0.9286	0.6753	0.8141	0.8694	0.8935	0.8144

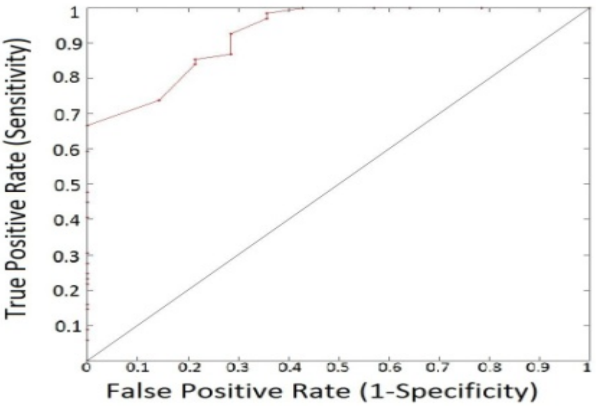


Fig. 4. ROC Curve for 5NN Test Set

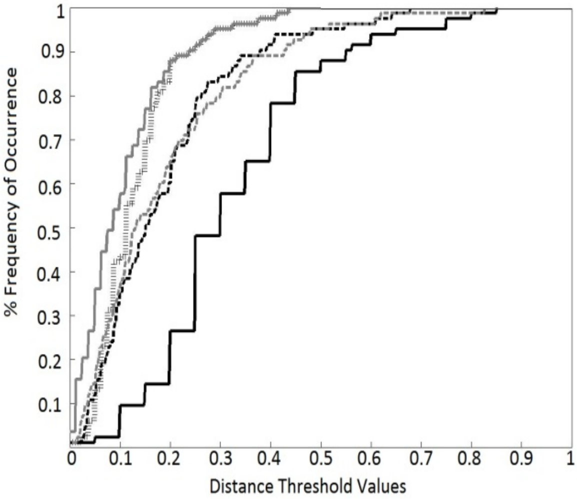


Fig. 5. Distance threshold curve for 5NN Test set—black solid: CBD, black thick dash: JD, black thin dash: EMD, grey solid: weighted CBD, and grey thick dash: weighted JD

5 Discussion

From Table 5 find that the accuracy achieved by the 5NN classifier was 67.60% for training and 55.42% for testing sets. In comparing to the random performance of accuracy in Table 3 (~20%) This low accuracy can namely be attributed to the conversion of our 5-class probabilistic distributions to a single, deterministic label. These results attest that accuracy is not an appropriate measure for evaluating a multiclass case.

The results for the area under the ROC curve in Table 5 show clearly that the 5-Nearest Neighbor classifier is performing well for the binary case: for all definitions of label “3”, the AUC has an area above 0.9. However, these values of AUC are different which confirms that AUC is dependent on how the two classes are defined. While our classifier can correctly distinguishing between two distinct classes, we still don’t receive information about how well the classifier discriminates single labels.

In regards to our various distance measures for the areas under the distance threshold curve found in Table 5, we find that all of our measures exceed their random performance values from Table 3. Furthermore we note that the weighted JD value of the AUCdt only increased by 0.0003 from its normal JD value. Jeffrey Divergence likely had very little change because the algorithm that calculates the distance is created using a logarithm, where single digit degrees contribute to fewer changes. This suggests that weighting JD is probably not necessary for future applications.

We also can use the values obtained in Table 4 to quantify at about which percentage level of agreement our classifier predicted compared to radiologists’ labels. This is done for the different distances used in AUCdt testing sets as follows: for CB, 40-50%, for JD, 50-60%, for EMD, 40-50%, for Wtd. Cb, 40-50%, and for Wtd. JD 50-60%. It is important to note that the range of agreement for all distance measures is around 40-60%. This allows us to argue that our 5-NN classifier is able to match the radiologists’ distributions at about 40-60% similarity. Furthermore, the consistency within the AUCdt validates it as a viable measure for the probabilistic, multiclass case.

6 Conclusion

Throughout this paper we have discussed the use and meaning behind traditional, binary measures for evaluation of classifier performance. Our results using a K-Nearest Neighbor classifier with K=5 for malignancy from lung nodule image data from the LIDC database confirm that AUC is a more-effective evaluation tool compared to accuracy. We have also emphasized the need for an evaluation metric for probabilistic, multiclass cases, examining the area under the distance threshold curve. We constructed $\mathrm{AUC}_{\mathrm{dt}}$ using three distance metrics: City Block Distance, Jeffrey Divergence, and Earth Mover’s Distance along with Weighted JD and CB. We have found the random performance values for all distance metrics and showed that Jeffrey Divergence will produce a similar $\mathrm{AUC}_{\mathrm{dt}}$ when it is weighted and when it is not. This also gives us a comparison to AUC and $\mathrm{AUC}_{\mathrm{dt}}$ in that they both have known random classification and a perfect classification value of 1.00. We also found that regardless of the distance measure, the $\mathrm{AUC}_{\mathrm{dt}}$ consistently evaluated our classifier performance as matching the radiologists’ labels to about 40-50%. This confirms that $\mathrm{AUC}_{\mathrm{dt}}$ can be used as an evaluation measure. Furthermore, $\mathrm{AUC}_{\mathrm{dt}}$ presents an advantage for the multiclass case: ability to distinguish between misclassification of higher or lower labels by using weighted differences.

The use of the $\mathrm{AUC}_{\mathrm{dt}}$ is especially valuable for the case of medical image annotation. Distinguishing between five distinct semantic classes is essential in radiology, where the difference between a malignancy ranking of “3” and “4” could be whether to do a lung tissue biopsy or not [14]. Because $\mathrm{AUC}_{\mathrm{dt}}$ maintains the integrity of each distinct class, it is directly applicable to multiclass classification problems such as those presented within the LIDC.

In the future we will investigate how different distribution metrics can be selected based on the application purposes of the classifier itself; that there is no “perfect” evaluation measure for all cases. We will also explore how viewing this problem as an ordinal regression problem (instead of discrete, deterministic labels) will affect the area under the distance threshold curve. In this case, Earth Mover’s Distance will likely be an appropriate measure as it looks at distance amongst classes which is also important in ordinal regression.

Acknowledgements. This work was supported by the National Science Foundation under Grant No. 1062909. We would also like to thank Dimitry Zinovev of DePaul University’s College of Computing and Digital Media for providing us with the LIDC image features for each nodule.

References

1. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. 30, 1145–1159 (1997), doi:10.1016/j.bbr.2011.03.031
2. Drummond, C., Holte, R.: Cost curves: an improved method for visualizing classifier performance. Mach. Learn. 65, 95–130 (2006)
3. Zinovev, D., Furst, J., Raicu, D.: Building an ensemble of probabilistic classifiers for lung nodule interpretation. In: Tenth Intern. Conferen. on Mach. Learn. and App (ICMLA 2011), pp. 151–167. IEEE Press (December 2011), doi:10.1109/ICMLA.2011.44
4. Amor, N.B., Benferhat, S., Elouedi, Z.: Information-based evaluation functions for probabilistic classifiers. In: Eleventh Internat. Conferen. on Infor. Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 428–433 (July 2006)
5. Ling, C.X., Huang, J., Zhang, H.: AUC: a statistically consistent and more discriminating measure than accuracy. In: Proc. of Eighteenth Internat. Conf. on Artifical Intelligence (IJCAI 2003), pp. 519–526 (August 2003)
6. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In: Proc. Third Internat. Conf. on Knowledge Discovery and Data Mining (KDD 1997), pp. 43–48. AAAI Press (August 1997)
7. Fawcett, T.: ROC graphs: notes and practical considerations for data mining. Technical report, HPL-2003-4
8. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. 27, 861–874 (2006)
9. Hérnandez-Orallo, J., Flach, P., Ferri, C.: Brier curves: a new cost-based visualization of classifier performance. In: Proc. Twenty-Eighth Internat. Conf. on Mach. Learn (ICML 2011), pp. 585–592 (June 2011)
10. Hand, D.J., Till, R.T.: A simple generalization of the area under the ROC curve for multiple class classification problems. Machine Learning 45, 171–186 (2001)
11. Jain, P., Kapoor, A.: Active learning for large multi-class problems. Comp. Vision and Pattern Recogn (CVPR 2009), 762–769 (June 2009)

12. Liu, H., et al.: Comparing dissimilarity measures for content-based image retrieval. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 44–50. Springer, Heidelberg (2008)
13. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Sixth Internat. Conf. Comp. Vis. (ICCV 1998), pp. 59–66. IEEE Press (January 1998), doi:10.1109/ICCV.19
14. Raicu, D.S., Varutbangkul, E., Furst, J.D., Armato III, S.G.: Modeling semantics from image data: opportunities from LIDC. Internat. Jour. of Biomed. Eng. and Tech. 3(30:1-2), 83–113 (2009)

Author Index