Data Mining and Text Mining for Bioinformatics: Proceedings of the European Workshop

Held in Conjunction with ECML / PKDD- 2003 in Dubrovnik, Croatia
22 September, 2003

Tobias Scheffer and Ulf Leser (Editors)

Preface

In the past years, research in molecular biology and molecular medicine has accumulated enormous amounts of data. This includes genomic sequences gathered by the Human Genome Project, gene expression data from microarray experiments, protein identification and quantification data from proteomics experiments, and SNP data from high-throughput SNP arrays. However, our understanding of the biological processes underlying these data lags far behind. There is a strong interest in employing methods of knowledge discovery and data mining to generate models of biological systems. Mining biological databases imposes challenges which knowledge discovery and data mining have to address, and which form the focus of the European Workshop on Data Mining and Text Mining for Bioinformatics.

This volume contains the papers presented at the European Workshop on Data Mining and Text Mining for Bioinformatics, held at the European Conference on Machine Learning and the European Conference on Principles and Practice of Knowledge Discovery in Databases, in Dubrovnik, Croatia, on September 22, 2003. Three invited and ten contributed papers were presented at the workshop; invited presentations were given by

- Luc Dehaspe, PharmaDM: “Great Expectations: A To-Do List for the Biologist’s in Silico Research Assistant”,
- Udo Hahn, Freiburg University: “Challenging Natural Language Processors – Prospects for Bioinformatics in the Natural Language Engineering Age”, and
- Steven J. Barrett, GlaxoSmithKline Research and Development: “Recurring Analytical Problems within Drug Discovery and Development”.

We would like to thank the members of the program committee.

- Sourav Bhomwick, Nanyang Technological University, Singapore.
- Christian Blaschke, Centro Nacional de Biotecnologia.
- Vladimir Brusic, Biodiscovery Group, Institute for Infocomm Research, Singapore.
- Mark Craven, University of Wisconsin.
- Saso Dzeroski, Jozef Stefan Institute.
- George Forman, Hewlett Packard.
- Jiawei Han, University of Illinois at Urbana Champaign.
- Ross King, University of Wales, Aberystwyth, and PharmaDM.
- Adam Kowalczyk, Telstra.
- Stefan Kramer, University of Munich.
- Knut Reinert, Free University, Berlin.
- Steffen Schulze-Kremer, Max-Planck-Institute, Berlin.
- Myra Spiliopoulou, University of Magdeburg.
- Alfonso Valencia, Centro Nacional de Biotecnologia, Spain.
- David Vogel, AI Insight.
- Stefan Wrobel, Fraunhofer AiS and University of Bonn.
- Mohammed Zaki, Rensselaer Polytechnic Institute.

Additional reviews were written by Jörg Hakenberg. Based on these reviews, we selected nine research papers and one research note for presentation at the workshop.

We gratefully acknowledge support from the European Network on Excellence in Knowedge Discovery, KD-Net. We particularly wish to thank Codrina Lauth for her support.

Berlin, 14 July 2003

Tobias Scheffer
Ulf Leser

Humboldt University, Berlin

Table of Content

Invited Talks

Contributed Papers

Great expectations: a to do list for the biologist's in silico research assistant

Luc Dehaspe
PharmaDM
Kapeldreef, 60
B-3001 Leuven, Belgium
++32 16 387472
Luc.Dehaspe@PharmaDM.com

ABSTRACT

Witness the response to this workshop, biology is increasingly attractive to text and data mining researchers in search of an application domain to validate their technology. Vice-versa, biologists have set their hopes on text and data mining for providing them with the biologist-compliant software tools to *explore and digest* data directly (rather than indirectly, e.g., via a bioinformatician). The attraction is clearly mutual, but the match between the research agenda of the provider/miner and the needs and expectations of the user/biologist is far less obvious. This presentation looks at this complex relationship from the user perspective. Not a biologist myself, I will list some requirements that have come up in discussions with biologists on data and text mining applications. I will also present some (intermediate) research results inspired by those discussions.

Text mining requirements will be mainly taken from the EC funded Biological Text Mining project *BioMinT* (**www.biomint.org**), in particular from feedback provided by

1. two user partners (i.e., University of Manchester, and Swiss Institute of Bioinformatics) involved in database (i.e., PRINTS and SWISS-PROT) curation; and
2. biology researchers from partner institutions and interested companies.

For the data mining requirements I will focus on "integrated data analysis". Various interpretations of that highly popular phrase will be discussed. I will also argue that Relational Data Mining, supplemented with Data Cubes for visualization and interaction purposes, can address this issue quite naturally.

Challenging Natural Language Processors – Prospects for Bioinformatics in the Natural Language Engineering Age

Udo Hahn

Text Knowledge Engineering Lab

Universität Freiburg, Germany

The recent gain in maturity of natural language processing (NLP) methodology can, by and large, be attributed to a particular research (and funding) strategy actively pursued in the past decade. It is based on the external definition of challenging NLP tasks which can be probed by a number of research groups under controlled experimental conditions. Briefly considering the experience from past competition rounds in the field of document retrieval (TREC) and information extraction (MUC), this keynote will outline a framework for automatic knowledge capture from bioinformatics literature, in which task descriptions and corresponding evaluation scenarios meet special application requirements from the bioinformatics domain. This way, biological research might substantially benefit from state-of-the-art NLP techniques.

Recurring Analytical Problems within Drug Discovery and Development.

S. J. Barrett

Data Exploration Sciences, GlaxoSmithKline, Research and Development,

Greenford, Middlesex, UK

The overall processes driving pharmaceuticals discovery and development research involve many disparate kinds of problems and problem-solving at multiple levels of generality and specificity. The discovery/pre-clinical processes are also highly technology-driven and specific aspects may be more dynamic over time compared to developmental research which is conducted in a more conservatively controlled manner, conducive to regulatory requirements.

The widespread utilisation of robotics and miniaturisation of synthesis and screening technologies has greatly increased the through-put of discovery processes, which now yield vast amounts of richer data to the extent that discovery has evolved a critical dependence upon computerised systems. In tandem with this, the core need for specialists in disciplines related to biological and chemical information processing and analysis has increased greatly to enable organisation of this data into operational datamarts to support improved domain understanding and better decision-making for triaging compounds and reducing drug candidate attrition at later (more expensive) stages.

Combinatorial library design involves the identification and selection of 'virtual' molecules to physically make and incorporates multiple sub-problems : modelling biological target-specific activity (often incoporating multiple - usually unknown - mechanisms for binding/activity), ADME properties prediction, overcoming compound space biases,etc.

A key goal of this QSAR-type modelling is to generalise beyond the current data, and this is usually further complicated by massive class imbalance within previously sampled areas of chemical space. Methods involving learning from partially-labelled data, such as transduction, would be very useful if made effective.

The library design and optimisation processes are heavily dependent upon the quality of the outputs of primary compound screening – vast numbers of compounds are now screened and cross-screened, increasingly using 'high content' methods, and the newer discipline of cheminformatics now supports the goal of improving this process via screening set selection, stringent data analysis of hits, identification of screening biases, etc.

More recently, another source of vast complex data has evolved from the incorporation of the various 'omics platforms into pharmaceuticals research, yielding increasing volumes of genomics, proteomics, metabolomics and lipomics data. With this comes new analysis challenges for relating cellular changes to chemical activity and new possiblities for crossing the divide and extrapolating between animal and human data/models.

Genetic and SNP data from large population genetics studies and further initiatives to capture DNA from subjects in clinical trials yield yet more datasets to be analysed alongside primary endpoints in the search for 'surrogate' biomarkers.

Much of the data coming from these newer platforms such as gene-chips is very high dimensional and additionally problematic in being small N x Large M type data. Much of the analysis and dimensional reduction effort currently relies upon conventional multivariate statistical approaches such as PCA, PLS and PLS-DA.

Increasingly, data from disparate sources and disciplines are being combined to ask new types of questions that are often not those which the data were originally generated to support. This also presents further problems for analysis from multi-relational data and disparately sized data sources.

Exploratory, descriptive and predictive data-mining involving machine learning methods can contribute to these analyses in a variety of ways, and some examples are presented here for newer 'general-purpose' approaches such as genetic programming and support vector machines. The application of traditional methods such as clustering is also covered as well as the following domain areas/problems:

- Compound 'bioprofiling' and data-mining of HTS screening data
- QSAR
- SNPs marker identification
- predictive gene's selection from affymetrix data

Improving Information Extraction through Biological Correlation

Francisco M. Couto, Mário J. Silva
LaSIGE, Departamento de Informática
Faculdade de Ciências
Universidade de Lisboa
Campo Grande, 1749-016 Lisboa, Portugal
{fjmc,mjs}@di.fc.ul.pt

Pedro Coutinho
UMR 6098, Architecture et Fonction des
Macromolécules Biologiques
Centre National de la Recherche Scientifique
13402 Marseille CEDEX 20, France
pedro@afmb.cnrs-mrs.fr

ABSTRACT

We present a new method for improving the efficiency of information extraction systems applied to biological literature, using the correlation between structural and functional classifications of gene products. The method evaluates extracted information by checking if gene products from a common family match a common set of biological properties. To evaluate the method, we implemented it in a case-study, where the method annotated carbohydrate-active enzymes with functional properties extracted from literature. Each carbohydrate-active enzyme is assigned to one or more families of catalytic and carbohydrate-binding modules according to its modular structure. To compute the relatedness between functional properties, we implemented a semantic similarity measure in GO, a biological ontology. The results present our quantitative measure of the correlation between the modular structures and functional properties, showing that our method is a viable approach for automatic validation of extracted biological information.

1. INTRODUCTION

Relevant facts discovered in molecular biology research, like in other fields, have been mainly published in scientific journals throughout the last century [9]. Extracting knowledge from this large amount of unstructured information is a painful and hard task, even to an expert. The solution was to create and maintain structured databases, such as GenBank and SwissProt that collect and distribute biological information, in particular biological sequences. These databases describe properties of common biological entities, such as genes and proteins. In the past few decades, the explosion of data has caused the exponential growth of these databases [3], where efforts to compensate the lack of annotation of many entries (mostly genomic) are at the origin of the significant misannotations, underprediction and overprediction of properties found today in biologic databases [5].

Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics, held in conjunction with ECML/PKDD. 2003, Dubrovnik, Croatia

The integration of literature-derived annotation to different sources of data corrects and completes our knowledge about these biological entities [16]. However, a substantial amount of knowledge important to the integration is still only recorded in literature [18], which motivates the development of automatic tools that could extract part of this knowledge.

Information extraction methods find relevant information in unstructured texts and encode it in a structured form, like a database [11]. The application of these methods to biological literature is a recent research topic with a high activity despite its youth [2, 8, 17]. However, the use of different nomenclatures, different data classifications, and misannotations are hard barriers to overtake.

Our work aims to enhance information extraction systems for automatic annotation of biological databases through a new method, which we named CAC (Correlate the Annotations' Components). It evaluates whether an annotation is valid or not, based on the biological correlation between structure and function of gene products [14]. To check the effectiveness of CAC, we implemented it in case-study, where CAC annotated carbohydrate-active enzymes with functional properties extracted from literature. From these annotations, we identified a correlation between biological structure and function by using a semantic similarity measure [10].

The rest of this paper is structured as follows. Section 2 describes CAC method in detail. In Section 3 we present our case-study, describing its sources of biological information, the validation process, and the results. Section 4 discusses related work. Finally in Section 5 we express our main conclusions and directions for future work.

2. CAC

In CAC, we restrict an annotation to a pair composed by a gene product and a biological property. The gene products have to be classified in families according to their structural information, and the biological properties have to be organized in an ontology structured as a graph. CAC aims to validate an annotation only when its components have a biological relationship between them.

We define that two annotations converge if they relate differ-

ent gene products from a common family with similar biological properties. CAC assumes that an annotation is valid if it has a significant number of convergent annotations in any of its gene product's families. This assumption is supported by the dogma of molecular biology, which postulates that sequences should be correlated with their biological activity, i.e. gene products from a common family usually share a common set of biological properties. To validate an annotation in a family, it is not necessary to have all gene products from the family sharing the biological property, but only a significant subset of them.

Similarity of gene products and biological properties are fuzzy concepts, but we can still define metrics to estimate them. In our case, we need to define two type of metrics:

- Given two gene products g_1 and g_2 from a common family, we express their *structural distance* as $\Delta(g_1, g_2)$.
- Given two biological properties p_1 and p_2, we express their *functional distance* as $\Delta(p_1, p_2)$.

Since it is only necessary to calculate the structural distance between gene products from a common family, we should calculate the structural distance according to factors that characterize the family. For instance, we can measure the sequence similarity of common modules using BLAST (Basic Local Alignment Search Tool) [1]. Functional distance between biological properties can be measured through semantic similarity measures, which details are described in section 2.1

We define a measure of the convergence between two annotations as being proportional to its structural distance and inversely proportional to its functional distance. Without loss of generality, we consider a set of annotations $\mathcal{A}_f$ whose gene products belong to a common family f.

Definition 1. Given two annotations $(g_1, p_1) \in \mathcal{A}_f$ and $(g_2, p_2) \in \mathcal{A}_f$, their *annotation convergence* is defined as:

$$\Gamma((g_1, p_1), (g_2, p_2)) = \frac{\Delta(g_1, g_2)}{\Delta(p_1, p_2)}$$

We can visualize the notion of annotation convergence by representing an annotation as an arrow from a gene product to a biological property. If two arrows start from distant locations and finish in close locations, then they are convergent. In other words, two annotations are convergent if their structural distance is larger than their functional distance.

EXAMPLE 1. *Figure 1 presents three different cases for the annotations* $a_1=(g_1, p_1)$, $a_2=(g_2, p_2)$ *and* $a_3=(g_3, p_3)$, *whose components we placed on the shaded regions according to their structural or functional distance. In case (a) we have* $\Delta(g_1, g_2) = \Delta(g_2, g_3) = \Delta(g_1, g_3)/2$ *and* $\Delta(p_1, p_2) = \Delta(p_2, p_3) = \Delta(p_1, p_3)$, *which makes* $\Gamma(a_1, a_3) > \Gamma(a_1, a_2)$. *In case (b) we have* $\Delta(g_1, g_2) = \Delta(g_2, g_3) = \Delta(g_1, g_3)$ *and* $\Delta(p_1, p_2) = \Delta(p_2, p_3) = \Delta(p_1, p_3)/2$, *thus* $\Gamma(a_1, a_2) > \Gamma(a_1, a_3)$. *Finally, in case (c) we have* $\Gamma(a_1, a_2)=0$ *and* $\Gamma(a_2, a_3)$ $=\infty$, *since* a_1 *and* a_2 *annotate the same gene product (origin) whereas* a_2 *and* a_3 *annotate the same biological property (destiny).*

Since we defined the concept of annotation convergence by two measures from different universes, it is not reasonable to establish a coefficient from which we can consider the annotations convergent. However, it is possible to define a more flexible relation that considers two annotations convergent if their annotation convergence is greater than a certain value.

Definition 2. Given two annotations $a_1 \in \mathcal{A}_f$ and $a_2 \in \mathcal{A}_f$, and a threshold h, they are *h-convergent* if $\Gamma(a_1, a_2) \geq h$.

Finally, given a threshold h, we define the correlation degree of an annotation as the number of h-convergent annotations in a common family.

Definition 3. Given an annotation $a_0 \in \mathcal{A}_f$ and a threshold h, the *correlation degree* of a_0 for h is defined as $\mathcal{D}_h(a_0) = \#\{a_x : a_x \in \mathcal{A}_f \wedge \Gamma(a_0, a_x) \geq h\}$.

The method validates annotations that have a correlation degree larger than a certain value in at least one family. This value and the convergence threshold are parameters that statistical classification methods can adjust [26].

2.1 Semantic Similarity Measures

Semantic similarity measures compute distances between terms structured in a hierarchical taxonomy. Two kinds of approaches are prevalent: information content (node based) and conceptual distance (edge based). Information content considers the similarity between two terms the amount of information they share, where a term contains less information when it occurs very often. Conceptual distance is a more intuitive approach. It identifies the shortest topologic distance between two terms in the scheme taxonomy. Budanitsky et al. experimentally compared five different proposed semantic similarity measures in WordNet [10]. The comparison shows that Jiang and Conrath's semantic similarity measure provides the best results overall [19]. This semantic similarity measure is a hybrid approach, i.e. it combines information content and conceptual distance with some parameters that control the degree of each factor's contribution.

To compute the functional distance between functional properties we propose to use Jiang and Conrath's measure. However, to measure the distance between two functional properties, we must be able to compute the following factors: their closest common ancestor; the shortest path between each term and their common ancestor; and for each term in these paths its information content, its depth and the number of its direct descendents (i.e. local density).

The information content computation depends on the terms' frequency. We have to compute the number of occurrences

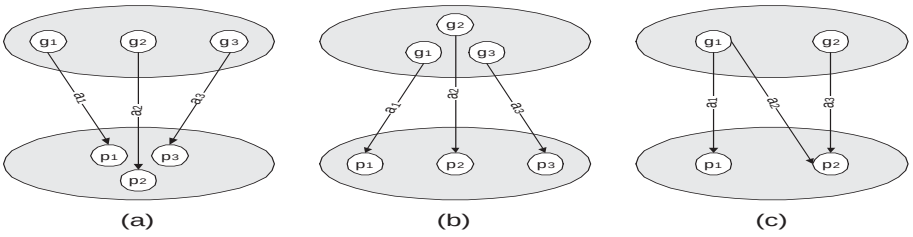


Figure 1: Annotations examples

of each term in the corpora. However, if a term occurs then all its ancestor terms also occur. Thus, we have to propagate the term occurrences throughout the hierarchy, reaching a frequency for the root node equal to the sum of all the occurrences, as it does not represent any relevant information.

The conceptual distance is based on the node depth and density factors. The node depth factor relies on the argument that similarity increases as we descend the hierarchy, since the relations are based on increasingly finer details. The density factor relies on the argument that when the parent node has several child nodes (high density) they tend to be more similar.

3. CASE-STUDY

This section presents a case-study to evaluate if CAC method is a viable approach. The case-study uses the following biological information sources:

- CAZy (Carbohydrate Active enZYmes) is a database of carbohydrate-active enzymes identified and classified in various families by careful sequence and structural comparisons [13]. It describes the families of structurally-related catalytic and carbohydrate binding modules (or functional domains) of enzymes that degrade, modify, or create glycosidic bonds. It also links the sequences to GenBank(GenPept) [6], SwissProt [4] and PDB [7] entries. These databases are repositories of gene and protein sequence and structural data used to characterize CAZy's enzymes.

- GO (Gene Ontology) provides a structured controlled vocabulary of gene and protein biological roles [12]. The three organizing principles of GO are molecular function, biological process and cellular component. Rison et al. discuss the reasons for choosing GO as the functional scheme in a survey about functional classification schemes [23]. They describe GO as "representative of the 'next generation' of functional schemes". Unlike other schemes, GO is not a tree-like hierarchy, but a directed acyclic graph (DAG), which permits a more complete and realistic annotation.

CAZy and GO provide the structural and functional classification schemes, respectively, for our case-study. Thus, CAZy enzymes and GO terms will assume the role of gene products and biological properties, respectively, in our concept of annotation.

PubMed is an online interface for the MEDLINE database [20]. MEDLINE provides a vast collection of abstracts and bibliographic information, which have been published in biomedical journals. In this paper, we consider a document as a bibliographic item whose citation is present in MEDLINE.

3.1 Validation

To assure that CAC method produces valuable results, we need to identify a correlation between CAZy and GO classification schemes. Our strategy to identify this correlation is to compare the probability of extracting similar terms in a family with the probability of extracting similar terms in general. We structured the validation process in three steps:

1. Retrieve a set of documents related to each enzyme from available literature.

2. Extract annotations that associates each enzyme with GO terms extracted from its related documents.

3. Compute the probability of similar terms inside a family and in general.

Instead of extracting information from the entire available corpora, we retrieve only documents somehow related to each enzyme. CAZy links its enzymes to external databases (GenBank, SwissProt and PDB) that contain bibliographic references. We retrieve for each enzyme the documents cited in its linked external database entries.

We extract the annotations based on the occurrences in text [18, 15, 25]. We assume that if a document mentions a GO term then there is an underlying biological relation between the enzymes related to the document and the GO term, i.e. we annotated the enzymes with the GO term. This is a very strong assumption and a source of misannotations. However, it satisfies our goal of evaluating CAC by using it to filter misannotations.

The three organizing principles of GO represent three orthogonal ontologies, thus we did not mix annotations from different organizing principles. We choose to start by extracting only molecular functional terms, given its greater importance to CAZy.

We consider that two GO terms are similar if their functional distance is smaller than a given threshold. We implemented Jiang and Conrath's semantic similarity measure in GO to compute the functional distance. Given a specific term, we can define its probability of similar terms in a set of terms

as the number of its similar terms over the total number of terms.

Definition 4. Given a term t, a set of terms T, and a similarity threshold k, we define the term's *probability of similar terms* in the set as:

$$\mathcal{P}_{sim}(t,T) = \frac{\#\{t_x : t_x \in T \ \wedge \ 0 \leq \Delta(t,t_x) \leq k\}}{\#T}$$

We assign to each family the set of terms annotated with its enzymes.

Definition 5. Considering the set of all extracted annotations $\mathcal{A}$, we define *the set of all extracted terms* as $\mathcal{T} = \{t : (e,t) \in \mathcal{A}\}$, and given a family f we define its *set of terms* as $\mathcal{T}_f = \{t : (e,t) \in \mathcal{A} \wedge e \in f\}$.

We define the probability of extracting similar terms in a particular family as the average of its term's probability of similar terms in the family.

Definition 6. Given a family f, we define the *family's probability of extracting similar terms in it* as $\mathcal{P}_{in}(f) = \overline{\{\mathcal{P}_{sim}(t, \mathcal{T}_f \setminus \{t\}) : t \in \mathcal{T}_f\}}$.

We define the probability of extracting similar terms in a family as the average of $\mathcal{P}_{in}(f)$ for all the families.

Definition 7. Given a set of families F, we define the *probability of extracting similar terms in a family* as $\mathcal{P}_{in} = \{\mathcal{P}_{in}(f) : f \in F\}$.

To provide a good source of comparison, we define the probability of extracting similar terms in general analogously to $\mathcal{P}_{in}$. The only difference is that for each family's term we identify its similar terms in all the extracted annotations.

Definition 8. Given a family f, we define the *family's probability of extracting similar terms in general* as $\mathcal{P}_{all}(f) = \overline{\{\mathcal{P}_{sim}(t, \mathcal{T} \setminus \{t\}) : t \in \mathcal{T}_f\}}$.

Definition 9. Given a set of families F, we define the *probability of extracting similar terms in general* as $\mathcal{P}_{all} = \{\mathcal{P}_{all}(f) : f \in F\}$.

If $\mathcal{P}_{in}$ is significantly larger than $\mathcal{P}_{all}$ for a given similarity threshold then there is a correlation between CAZy and GO classification schemes, which is a strong argument to conclude that the annotations validated by CAC method have a larger precision than all the extracted annotations.

EXAMPLE 2. *Consider* $\mathcal{T}=\{t_1,t_2,t_3\}$ *with* $\Delta(t_i,t_j) = i+j$, $\mathcal{T}_f=\{t_1,t_2\}$ *for the family* f, *and* $k=4$. *Then we have* $\mathcal{P}_{in}(f) = \overline{\{\mathcal{P}_{sim}(t_1,\{t_2\}), \mathcal{P}_{sim}(t_2,\{t_1\})\}} = \overline{\{1,1\}}=1$, $\mathcal{P}_{all}(f)=\overline{\{\mathcal{P}_{sim}(t_1,\{t_2,t_3\}), \mathcal{P}_{sim}(t_2,\{t_1,t_3\})\}} = \overline{\{1,1/2\}}=3/4$, *and therefore* $\mathcal{P}_{in}(f) > \mathcal{P}_{all}(f)$ *because* $\Delta(t_2,t_3) > k$.

	bibliographic references	distinct documents
GenBank	22849	4575
SwissProt	8998	4006
PDB	3561	785
Total		6377

Table 1: Number of items retrieved

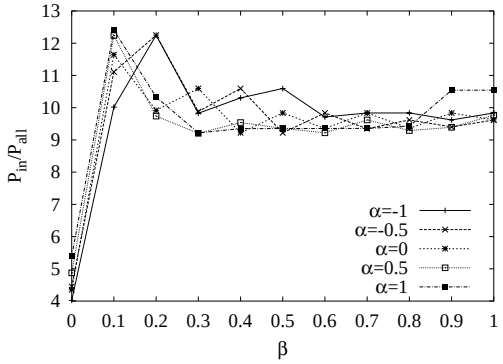


Figure 2: $\mathcal{P}_{in}$ over $\mathcal{P}_{all}$

3.2 Results

This section describes the results of our last analysis performed on the January 2003 release of GO and CAZy databases. Table 1 presents the number of bibliographic references retrieved and the number of documents cited by them. From these documents, we extracted 13869 annotations. We computed the probability of extracting similar terms for 90 families of glycoside hydrolases (GHs), which are the best curated enzymes in CAZy. These families were associated with 3748 documents, from which were extracted 343 distinct GO terms.

Figure 2 shows the ratio of $\mathcal{P}_{in}/\mathcal{P}_{all}$. We computed these values for the similarity threshold that maximized the difference between $\mathcal{P}_{in}$ and $\mathcal{P}_{all}$. The parameters α and β control the degree of how much the node depth and density factors contribute to semantic similarity computation. These contributions become less significant when α approaches 0 and β approaches 1. The values achieved show that the

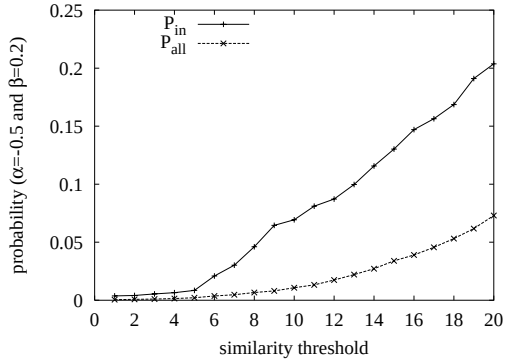


Figure 3: $\mathcal{P}_{in}$ against $\mathcal{P}_{all}$

probability of extracting similar terms is significantly larger inside a family, as anticipated. The graph has a peak when $\beta \in [0.1, 0.2]$, where $\mathcal{P}_{in}/\mathcal{P}_{all}$ is larger than 12 for all α, except for α=0. This means that the density of the DAG and the depth of each node are important conceptual distance factors to amplify the correlation.

The maximum value of $\mathcal{P}_{in}/\mathcal{P}_{all}$ is obtained with α=1.0 and β=0.1. Figure 3 uses this configuration to show $\mathcal{P}_{in}$ against $\mathcal{P}_{all}$ for different similarity thresholds. As expected, both probabilities are proportional to the similarity threshold, since a larger similarity threshold implies also a larger number of similar terms. The relevant fact in the graphic is that $\mathcal{P}_{in}$ is always significantly larger than $\mathcal{P}_{all}$, which shows that enzymes with similar modular structure tend to be annotated with similar functional terms.

4. RELATED WORK

Different techniques for computing similarity measures between terms have been developed to address a variety of problems. Early approaches were based only on counting edge distances between terms [22]. These were later improved by using the information content of each term, a classic Information Retrieval technique [24].

More recently, Lord et al. investigated an information content semantic similarity measure, and its application to annotations found in SwissProt [21] that also associate gene products with GO terms. They present results showing that semantic similarity is correlated with sequence similarity, i.e. function is correlated with structure. Since we propose an effective information extraction tool for biological literature, we evaluated the measure of this correlation with annotations automatically extracted from free text, instead of using human curated annotations. In our work, we replaced the sequence similarity by a modular structure classification, which is a more precise structural classification. We also tested the application of a hybrid semantic similarity measure, which integrates the information content with other valuable factors.

5. CONCLUSIONS & FUTURE WORK

We presented CAC method, which improves the efficiency of information extraction systems applied to biological literature. The method uses the correlation between structure and function to increase the precision of automatically extracted annotations.

We automatically annotated carbohydrate-active enzymes with functional terms extracted from literature. From the annotations, we computed the probability of extracting similar terms, which was significantly larger for enzymes from a common family. This result shows a correlation between modular structure and molecular function, which assures that CAC method increases the precision of extracted annotations, thus making it an effective tool for automatic information extraction of biological literature.

We implemented a hybrid semantic similarity measure to compute the similarity between GO terms, which shows that this kind of measures is feasible in a biological setting. Moreover, our results show that the information content measure improves its effectiveness when integrated with a conceptual distance measure.

To present results where the annotations validated by CAC have a larger precision than all the extracted annotations we first need to curate the extracted annotations. Besides human curation, we also intend to incorporate human curated annotations recorded in different biological sources to accelerate this procedure.

6. REFERENCES

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, pages 403–410, 1990.

[2] M. Andrade and P. Bork. Automated extraction of information in molecular biology. *FEBS Letters*, 476:12–17, 2000.

[3] T. Attwood and D. Parry-Smith. *Introduction to Bioinformatics*. Longman Higher Education, 1999.

[4] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, 28:45–48, 2000.

[5] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach, Second Edition*. MIT Press, 2001.

[6] D. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, B. Rapp, and D. Wheeler. GenBank. *Nucleic Acids Research*, 30:17–20, 2002.

[7] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000.

[8] C. Blaschke, L. Hirschman, and A. Valencia. Information extraction in molecular biology. *Briefings in BioInformatics*, 3:1–12, 2002.

[9] C. Blaschke, R. Hoffmann, J. Oliveros, and A. Valencia. Extracting information automatically from biological literature. *Comparative and Functional Genomics*, 2:310–313, 2001.

[10] A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000), Pittsburgh, PA, June 2001.

[11] C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18:65–79, 1997.

[12] T. G. O. Consortium. Creating the gene ontology resource: design and implementation. *Genome Res*, 11:1425–1433, 2001.

[13] P. Coutinho and B. Henrissat. Carbohydrate-active enzymes: an integrated database approach. *Recent Advances in Carbohydrate Bioengineering*, pages 3–12, 1999.

[14] F. Couto, M. Silva, and P. Coutinho. Curating extracted information through the correlation between structure and function. In *third meeting of the special interest group on Text Data Mining*. Intelligent Systems for Molecular Biology (ISMB), 2003.

[15] J. D. et al. Mining MEDLINE: Abstracts, sentences, or phrases? In *PSB*, pages 326–337, 2002.

[16] M. Gerstein. Integrative database analysis in structural genomics. *Nature Structural Biology*, Structural genomics supplement:960–963, November 2000.

[17] L. Hirschman, J. Park, J. Tsujii, L. Wong, and C. H. Wu. Accomplishments and challenges in literature data mining for biology. *Bioinformatics*, 18(12):1553–1561, 2002.

[18] T. Jenssen, A. L. greid, J. Komorowski, and E. Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28:21–28, may 2001.

[19] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research on Computational Linguistics (ROCLING X)*, Taiwan, 1997.

[20] MEDLINE. PubMed database at the National Library of Medicine. www.ncbi.nlm.nih.gov.

[21] P.W.Lord, R. Stevens, A. Brass, and C.A.Goble. Semantic similarity measures as tools for exploring the Gene Ontology. In *Pacific Symposium on Biocomputing*, pages 601–612, 2003.

[22] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems*, 19(1):17–30, 1989.

[23] S. Rison, T. Hodgman, and J. Thornton. Comparison of functional annotation schemes for genomes. *Funct. Integr. Genomics*, 1:56–69, 2000.

[24] G. Salton and M. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[25] B. Stapley and G. Benoit. Biobiblimetrics: Information retrieval and visualization from co-occurrences of gene names in MEDLINE abstracts. In *PSB*, pages 326–337, 2002.

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

Incremental SVM and Visualization Tools for Bio-medical Data Mining

Thanh-Nghi Do, François Poulet

ESIEA Recherche
38, rue des Docteurs Calmette et Guérin
Parc Universitaire de Laval-Changé
53000 Laval - France
{dothanh, poulet}@esiea-ouest.fr

Abstract. Most of the bio-data analysis problems process datasets with a very large number of attributes and few training data. This situation is usually suited for support vector machine (SVM) approaches. We have implemented a new column-incremental linear proximal SVM to deal with this problem. Without any feature selection step, the algorithm can deal with very large datasets (at least 10^9 attributes) on standard personal computers. We have evaluated its performance on bio-medical datasets and compared the results with other SVM algorithms. Furthermore, we propose a new visualization tool to try to explain the results of automatic SVM algorithms by displaying the data distribution according to the distance to the separating plane.

1 Introduction

In recent years, data stored worldwide has doubled every 20 months, so that the need to extract knowledge from very large databases is increasing. Knowledge Discovery in Databases (KDD) has been defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Data mining can be defined as one particular step of the KDD process: the identification of interesting structures in data. It uses different algorithms for classification, regression, clustering or association rules. These new progresses in data mining have already been applied to bio-data analysis (Zaki et al., 2002). SVM algorithms (Vapnik, 1995) are one of the most well-known of a class of performing methods for bio-data analysis. Recently, a number of powerful SVM learning algorithms have been proposed (Bennett et al., 2000). SVM uses the idea of kernel substitution for classifying the data in a high dimensional feature space. They have been successfully applied to various applications, for example in face identification, text categorization, bioinformatics (Guyon, 1999). Especially, SVM performs particularly well in bio-data analysis problems (Mukherjee et al., 1999, Brown et al., 1999, Furey et al., 2000). However, SVM solution is obtained from quadratic programming problem possessing a global solution, so that, the computational cost of a SVM approach depends on the optimization algorithm used. The very best algorithms today are typically quadratic and require multiple scans of the data. The new proximal SVM algorithm proposed by (Fung et al., 2001) changes the inequality constraints to equalities in the optimization problem, thus the training task requires the solution of a system of linear equations, so that PSVM is very fast to train. Furthermore, PSVM can construct incrementally the model without loading the whole dataset in main memory. The Sherman-Morrison-Woodbury formula (Golub et al., 1996) is used to adapt PSVM to a row-incremental or column-incremental version (but not both yet). Thus the algorithm can deal with large datasets (at least 10^9 in one dimension: row or column) on standard personal computers (with standard RAM and disk capacities). The algorithm has been used on bio-medical datasets, the results are compared with some others in terms of learning time and classification accuracy. We have also developed a visualization tools to try to explain the SVM results. The display of the datapoint distribution according to the distance to the separating plane helps us to verify the robustness of obtained models.

We summarize the content of the paper now. In section 2, we introduce very general SVM classifier. In section 3, we describe PSVM and its incremental versions. We present the graphical method for visualizing the SVM results in section 4. Some numerical test results on bio-medical datasets can be found in section 5 before the conclusion in section 6.

Some notations will be used in this paper: all vectors are column vectors, the 2-norm of the vector x is denoted by $\|x\|$, the matrix A[mxn] is the m training points in the n-dimensional input space R^n. The diagonal matrix D[nxn] of ± 1 contains the classes +1 and –1 of m training points. e is the column vector of 1. w, b are the coefficients and the scalar of the hyperplane. z is the slack variable and ν is a positive constant. The identity matrix is denoted I.

2 Support Vector Machines

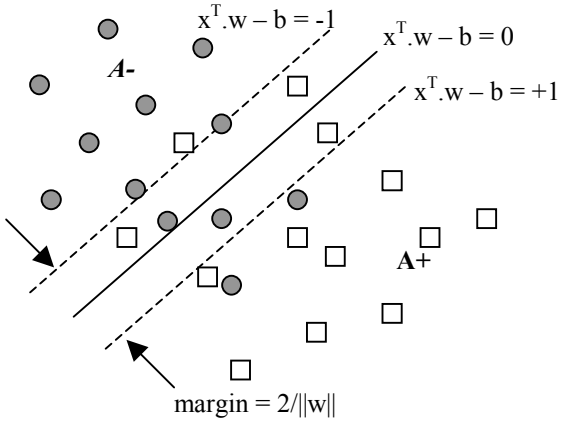


Fig. 1. Linear separation of the datapoints into two classes

Let us consider a linear binary classification task, as depicted in figure 1, with *m* data points in the n-dimensional input space R^n, represented by the mxn matrix *A*, having corresponding labels ±1, denoted by the mxm diagonal matrix *D* of ±1.

For this problem, the SVM try to find the best separating plane, i.e. furthest from both class +1 and class -1. It can simply maximize the distance or margin between the support planes for each class ($x^Tw – b = +1$ for class +1, $x^Tw – b = -1$ for class -1). The margin between these supporting planes is 2/||w||. Any point falling on the wrong side of its supporting plane is considered to be an error. Therefore, the SVM has to simultaneously maximize the margin and minimize the error. The standard SVM solution with linear kernel is given by the following quadratic program (1):

$$\begin{aligned} \min \quad & f(z,w,b) = \nu e^T z + (1/2)\|w\|^2 \\ & \text{s.t.} \quad D(Aw - eb) + z \geq e \end{aligned} \qquad (1)$$

where slack variable $z \geq 0$ and constant $\nu > 0$ is used to tune errors and margin size.

The plane (w,b) is obtained by the solution of the quadratic program (1). And then, the classification function of a new data point *x* based on the plane is:

$$f(x) = \text{sign}(w.x – b) \qquad (2)$$

SVM can use some other classification functions, for example a polynomial function of degree d, a RBF (Radial Basis Function) or a sigmoid function. To change from a linear to non-linear classifier, one must only substitute a kernel evaluation in the objective function instead of the original one. The details can be found in (Bennett et al. 2000) and (Cristianini et al., 2000).

3 Linear Proximal Support Vector Machines

The proximal SVM classifier proposed by Fung and Mangasarian changes the inequality constraints to equalities in the optimization problem (1) and adding a least squares 2-norm error into the objective function *f*, it changes the formulation of the margin maximization to the minimization of $(1/2)\|w,b\|^2$. Thus substituting for *z* from the constraint in terms (w,b) into the objective function *f* we get an unconstraint problem (3):

$$\min f(w,b)= (\nu/2)\|e – D(Aw–eb)\|^2 + (1/2)\|w,b\|^2 \qquad (3)$$

The Karush-Kuhn-Tucker optimality condition of (3) will give the linear equation system (4) of (n+1) variables (w,b):

$$[w_1\ w_2\ ..\ w_n\ b]^T = (I/\nu + E^TE)^{-1}E^TDe \qquad (4)$$

where $E = [A \quad -e]$

Therefore, the linear PSVM is very fast to train because it expresses the training in terms of solving a set of linear equations of (w,b) instead of quadratic programming. The accuracy can be compared with standard SVM. Note that all we need to store in memory is the (m)x(n+1) training data matrix E, the (n+1)x(n+1) matrix E^TE and the (n+1)x1 vector $d = E^TDe$. If the dimensional input space is small enough (less than 10^3), even if there are millions datapoints, PSVM is able to classify them on a standard personal computer. For example, the linear PSVM can easily handle large datasets as shown by the classification of 2 million 10-dimensional points in 15 seconds on a Pentium-4 (2.4GHz, 256 Mb RAM, Linux).

The algorithm is limited by the storage capacity of the (m)x(n+1) training data matrix E. In order to deal with very large (at least one billion points) datasets, the incremental version is extended from the computation of E^TE and $d=E^TDe$.

3.1 Row-incremental PSVM (Fung et al., 2002)

We can split the training dataset E into blocks of lines E_i, D_i and compute E^TE and $d=E^TDe$ from these blocks:

$$E^TE = \sum E_i^T E_i$$
$$d = \sum d_i = \sum E_i^T D_i e$$

For each step, we only need to load the (blocksize)x(n+1) matrix E_i and the (blocksize)x1 vector D_ie for computing E^TE and $d=E^TDe$. Between two incremental steps, we need to store in memory (n+1)x(n+1) and (n+1)x1 matrices although the order of the dataset is one billion data points. The authors have performed the linear classification of one billion data points in 10-dimensional input space into two classes in less than 2 hours and 26 minutes on a Pentium II (400 MHz, 2 GB RAM, 30% of the time being spent to read data from disk).

3.2 Column-incremental PSVM

The algorithm described in the previous section can handle datasets with a very large number of datapoints and small number of attributes. But some applications (like bioinformatics or text mining) require datasets with a very large number of attributes and few training data. Thus, the (n+1)x(n+1) matrix E^TE is too large and the solution of the linear equation system of (n+1) variables (w,b) has a high computational cost. To adapt the algorithm to this problem, we have applied the Sherman-Morrison-Woodbury formula to the linear equation system (4) to obtain:

$$[w_1\ w_2\ ..\ w_n\ b]^T = (I/\nu + E^TE)^{-1}E^TDe = \nu E^T[De - (I/\nu + EE^T)^{-1}EE^TDe] \qquad (5)$$
where $E = [A \quad -e]$

The solution of (5) depends on the inversion of the (m)x(m) matrix $(I/\nu + EE^T)$ instead of the (n+1)x(n+1) matrix $(I/\nu + E^TE)$ in (4). The cost of storage and computation depends on the number of training data. This formulation can handle datasets with very large number of attributes and few training data.

We have imitated the row-incremental algorithm for constructing the column-incremental algorithm able to deal with very large number of dimensions.

The data are split in blocks of columns E_i and then we perform the incremental computation of $EE^T = \sum E_iE_i^T$. For each step, we only need to load the (m)x(blocksize) matrix E_i for computing EE^T. Between two incremental steps, we need to store in memory the (m)x(m) matrix EE^T although the order of the dimensional input space is very high.

With these two formulations of the linear incremental PSVM, we are able to deal with very large datasets (large either in training data or number of attributes, but not yet both simultaneously). We have used them to classify biomedical datasets with interesting results in terms of learning time and classification accuracy.

4 Visualization of Linear SVM Results

Although the SVM algorithms have been successfully applied to a number of applications, they provide limited information. Most of the time, the user only knows the classification accuracy and the hyperplane equation. It is difficult to really explain or verify the constructed model or to understand why the SVM algorithms are more efficient than the other ones. Visualization techniques can be used to improve the results comprehensibility. We have developed a new method that can help the user to understand the model constructed by the SVM algorithm.

While the classification task is processed (based on the hyperplane obtained), we also compute the datapoint distribution according to the distance to the hyperplane as shown in figure 2.

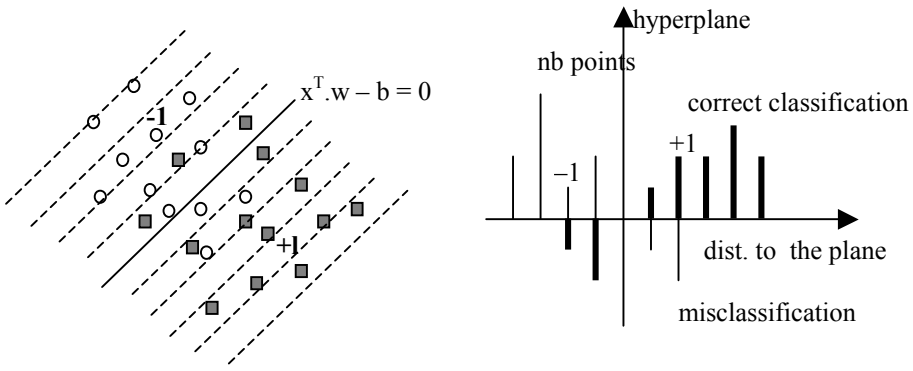


Fig. 2. Datapoints distribution based on parallel planes

For each class, the positive distribution is the set of correctly classified data points, and the negative distribution is the set of misclassified data points. An example of such a distribution is shown in the right part of the figure 2. The method is applied to Lung Cancer dataset (2 classes, 32 training, 149 testing, 12533 numerical attributes) as shown in figure 3.

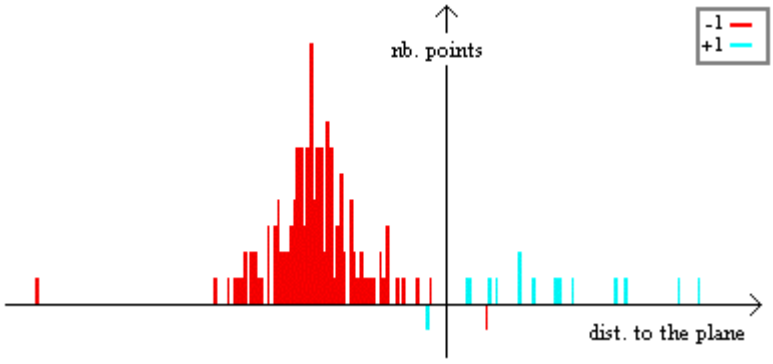


Fig. 3. Data distribution of Lung Cancer data according to distance to the hyperplane

The visualization of the datapoint distribution according to the distance to the separating plane helps us to evaluate the robustness of obtained models by SVM algorithm.

5 Numerical Test Results

The software program is written in C/C++ on SGI-O2 workstation (IRIX) and PC (Linux). To validate the performances of the incremental algorithms, we have classified bio-medical datasets (Jinyan et al., 2002) and the Thrombin drug design dataset from the KDD cup'01 (Hatzis et al., 2001). These datasets are described in table 1.

Table 1. Dataset description.

	classes	datapoints	attributes	evaluation method
ALL-AML Leukemia	2	72	7129	38 Tr – 34 Tst
Breast Cancer	2	97	24481	78 Tr – 19 Tst
Central Nervous System	2	60	7129	leave-one-out
Colon Tumor	2	62	2000	leave-one-out
MLL Leukemia	3	72	12582	57 Tr – 15 Tst
Ovarian Cancer	2	253	15154	leave-one-out
Prostate Cancer	2	136	12600	102 Tr – 34 Tst
Subtypes of Acute Lymphoblastic Leukemia	7	327	12558	215 Tr – 112 Tst
Lung Cancer	2	181	12533	32 Tr – 149 Tst
Translation Initiation Sites	2	13375	927	10-fold

Thrombin Drug Design	2	2543	139351	1909 Tr – 634 Tst

Thus, we have obtained the results (concerning the training time and accuracy) shown in table 2 on a personal computer Pentium-4 (2.4GHz, 256 Mb RAM, Linux Redhat 7.2). Without any feature selection, almost all datasets are classified by the column-incremental linear PSVM with one exception Translation Initiation Sites is classified by the row-incremental and the column-incremental (*) versions of the linear PSVM. The one-against-all approach has been used to classify multi-class datasets (more than 2 classes), thus we have taken the average accuracy. The results are compared with the linear kernel of SVMLight (Joachims, 2002).

Table 2. Numerical test results.

	Learning time (secs)		Accuracy (%)	
	IPSVM	SVMLight	IPSVM	SVMLight
ALL-AML Leukemia	**0.64**	9.14	**97.06**	**97.06**
Breast Cancer	**4.43**	269.66	**78.95**	73.68
Central Nervous System	**0.68**	15.29	**71.67**	68.33
Colon Tumor	**0.41**	1.80	**90.32**	**90.32**
MLL Leukemia	**3.42**	133.68	**100**	97.78
Ovarian Cancer	**12.51**	403.60	**100**	**100**
Prostate Cancer	**2.73**	61.97	**97.06**	79.41
SAL Leukemia	**8.26**	42.10	**98,65**	97.41
Lung Cancer	**1.25**	20.80	**98.66**	**98.66**
Translation Initiation Sites	**63** (4238*)	314	90.05	**92.41**
Thrombin Drug Design	425.12	**88.87**	**79.02**	76.34

As we can see in table 2, the incremental version of linear PSVM is generally outperforming SVMLight on learning time and classification correctness. The results shown in table 2 indicate that the column-incremental PSVM can be compared with standard SVM in terms of classification accuracy. But its training time is always better (from 4 to 60 times) than SVMLight one. The only one exception is the Thrombin Drug Design: this dataset has very few not null values (0.6823 %) and then SVMLight is faster than our column-incremental PSVM because we did not use a compressed data representation like SVMLight. The other algorithms can not run even with 1 GB RAM. They usually have to use a pre-processing task. Our column-incremental PSVM has given 79.02 % accuracy without any feature selection. The results concerning the Translation Initiation Sites show the interest of being able to choose between the column or the row incremental versions of the PSVM. The time needed for the classification task is divided by 70 (with exactly the same accuracy) when using the most appropriate version of the incremental algorithms.

6 Conclusion and Future Work

We have presented a new implementation of incremental linear PSVM which can deal with large datasets (at least 10^9 in one dimension: row or column if the other one stays under 10^4) on standard personal computers without any pre-processing task. The Sherman-Morrison-Woodbury formula is used to adapt PSVM to row-incremental or column-incremental. The algorithm has been estimated on bio-medical datasets. The column-incremental is a very convenient way to handle bio-medical datasets because it avoids loading the whole dataset in main memory and is very fast to train datasets with a very high number of attributes and few training data. Its performance concerning training time and classification accuracy can be compared with standard SVM.

We have also proposed a way to visualize SVM results. The datapoints distribution according to the distance to the separating plane helps us to estimate the robustness of models obtained by SVM algorithms.

A forthcoming improvement will be to extend these algorithms to the non-linear kernel cases. Another one will be to combine our method with other graphical techniques (Fayyad et al., 2001) to construct another kind of cooperation between SVM and visualization tools for applying it to bio-medical data analysis.

References

1. Bennett, K., Campbell, C.: Support Vector Machines : Hype or Hallelujah ?. in SIGKDD Explorations, Vol.2, Issue 2 (2000) 1-13

2. Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Ares, M., and Haussler, D.: Support Vector Machine Classification of Microarray Gene Expression Data. Technical Report UCSC-CRL-99-09, Dept. of Computer Science, University of California, Santa Cruz, USA (1999)
3. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, (2000)
4. Fayyad, U., Grinstein, G., Wierse, A.: Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann Publishers (2001)
5. Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., and Haussler, D.: Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. in Bioinformatics, Vol. 16, No. 10 (2000) 906-914
6. Fung, G., Mangasarian O.: Proximal Support Vector Machine Classifiers. in proc. of the 7th ACM SIGKDD, Int. Conf. on KDD'01, San Francisco, USA (2001) 77-86
7. Fung, G., Mangasarian O.: Incremental Support Vector Machine Classification. in proc. of the 2nd SIAM Int. Conf. on Data Mining SDM'02 Arlington, Virginia, USA (2002) 247-260
8. Golub, G., van Loan, C.: Matrix Computations. The John Hopkins University Press, Baltimore, Maryland, 3rd edition (1996)
9. Guyon, I.: Web Page on SVM Applications. http://www.clopinet.com/isabelle/Projects/-SVM/app-list.html
10. Hatzis, C., Page, D.: KDD Cup 2001. http://www.cs.wisc.edu/~dpage/kddcup2001
11. Jinyan, L., Huiqing, L.: Kent Ridge Bio-medical Dat Set Repository (2002). http://sdmc.-lit.org.sg/GEDatasets
12. Joachims, T.: SVM-Light : Support Vector Machine (2002). http://www.cs.cornell.edu/-People/tj/svm_light/
13. Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J., and Poggio, T.: Support Vector Machine Classification of Microarray Data. AI Memo 1677, MIT (1999)
14. Poulet, F., Do, N.: Mining Very Large Datasets with Support Vector Machine Algorithms. in proc. of the 5th ICEIS, Int. Conf. on Enterprise Information Systems, Angers, France (2003) 140-147
15. Vapnik V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
16. Zaki, M., Wang, J., Toivonen, H.: BIOKDD 2002: Recent Advances in Data Mining for Bioinformatics. In SIGKDD Explorations, Vol. 4, Issue 2 (2002) 112-114

litsift: Automated Text Categorization in Bibliographic Search

Lukas C. Faulstich
Peter F. Stadler
Bioinformatik, Institut für Informatik
Universität Leipzig, Germany
{lukas.faulstich, peter.stadler}
@bioinf.uni-leipzig.de

Caroline Thurner
Christina Witwer
Institut für Theoretische Chemie und
Strukturbiologie, Universität Wien, Austria
{caro,xtina}@tbi.univie.ac.at

ABSTRACT

In bioinformatics there exist research topics that cannot be uniquely characterized by a set of key words because relevant key words are (i) also heavily used in other contexts and (ii) often omitted in relevant documents because the context is clear to the target audience. Information retrieval interfaces such as entrez/Pubmed produce either low precision or low recall in this case. To yield a high recall at a reasonable precision, the results of a broad information retrieval search have to be filtered to remove irrelevant documents. We use automated text categorization for this purpose.

In this study we use the topic of conserved secondary RNA structures in viral genomes as running example. Pubmed result sets for two virus groups, *Picornaviridae* and *Flaviviridae*, have been manually labeled by human experts. We evaluated various classifiers from the Weka toolkit together with different feature selection methods to assess whether classifiers trained on documents dedicated to one virus group can be successfully applied to filter literature on other virus groups. Our results indicate that in this domain a bibliographic search tool trained on a reference corpus may significantly reduce the amount of time needed for extensive literature recherches.

Keywords

Automated Text Categorization, Document Filtering

1. INTRODUCTION

An important part of bioinformatics research is the comparison of computational results with experimental results. These are, unfortunately, often hidden in the vast body of molecular biology literature. More often than not, the data that are of interest for a particular computational study are mentioned only in passing and in a different context in the experimental literature. As a concrete example we consider here the survey of conserved RNA secondary structures in viral genomes[1] that has been initiated a few years ago by the Vienna group [9, 14, 11]. To our surprise, the bibliographic search for experimental evidence of and further information on *RNA secondary structures* in a given group of virus — a seemingly rather straightforward task — turned out to be more tedious than the work on the actual sequence and structure data.

There are several reasons for this difficulty: (i) RNA secondary structure is usually referred to only as *secondary structure* or simply as *structure* since the context *RNA* is clear. The term *secondary structure*, however, appears much more frequently in the context of protein structures for the same virus group because proteins are usually discussed more frequently and in much more detail. (ii) RNA secondary structures are rarely the main topic of research papers on viruses. Rather, only one or a few paragraphs are devoted to them. (iii) With few exceptions there is no well-established nomenclature of RNA features in viruses so that keyword searches for specific structural motifs are not very effective. (iv) Relevant articles are written by authors from rather diverse scientific communities, from clinical virologists to structural biologists.

Our target topic of *"conserved RNA secondary structure in viral genomes"* consists of several subtopics, each dedicated to a specific group of RNA viruses (e.g., *Picornaviridae*, *Flaviviridae*, *Coronaviridae*, or *Hepadnaviridae*). For some of these subtopics, manually labeled document corpora exist. The question addressed in this exploratory study is whether classifiers trained for one subtopic can be applied successfully to other subtopics. This would be in particular attractive for subtopics with a large amount of available literature, e.g., on the HIV virus in the case of *Retroviridae*. In our context, successful means a high recall (e.g., 80%) with a not too low precision (e.g., 30%) because the emphasis is on finding most of the relevant literature with a tolerable overhead caused by false positives.

Our goal is to make bibliographic search more effective by using classifiers trained on sample corpora in a system that filters and ranks search results from bibliographic databases such as Pubmed. This kind of application is known as *document filtering*. The filtering part is essentially a binary text categorization problem. Ranking comes for free in conjunction with distribution classifiers because they return probabilities that can be used as document scores. The vast body of literature on automated text categorization is surveyed in [8]. In the Information Retrieval community, much work (from [6] to [1, 3, 4, 10, 15]) has been done on *adaptive* document filtering, where relevance feedback from users is

[1] http://rna.tbi.univie.ac.at

Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics, held in conjunction with ECML/PKDD. 2003, Dubrovnik, Croatia

Table 1: The training corpora.

Corpus	Source	Size	Positive
picorna	Pubmed query: picornavirus RNA secondary structure	40	68%
picorna2	picorna + 24 extra documents	64	58%
flavi	Pubmed query: RNA AND (IRES OR "secondary structure" OR "conserved structure" OR "5'utr" OR "3'utr" OR "coding region") AND ("hepatitis C virus" OR "hepatitis G virus" OR pestivirus OR dengue OR "japanese encephalitis virus" OR "yellow fever virus"OR "tick-borne encephalitis virus")	153	8%
flavi2	flavi + 34 extra documents	187	12%
hepadna	Pubmed query: (Hepadnaviridae OR "Hepatitis B" OR "HBV") AND (RNA secondary structure) NOT delta	16	69%

employed to adjust document filters.

The preliminary results presented here indicate that a classifier trained on one virus group can be applied successfully to search the literature on other virus groups. Therefore, a system for supporting bibliographic search based on automated text categorization seems feasible for our target topic.

The remainder of this article is organized as follows: in Sec. 2 we present the data sets used for this work. The methods and tools used are described in Sec. 3. Our experiments and their results are presented in Sec. 4. Finally we give in Sec. 5 a conclusion and outline our future research.

2. DATA SETS

Training data has been obtained from searching the Pubmed collection via the entrez interface[2] and then downloading the referenced articles as PDF documents (as far as available). The search queries (see Table 1) have been specified by our domain experts (PFS, CT, CW). The resulting corpora are referred to as picorna, flavi, and hepadna. They are dedicated to the virusgroups *Picornaviridae*, *Flaviviridae*, and *Hepadnaviridae*, respectively.

Since corpus picorna is quite small and corpus flavi contains only few positive examples, we decided to add more documents. These documents were provided by our domain experts from their private bibliographical collections. The resulting corpora are referred to as picorna2 and flavi2. The small corpus hepadna is only used for testing classifiers trained on the latter two corpora.

A document is considered a positive example within its corpus if it contains information on the secondary structure of the RNA of viruses belonging to the virus group the corpus is dedicated to.

[2]http://www.ncbi.nlm.nih.gov/Entrez/

3. METHODS

3.1 Data Preparation

The PDF documents where converted into text using the Unix tools pdftotext and ps2ascii. The ConceptComposer text analysis suite [2] was used to build a full text index of the resulting text documents in a relational database (mysql).

Based on this index, the documents were transformed into vector representation using a SQL script. We computed term weights according to the standard tfidf method (see e.g. [7]). Each corpus is stored in a separate mysql database.

For feature selection we implemented the term relevance measures *Odds Ratio* and *Mutual Information* (see [8]). In addition we implemented derived term relevance measures where the original relevance value for a term is weighted with its frequency in the test database that is used for evaluation.

3.2 Text Categorization

We built the Java application litsift on top of the Weka 3 machine learning software [13] to classify the document corpora. This enabled us to experiment with the variety of classifiers provided by Weka. Further parameters that can be varied are

- the term relevance measure to use for feature selection
- the number of features to be taken into account
- the target recall when evaluating a classifier on the test corpus
- classifier specific parameters

The application reads class labels for documents and their term weights for the selected features from the training database and creates a set of Weka instances from it. This instance set is either used for cross evaluation on the training corpus or it is used to train a classifier that is evaluated on a separate test corpus. In the latter case, only those documents are classified as positive whose predicted class-membership probability exceeds a certain threshold. This threshold is adjusted automatically to achieve at least the chosen target recall (if possible at all) in a trade-off with the achieved precision. The threshold is found by computing histograms on the number of positives and true positives over the predicted probabilities.

4. RESULTS

Before we assess the applicability of classifiers trained on one corpus to another corpus, we present cross-evaluation results on each corpus as a base line for comparison.

4.1 Feature Selection

To assess the performance of different term relevance measures, we varied the number N of features. From the corpus we filtered those documents that contained at least one of the N best terms of the chosen measure. Then we computed precision and recall of this filter by counting the selected documents as positives and the rest of the corpus as negatives. The results are shown in Table 2. It shows that 10–30 features are always sufficient to retrieve all positive examples. Moreover it shows that the corpora picorna and picorna2 are quite trivial since they can be classified completely and correctly by using just the first 20 (picorna) or 30 (picorna2) features selected by Mutual Information.

Table 2: Filtering results for different corpora, and relevance measures (column "msr"), with target recall 100%. The relevance measures Mutual Information and Odds Ratio are abbreviated as "MI" and "OR", respectively. Column "$p_{\mathbf{avg}}$" shows the average precision over all feature counts where the target recall is exceeded. Column "$p_{\mathbf{max}}$" shows the maximum precision. The minimum feature count at which this maximum precision is reached is labeled"d". The recall achieved with this number of features is shown in column "r".

corpus	msr	p_{avg}	p_{max}	r	d
flavi	MI	11.2%	23.1%	100.0%	20
flavi	OR	7.8%	7.9%	100.0%	10
flavi2	MI	20.2%	40.7%	100.0%	20
flavi2	OR	11.8%	11.8%	100.0%	10
picorna	MI	76.7%	100.0%	100.0%	20
picorna	OR	67.6%	69.2%	100.0%	10
picorna2	MI	69.3%	100.0%	100.0%	30
picorna2	OR	58.0%	59.7%	100.0%	10

4.2 Cross Evaluation on Each Corpus

As a base line for comparison we cross-evaluated several classifiers from the Weka toolkit, namely C4.5 ("J48"), Support Vector Machine ("SMO"), and Naive Bayes ("N.B."), in combination with the available term relevance measures on each corpus. The results are shown in Table 3. It shows that

1. on flavi and flavi2 the target recall of 80% can be reached only by the NaiveBayes classifier
2. with few exceptions, less than 50 features are needed to achieve maximum recall
3. corpora picorna and picorna2 can be almost perfectly classified in most cases
4. J48 seems sensitive with respect to the relevance measure: on flavi2, Odds Ratio performs much better, on picorna2, Mutual Information performs much better.

4.3 Validation on a Separate Test Corpus

We first present some exemplary experiments with SMO and then give in an overview of all experiments in form of a table.

4.3.1 *Training on* flavi, *Validation on* picorna

A SMO classifier trained on flavi with Odds Ratio measure evaluated on picorna2 reaches the target recall of 80% beginning with 30 features. The precision reaches a maximum of 80% at about 150 features (see Fig. 1a). Using Mutual Information yields similar results.

4.3.2 *Training on* flavi2, *Validation on* picorna2

Compared to Sec. 4.3.1, the average precision of SMO drops slightly from 74% to 66% (see Fig. 1b) which is still quite acceptable for bibliographic search.

Table 3: Cross evaluation results for different corpora, classifiers, and relevance measures, with target recall 80%. The classifiers shown in column "class" are C4.5 ("J48"), Support Vector Machine ("SMO"), and Naive Bayes ("N.B."). For the meaning of the remaining columns see Table 2. The average precision is ommitted in cases where the target recall was not reached.

corpus	class	msr	p_{avg}	p_{max}	r	d
flavi	J48	MI	–	85.7%	60.0%	10
flavi	J48	OR	–	72.7%	66.7%	60
flavi	N.B.	MI	21.3%	38.5%	83.3%	30
flavi	N.B.	OR	25.2%	44.0%	91.7%	40
flavi	SMO	MI	–	75.0%	30.0%	10
flavi	SMO	OR	–	80.0%	33.3%	30
flavi2	J48	MI	–	73.7%	63.6%	160
flavi2	J48	OR	–	77.3%	77.3%	30
flavi2	N.B.	MI	28.3%	41.9%	81.8%	80
flavi2	N.B.	OR	37.8%	58.1%	81.8%	30
flavi2	SMO	MI	–	66.7%	54.5%	30
flavi2	SMO	OR	–	73.3%	50.0%	40
picorna	J48	MI	99.3%	100.0%	100.0%	10
picorna	J48	OR	89.2%	92.6%	92.6%	50
picorna	N.B.	MI	79.3%	100.0%	95.2%	10
picorna	N.B.	OR	80.5%	100.0%	100.0%	20
picorna	SMO	MI	90.6%	100.0%	100.0%	10
picorna	SMO	OR	91.7%	100.0%	85.2%	30
picorna2	J48	MI	95.3%	100.0%	100.0%	10
picorna2	J48	OR	85.2%	88.2%	81.1%	110
picorna2	N.B.	MI	77.6%	100.0%	96.7%	10
picorna2	N.B.	OR	79.7%	100.0%	94.6%	20
picorna2	SMO	MI	93.5%	100.0%	100.0%	10
picorna2	SMO	OR	93.1%	100.0%	81.1%	40

4.3.3 *Training on* picorna, *Validation on* flavi

While SMO trained on corpus flavi can be successfully applied to the corpora picorna and picorna2, the inverse setting is not as successful. At 80% recall, SMO achieves a maximum precision of 23% precision at 60 features (see Fig. 2a).

With Mutual Information, the precision is even lower (about 10%).

Using a derived term relevance measure (Odds Ratio, weighted with term frequencies from flavi) did not yield any improvement, either.

23% precision may not seem high, but in our application to bibliographic search it is still more tolerable than in other fields of text classification.

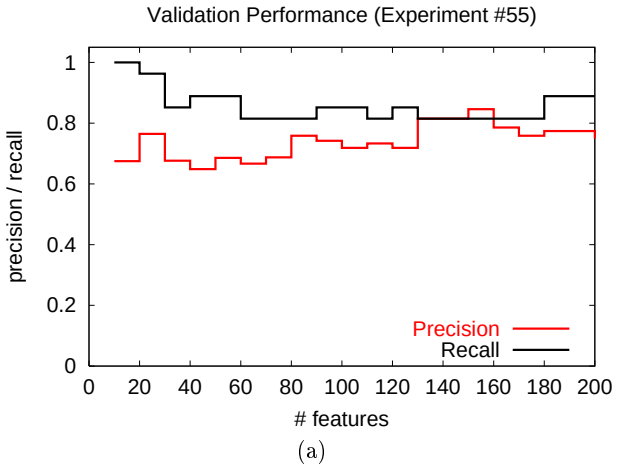
4.3.4 *Training on* picorna2, *Validation on* flavi2

Compared to Sec. 4.3.3, precision of SMO increases to 30% starting from 40 features (see Fig. 2b).

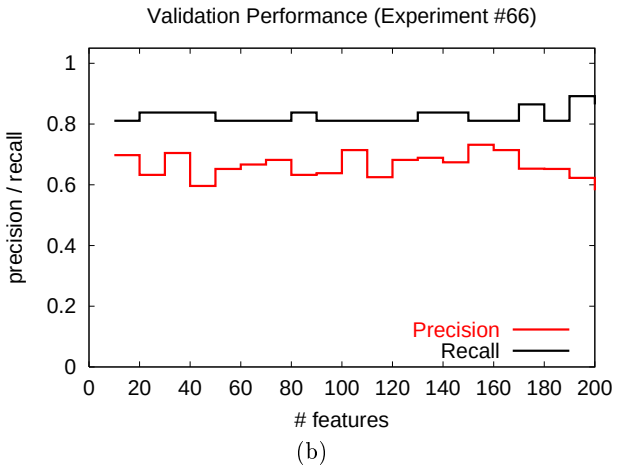
4.3.5 *Discussion*

In Table 4, all experiments with evaluation on a separate corpus are listed. We may summarize these results as follows:

1. Corpora picorna and picorna2 can quite successfully be classified after training on flavi and flavi2, respectively.

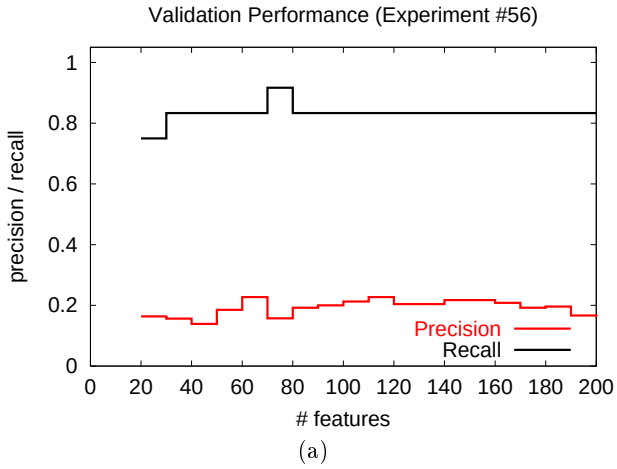


(a)

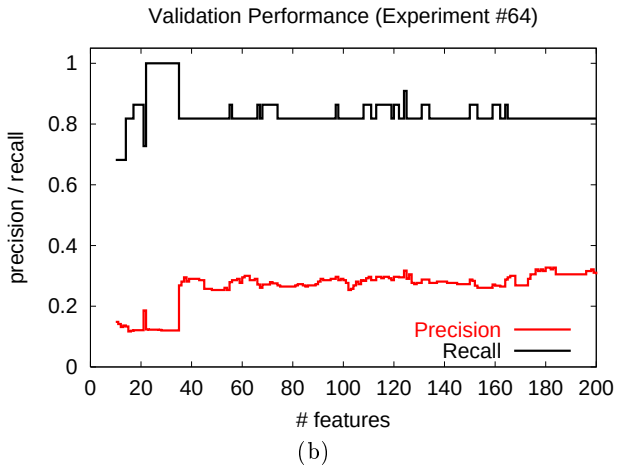


(b)

Figure 1: Performance for Weka SMO with Odds Ratio, target recall 80%: (a) on corpus picorna after training on corpus flavi, (b) on corpus picorna2 after training on corpus flavi2.



(a)



(b)

Figure 2: Performance for Weka SMO with Odds Ratio, target recall 80%: (a) on corpus flavi after training on corpus picorna, (b) on corpus flavi2 after training on corpus picorna2.

   (a) With J48 or NaiveBayes, 100% recall can be achieved with maximum precisions above 70%, using only few features (10–30).

   (b) Mutual Information seems to perform better than Odds Ratio.

2. Corpora flavi and flavi2 can not as easily classified after training on picorna and picorna2, respectively.

   (a) The best maximum precision is achieved by SMO with Odds Ratio

   (b) Corpus flavi2 is easier to classify than flavi

3. Corpus hepadna can quite successfully be classified after training on picorna2 or flavi2.

   (a) In both cases, SMO performs best, reaching a maximum precision of 90%.

   (b) In most cases Odds Ratio performs much better than Mutual Information, i.e., it needs much fewer features to achieve a better maximum precision.

4. In most cases the difference between average and maximum precision is quite small. This supports the observation from Figs. 1 and 2 that precision does not depend too much on the number of features.

The asymmetry between the picorna* and flavi* corpora can to some extent be explained by the fact that the *Flaviviridae* virus group is more heterogenous than the *Picornaviridae* group. For instance, while all *Picornaviridae* genomes have so-called IRES (Internal Ribosomal Entry Site) regions, this does not hold for all *Flaviviridae*. This means that a classifier trained on a picorna* corpus only finds those positive examples in flavi* that are similar to those in the training corpus. In the other direction this partition within a flavi* corpus seems to be sufficient to learn the characteristics of the positive examples in the picorna* corpora. The additional positives in corpus flavi2 might be more "picorna"-like which would explain the better performance when testing on flavi2 instead of flavi.

Table 4: Transfer results for different training and test corpora, classifiers, and relevance measures, with target recall 80%. The classifiers shown in column "class" are C4.5 ("J48"), Support Vector Machine ("SMO"), and Naive Bayes ("N.B."). For the meaning of the remaining columns see Table 2.

training	test	class	msr	p_{avg}	p_{max}	r	d
flavi	picorna	J48	MI	69.1%	76.7%	100.0%	30
flavi	picorna	J48	OR	67.3%	67.5%	100.0%	10
flavi	picorna	N.B.	MI	69.1%	76.7%	100.0%	30
flavi	picorna	N.B.	OR	67.5%	67.5%	100.0%	10
flavi	picorna	SMO	MI	74.4%	80.6%	92.6%	160
flavi	picorna	SMO	OR	74.0%	84.6%	81.5%	150
flavi2	picorna2	J48	MI	65.8%	100.0%	80.0%	10
flavi2	picorna2	J48	OR	57.8%	57.8%	100.0%	10
flavi2	picorna2	N.B.	MI	65.9%	83.3%	100.0%	10
flavi2	picorna2	N.B.	OR	57.8%	57.8%	100.0%	10
flavi2	picorna2	SMO	MI	68.3%	83.3%	100.0%	10
flavi2	picorna2	SMO	OR	66.2%	73.2%	81.1%	150
picorna	flavi	J48	MI	12.6%	16.4%	91.7%	90
picorna	flavi	J48	OR	13.7%	15.7%	91.7%	60
picorna	flavi	N.B.	MI	9.5%	14.7%	83.3%	10
picorna	flavi	N.B.	OR	9.4%	12.2%	100.0%	30
picorna	flavi	SMO	MI	12.3%	20.0%	83.3%	110
picorna	flavi	SMO	OR	18.6%	22.7%	83.3%	60
picorna2	flavi2	J48	MI	15.1%	22.6%	86.4%	130
picorna2	flavi2	J48	OR	16.3%	19.3%	100.0%	40
picorna2	flavi2	N.B.	MI	16.1%	20.0%	100.0%	20
picorna2	flavi2	N.B.	OR	14.0%	15.4%	95.5%	180
picorna2	flavi2	SMO	MI	18.7%	23.8%	86.4%	130
picorna2	flavi2	SMO	OR	26.2%	32.7%	81.8%	180
flavi2	hepadna	J48	MI	78.4%	81.8%	81.8%	180
flavi2	hepadna	J48	OR	68.8%	71.4%	90.9%	20
flavi2	hepadna	N.B.	MI	78.4%	81.8%	81.8%	180
flavi2	hepadna	N.B.	OR	68.8%	68.8%	100.0%	10
flavi2	hepadna	SMO	MI	75.0%	75.0%	81.8%	200
flavi2	hepadna	SMO	OR	73.6%	90.9%	90.9%	50
picorna2	hepadna	J48	MI	76.6%	83.3%	90.9%	90
picorna2	hepadna	J48	OR	70.1%	71.4%	90.9%	40
picorna2	hepadna	N.B.	MI	76.6%	83.3%	90.9%	90
picorna2	hepadna	N.B.	OR	69.6%	75.0%	81.8%	170
picorna2	hepadna	SMO	MI	76.7%	81.8%	81.8%	90
picorna2	hepadna	SMO	OR	77.9%	90.0%	81.8%	70

4.3.6 *Usefulness*

How useful could a bibliographic search tool based on automated classification be for a scientist who wants to perform a literature recherche? To assess this, we consider the following scenario: the scientist wants to identify relevant literature with minimal effort without loosing to many relevant articles. For a fixed recall r, the amount of work is determined by the number of articles that the scientist has to inspect.

We assume a fixed corpus of articles that have been returned by the bibliographic database (i.e., Pubmed). By randomly selecting documents with a probability r, we achieve also recall r since the probability for a relevant document to be selected is r. In this case, the scientist has to inspect a fraction $P_{\text{rand}} = r$ of all documents. This is the baseline for a comparison with an automated classifier.

The fraction P_{auto} of documents selected by an automated classifier with precision p and recall r on a corpus with a fraction c of relevant documents is $P_{\text{auto}} = cr/p$. Hence the work is reduced by the factor $s = P_{\text{auto}}/P_{\text{rand}} = c/p$. In Table 5 the work reduction s is shown for some of the cases from Table 4. Most work can be saved on corpora such as flavi2 with few relevant documents (assuming that the precision does not deteriorate too much). The percentage of relevant documents in the small corpora picorna2 and hepadna is too high to reach large work reductions.

Table 5: Work reduction s for selected sample configurations.

training	test	class	msr	p_{max}	r	s
flavi2	picorna2	SMO	MI	83.3%	100.0%	70%
picorna2	flavi2	SMO	OR	32.7%	81.8%	37%
flavi2	hepadna	SMO	OR	90.9%	90.9%	75%
picorna2	hepadna	SMO	OR	90.0%	81.8%	75%

5. CONCLUSION AND OUTLOOK

The results presented in this study are rather heterogeneous. Nevertheless, they indicate that classifiers trained on one subtopic can be applied to another subtopic and achieve precisions (here 20% – 100%) that will result in cost savings when searching for relevant literature while not too many (here 20%) relevant documents are lost.

The complications of bibliographic search that plague the case of RNA secondary structure features in viral RNAs are not a restricted to this particular topic. Whenever the available literature has to be searched for information that is rarely the main focus of the publication keyword-based searches tend to have either low recall or low precision. Regulatory sequences associated with certain classes of genes may serve as another example.

We thus plan to extend the litsift application into a bibliographic search tool that sends a user query to a bibliographic database such as Pubmed, retrieves the search results and the articles cited therein, and ranks the results according to the predictions of a classifier previously trained using the same tool. The user may choose to re-label some of the results manually and retrain the classifier in order to enhance its performance. An interesting option for further improving this tool would be to include classification techniques that take unlabeled data into account, e.g. [5, 12].

Open questions that require further research are: (i) what are good heuristics for choosing a number of features and (ii) are there indicators for the transferability of a classifier to another corpus?

6. ACKNOWLEDGEMENTS

This work is supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung*, Project Nos. P-13545-MAT and P-15893 and the German *DFG* Bioinformatics Initiative.

7. REFERENCES

[1] G. Amati and F. Crestani. Probabilistic learning for selective dissemination of information. *Information Processing and Management*, 35(5):633–654, 1999.

[2] G. Heyer, U. Quasthoff, and C. Wolff. Automatic analysis of large text corpora — A contribution to structuring WEB communities. In H. Unger, T. Böhme, and A. Mikler, editors, *Innovative Internet Computing Systems*, volume 2346 of *Lecture Notes in Computer Science*, pages 15–26. Springer-Verlag, Heidelberg, 2002.

[3] R. D. Iyer, D. D. Lewis, R. E. Schapire, Y. Singer, and A. Singhal. Boosting for document routing. In A. Agah, J. Callan, and E. Rundensteiner, editors, *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 70–77, McLean, US, 2000. ACM Press, New York, US.

[4] Y.-H. Kim, S.-Y. Hahn, and B.-T. Zhang. Text filtering by boosting naive Bayes classifiers. In N. J. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 168–175, Athens, GR, 2000. ACM Press, New York, US.

[5] K. Nigam. *Using Unlabeled Data to Improve Text Classification*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, US, 2001.

[6] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART retrieval system: experiments in automatic document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, USA, 1971.

[7] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[8] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[9] R. Stocsits, I. L. Hofacker, and P. F. Stadler. Conserved secondary structures in hepatitis B virus RNA. In *Computer Science in Biology*, pages 73–79, Bielefeld, D, 1999. Univ. Bielefeld. Proceedings of the GCB'99, Hannover, D.

[10] D. R. Tauritz, J. N. Kok, and I. G. Sprinkhuizen-Kuyper. Adaptive information filtering using evolutionary computation. *Information Sciences*, 122(2/4):121–140, 2000.

[11] C. Thurner, C. Witwer, I. Hofacker, and P. F. Stadler. Conserved RNA secondary structures in Flaviviridae genomes. 2003. submitted.

[12] J.-N. Vittaut, M.-R. Amini, and P. Gallinari. Learning classification with both labeled and unlabeled data. *Lecture Notes in Computer Science*, 2430:468–479, 2002.

[13] I. H. Witten and E. Frank. Nuts and bolts: Machine learning algorithms in java,. In *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.*, pages 265–320. Morgan Kaufmann, 1999.

[14] C. Witwer, S. Rauscher, I. L. Hofacker, and P. F. Stadler. Conserved RNA secondary structures in Picornaviridae genomes. *Nucl. Acids Res.*, 29:5079–5089, 2001.

[15] K. L. Yu and W. Lam. A new on-line learning algorithm for adaptive text filtering. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, and L. Bouganim, editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 156–160, Bethesda, US, 1998. ACM Press, New York, US.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. 2003, Dubrovnik, Croatia

Fringe SVM Settings and Aggressive Feature Reduction

Adam Kowalczyk ADAM.KOWALCZYK@TEAM.TELSTRA.COM
Bhavani Raskutti BHAVANI.RASKUTTI@TEAM.TELSTRA.COM
Telstra Corporation, 770 Blackburn Road, Clayton, Victoria, Australia

Abstract

Statistical techniques for aggressive feature reduction are studied on data obtained in a gene knock-out experiment. The essential part of the process is automatic assessment of the quality of various feature selection methods. This is done by comparison of the performance of discriminating models built on candidate subsets of features. Experiments show that typical settings of popular 2-class discriminators, support vector machines (SVM), cannot be used as they produce models of very poor quality. The proposed way around is to use "fringe classifiers" such as SVMs trained on positive class data only or class centroids. Additionally, we also use models generated by such algorithms directly for identification of most discriminating features. We recommend that such simple machine learning techniques should be included into a repertoire of discriminators used on such occasions. We show that such relatively superior performance of fringe SVMs can also be observed on regular text-mining data bases, such as Reuters newswire benchmark, if only the less frequent features (words) are used.

1. Introduction

As our knowledge of genome of various organisms increases, the number of potentially relevant scientific articles proliferates, making it impossible for an individual to keep up. This creates a need for statistical tools capable of pre-filtering and identifying useful information prior to human inspection, thus alleviating the information overload.

Traditional statistical tools typically deal with situations when there are more "training " examples than potentially predictive features to consider. However, in biological domains, it can be expensive to perform laboratory tests to obtain training data, while it may be easy to obtain millions of potentially relevant features. Solving problems in such domains would therefore require methods that robustly deal with (1) extremely high dimensional spaces with only a handful of "training " examples and (2) "low" relevance of available features to any particular problem.

As a first step towards development of such methods, we explore, in this paper, the use of support vector machines (SVM) in many different modes. We consider, in particular, *fringe classifiers* that correspond to some extreme settings of parameters, e.g. learning from examples of a single class and settings that provide solutions that are equivalent to learning from data centroids. We show that for the 2002 KDD cup data (*AHR-data*) [4], these fringe classifiers provide robust discriminating models, which are far more accurate than typical 2-class SVMs. Additionally, we show that the phenomenon of domination of fringe classifiers is not unique to AHR-data, but is observable in the popular Reuters Newswires text mining benchmark.

We investigate the utility of such classifiers along with other discriminating models for feature selection. We use these classifiers, firstly for the evaluation of feature selection methods based on their accuracy of prediction, and secondly, for directly selecting informative features using the generated models. Our investigation shows that such model-based selection can both provide accurate classifiers and select a small subset of features which may be used in a variety of ways: (1) inspection by researchers for identification of biological mechanisms underlying an investigated phenomenon, (2) generation of key words for retrieval of relevant scientific articles, and (3) development of a more refined and simpler to interpret discriminating models.

The paper is organised as follows. Section 2 introduces the machine learning algorithms and the performance measure used in this research. Sections 3 and 4 present our experiments with AHR-data and Reuters data, respectively. In Section 5, we discuss the implications of our results and present some intuitive explanations of

the observed phenomena.

2. Classifiers and Metrics

In this section we introduce the basic machine learning algorithms used in this paper. We focus on linear classifiers, in particular, on Support Vector Machines (SVM). There are a number of reasons for this focus. Firstly, SVMs have been top performers in text [9, 16] and biomining tasks. Secondly, they have been top performers on AHR-data: 3 out of the top 5 submissions to KDD Cup 2002 were based on SVMs [6, 13, 11]. Finally and most importantly, SVMs are well suited to process sparse, high dimensional data.

Our classification problem is formulated as follows. Given a training sequence (x_i, y_i) of binary n-vectors $x_i \in \{0,1\}^n \subset \mathbb{R}^n$ and bipolar labels $y_i \in \{\pm 1\}$ for $i = 1, ..., m$. The case of prime interest here is when the target class, labelled $+1$, is much smaller than the background class (labelled -1), $\approx 1\%$ of the data. Our aim is to find a "good" discriminating linear function

$$f(x) := w \cdot x + b \tag{1}$$

that scores the target class instances higher than the background class instances. Here $x \in \mathbb{R}^n$, "$\cdot$" denotes the dot product in $\mathbb{R}^n$ and $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ is defined by one of the five learning algorithms described below.

The first four algorithms, requiring dedicated solvers, are versions of the popular SVMs. For all of them the solution $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ is defined as a minimiser of the regularised risk functional of the following form.

$$\|w, b\|_*^2 + \sum_{i=1}^{m} C_i \phi\big(1 - y_i(w \cdot x_i + b)\big), \tag{2}$$

where $\| \ \|_*^2$ is a squared "norm" penalising for the "complexity of the classifier", $\phi : \mathbb{R} \to \mathbb{R}_+$ is a convex loss function penalising for deviations of the machine from allocated labels and the regularisation constants are defined as follows:

$$C_i = \begin{cases} (1+B)C/(2m_+) & \text{if } y_i = +1, \\ (1-B)C/(2m_-) & \text{if } y_i = -1, \end{cases} \tag{3}$$

for $i = 1, ..., m$, where $C > 0$, m_+ and m_- denote the numbers of examples with labels $y_i = +1$ and $y_i = -1$, respectively. Here $-1 \le B \le 1$ is *a balance parameter* designed to balance the impact of instances from the positive and the negative class, respectively.

Now we specify four variations of the regularised risk (2) leading to four different machines to be used in this paper.

Algorithm 1, SVM^1_{BC}: This is the popular *SVM with linear penalty*. Here, we use the norm $||w, b||_*^2 := ||w||^2 = w \cdot w$ and the "hinge loss" $\phi(\theta) := \max(0, \theta)$, $\theta \in \mathbb{R}$ [5, 19, 20];

Algorithm 2, $hSVM^1_{BC}$: Replacing the norm in the above definition by

$$||w, b||_*^2 := ||w||^2 + b^2 \tag{4}$$

we obtain *the homogeneous SVM with linear penalty*;

Algorithm 3, $hSVM^2_{BC}$: For *the (homogeneous) SVM with quadratic penalty* [5] we use norm (4) and the squared hinge loss $\phi(\theta) := (\max(0, \theta))^2$ for $\theta \in \mathbb{R}$;

Algorithm 4, hRN^2_{BC}: For *the homogeneous regularisation network* [7, 21] or *the ridge regression* [5, 7, 21] we use norm (4) and ordinary square loss $\phi(\theta) := (\theta)^2$ for $\theta \in \mathbb{R}$.

Algorithm 5, $Cntr_B$: This is the simplest of the five algorithms. Here, we set $b := 0$ and

$$w := \frac{1+B}{2m_+} \sum_{i, y_i = +1} x_i - \frac{1-B}{2m_-} \sum_{i, y_i = -1} x_i.$$

For $B = +1$ vector w is exactly the centroid of the minority (the target) class, for $B = -1$ it is the centroid of the majority (the background) class while for $B = 0$ it is half of the difference between the centroids of the two classes.

Note that $Cntr_B$ can be viewed as the "limit" case of the SVM, in the sense that $Cntr_B = \lim_{C \to 0^+} \frac{SVM^p_{BC}}{C}$, for $p = 1, 2$. We refer to [12] for further discussion of this link.

2.1 1-class SVMs

Note that $hSVM^1_{BC}$, $hSVM^2_{BC}$ and hRN^2_{BC} implement classifiers that correspond to separation of the data $(x_i, 1, y_i) \in \mathbb{R}^n \times \mathbb{R} \times \{\pm 1\}$ by a hyperplane $\langle (w, b), (x, 1) \rangle = 0$ passing through the origin $(0, 0) \in \mathbb{R}^n \times \mathbb{R}$. One thing to stress is that (2) provides a non-trivial (i.e. $\neq$constant), unique classifier (1) in all "regular" cases of interest, in particular, for $B = \pm 1$ if at least one $C_i \neq 0$ and $(0, 0) \in \mathbb{R}^n \times \mathbb{R}$ does not belong to the convex shell spanned by all vectors $y_i(x_i, 1) \in \mathbb{R}^N \times \mathbb{R}$. These cases of B are equivalent to learning from data belonging to a single class label, the target class $y_i = +1$ for $B = +1$ and the background class $y_i = -1$ for $B = -1$. We shall call such machines the *1-class SVM's*.

On the level of the extended feature space $\mathbb{R}^n \times \mathbb{R}$, any $hSVM^1_{BC}$, $hSVM^2_{BC}$ or hRN^2_{BC} can be reduced to a single class machine. In fact, we can always absorb the signum y_i by considering the data $(\tilde{z}_i, \tilde{y}_i) :=$

$(y_i x_i, y_i, 1)$ rather than $(x_i, 1, y_i)$ and then minimising the following functional equivalent to (2):

$$\langle \tilde{w}, \tilde{w} \rangle + \sum_{i=1}^{m} C_i \phi \left(1 - \langle \tilde{w}, \tilde{z}_i \rangle\right) \tag{5}$$

where $\langle\ ,\ \rangle$ stands for the dot product in $\mathbb{R}^n \times \mathbb{R}$. This formally reduces the two class problem (2) to "single class learning". In the case of $hSVM^1$, the solution to (5) can be found using SVM^1 if an extra point, namely $(0,0) \in \mathbb{R}^N \times \mathbb{R}$, with the opposite label -1, is added to the data. Such a method for one-class learning has been considered previously in [14, 18].

We shall refer to any $hSVM^p_{\pm 1,C}$, $p = 1, 2$, $hRN^2_{\pm 1,C}$ or $Cntr_{\pm 1}$ as *fringe SVMs*. We include here $Cntr_{\pm 1}$ since the SVM solution for low C approaches the solution of $Cntr_B$ (cf. [12]). This equivalence also motivates our setting of C to 5000 (the "hard margin" case) for our feature selection experiments, since the low values of C are covered by the centroid solution.

2.2 Performance measures

We have used $AROC$, the Area under the Receiver Operating Characteristic (ROC) curve as our main performance measure. In that, we follow the steps of KDD 2002 Cup, but also, we see it as the natural metric of general goodness of classifier (as corroborated below) capable of meaningful results even if the target class is a tiny fraction of the data.

We recall that the ROC curve is a plot of the *true positive rate* or precision, $P(f(x_i) > \theta | y_i = 1)$, against the *false positive rate*, $P(f(x_i) > \theta | y_i = -1)$, as a decision threshold θ is varied. The concept of ROC curve originates in signal detection but these days it is widely used in many other areas, including data mining, psychophysics and medical diagnosis (cf. review [2]). In the latter case, $AROC$ is viewed as a measure of general "goodness" of a test, formalised as a predictive model f in our context, with a clear statistical meaning as follows. $AROC(f)$ is equal to the probability of correctly answering the two-alternative-forced-choice problem: given two cases, one x_i from the negative and the other x_j from the positive class, allocate scores in the right order, i.e. $f(x_i) < f(x_j)$. Additional attraction of $AROC$ as a figure of merit is its direct link to the well researched area of order statistics, via U-statistics and Wilcoxon-Whitney-Mann test [1].

There are some ambiguities in the case of $AROC$ estimated from a discrete set in the case of ties, i.e. when multiple instances from different classes receive the same score. Following [1] we implement in this paper the definition

$$\begin{aligned} AROC(f) = P(f(x_i) < f(x_j) | - y_i = y_j = 1) \\ +0.5 P(f(x_i) = f(x_j) | - y_i = y_j = 1) \end{aligned}$$

expressing $AROC$ in terms of conditional probabilities.

3. Analysis of AHR-data

In our main experiments we have used AHR-data set which is the combined training and test data sets used for task 2 of KDD Cup 2002. The data set is based on experiments by Guang Yao and Chris Bradfield of McArdle Laboratory for Cancer Research, University of Wisconsin. These experiments aimed at identification of yeast genes that, when knocked out, cause a significant change in the level of activity of the Aryl Hydrocarbon Receptor signalling pathway (cf. [4] for more details). In this paper we follow the setting of the "broad task" of the KDD Cup: the discrimination between 127 'positive' genes from the combined class encompassing the labels "change" and "control" and the remaining 4380 genes forming the 'negative' class. In our experiments this set has been repeatedly split into 70% for training and 30% for testing. All averages and standard deviations reported are for independent tests on 20 such random splits.

3.1 Data representation

Each training and test gene was represented by a vector of binary attributes extracted from the very rich data sources provided: function/localisation annotations, protein-protein interactions and Medline abstracts.

Hierarchical information about function, protein classes and localisation was converted to a vector per gene. For instance, the following two entries in the file function.txt

YGR072W CYTOPLASM | SUBCELLULAR LOCALISATION
YGR072W NUCLEUS | SUBCELLULAR LOCALISATION

yielded three function attributes: "cytoplasm", "subcellular localisation" and "nucleus" each with a value of 1 for the gene "YGR072W". This processing created 409 attributes: 213 for gene function, 154 for protein classes and 42 for localisation.

Textual information from all abstracts associated with a gene was converted to 'word token' presence vectors ('a bag of words'). A 'word token', in this context, is any string of alphanumeric characters, which may or may not correspond to an ordinary word. Word tokens corresponding to words in a standard list of stop words, such as "the", "a" and "in", have been

excluded. All ordinary words were stemmed using a standard Porter stemmer. The resulting word token attributes were then checked against the gene alias file, and all aliases were replaced by a single gene name.

The above abstract processing resulted in 48,089 word token attributes. Around 3/4th of these attributes were subsequently eliminated by discarding all those that occurred in only one gene, and by discarding all those which had a total frequency that was greater than one standard deviation from the mean. After this processing, we were left with 16,474 attributes from the abstracts.

The gene-gene interaction file is symmetric. Hence, each entry in the file interaction.txt creates two attributes. For instance, the entry, "YFL039C YMR092C" creates two interaction attributes: "YFL039C" and "YMR092C", and the attribute "YFL039C" is set to 1 for the gene "YMR092C" and vice-versa. Processing of the gene interactions file yielded a total of 1,447 attributes.

Thus, the total number of binary attributes used by the learning algorithm was 18,330 (= 409 + 16,474 + 1,447).

3.2 Feature selection

We now explore the utility of the classifiers for feature selection as (i) techniques for evaluation of various selected feature sets and (ii) tools for selection of such sets.

We investigate several strategies for scoring features: the first four score features based on their distribution in the training set (the filter approach as defined in [10]), while the others are based on the SVM models generated (w). In all cases, the computed score is used to sort the features so that the most informative features may be selected.

A: DocFreq (Document frequency thresholding): This method has its origins in information retrieval [17] and is based on the supposition that rare features are not informative for predicting classes. In this case the score of a feature is simply the number of instances where it has been equal to 1.

B: ChiSqua (χ^2): The χ^2 measures the lack of independence between a feature and a class of interest. First, for each feature and each class, i.e. $y = \pm 1$, a score is computed on the basis of the two-way contingency table [22].

C: MutInfo: (Mutual Information): The score is allocated to a feature on the basis of the joint and marginal probabilities of its usage estimated from the training set [22].

D: InfGain: (Information gain): This is frequently employed as a term goodness measure in machine learning [15], and measures the number of bits of information obtained for class prediction by knowing the presence or absence of a term in an instance.

E-J: Model-based feature selection: In this case, the score of a feature is simply the magnitude of the weight allocated to it by a linear model generated to discriminate two classes of interest, i.e. corresponding entries in the vector $w = (w_1, ..., w_n)$. Thus, our method is much simpler and easier to implement than the sophisticated SVM-model-based selection described in [8].

Our model-based approach is also different from the wrapper approach used in [10] in that we do not restrict the use of the selected features for the particular learning algorithm. Instead, the chosen features are then used as features for evaluating all algorithms.

In our experiments we employ a small variation: we average w over 20 models generated for 20 random stratified splits of the data into 70% training and 30% test sets (instances of both classes were split independently in the indicated proportions). We have used three learning machines, $hSVM^1_{B,5000}$(E,F), $hSVM^2_{B,5000}$(G,H) and $Cntr_B$(I,J) in two modes: positive 1-class mode $B = +1$ (E,G,I) and balanced 2-class mode, $B = 0$ (F,H,J), thus yielding six additional methods.

Figure 1 shows the results of evaluation of the ten feature selection techniques (columns A-J) by four different algorithms (rows 1-4). Each evaluation technique has been used in two modes: positive 1-class and balanced 2-class. (We have not shown evaluation results for SVM^1 as they are very similar to those for 2-class $hSVM^1$.) The results can be summarised as follows:

1. As a general rule, for all SVMs, 1-class models perform much better than 2-class models when using the same set of features. In addition, these two modes of SVM often give quite opposite evaluation of the utility of selected features (the notable exception being column H). While 1-class finds them informative ($AROC > 0.5$), 2-class finds them detrimental with $AROC < 0.5$, i.e. below that of the random classifier.

2. $DocFreq$ and $MutInfo$ both provide very poor results for low number of features, although they use completely different metrics for scoring. $MutInfo$ is strongly influenced by the marginal probability of terms and tends to favour rare terms, while $DocFreq$ selects the most common terms.

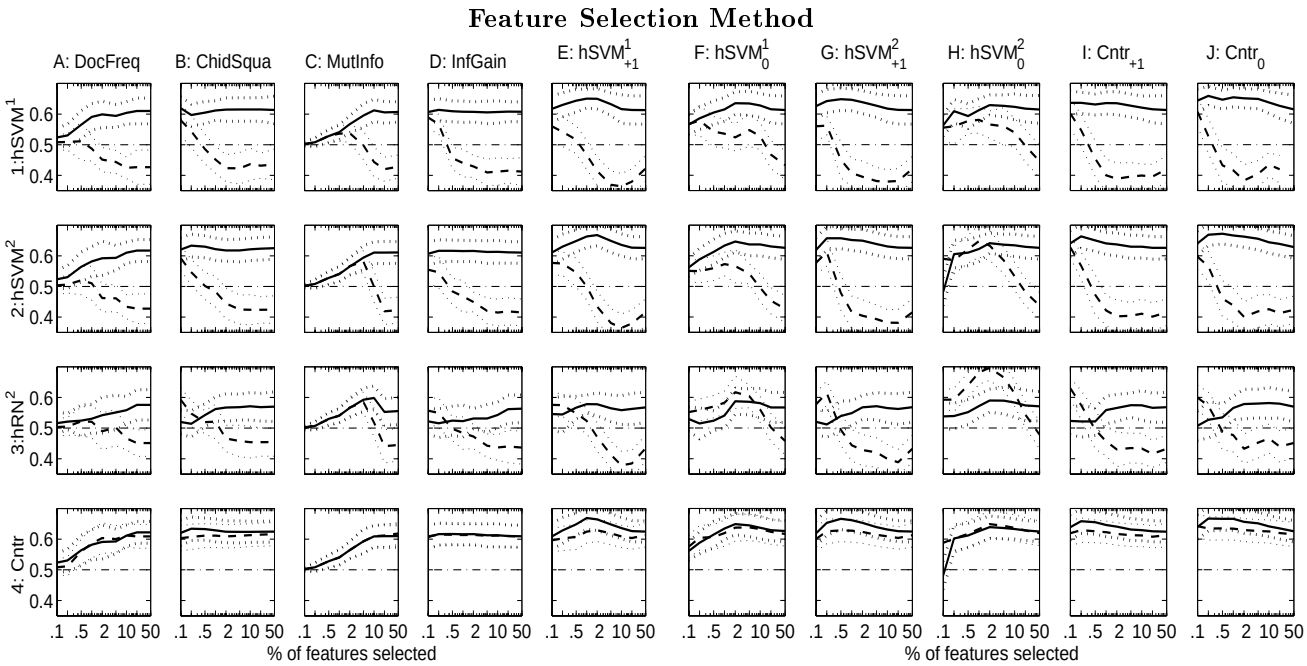


Figure 1. Evaluation of ten feature selection methods (columns A-J) by four classification algorithms (rows 1 - 4). Plots show mean AROC $\pm Std$ as an envelope as a function of the % of features selected out of the total 18,330. Curves are plotted for two modes: the positive 1-class ($B = +1$, solid lines) and balanced 2-class ($B = 0$, dashed lines). All SVMs used $C = 5000$.

3. $Cntr_0$ (row 4, dashed line) performs the best of all 2-class algorithms, generally matching 1-class centroid classifier, $Cntr_{+1}$.

4. 2-class $hSVM^2_{0,5000}$ (Columns H) provides very good features for 2-class mode classifiers, allowing them to perform above random $AROC > 0.5$. In fact, this feature selection in combination with ridge regression learning, $hRN^2_{0,5000}$, provides the best performance for around 2% (≈ 300) features.

5. As a general trend, features selected by models allow development of better discriminating models than features selected by the evaluated feature selection algorithms (A-D), provided positive 1-class mode is used for learning.

3.2.1 Selected Features

Table 1 lists the top 20 features selected according to different methods. We observe that there is a large overlap between features selected by χ^2 and positive 1-class methods: $hSVM^p_{+1,5000}$, $p = 1, 2$ and $Cntr_{+1}$. The features selected are primarily from function and localisation data. The 10 common features in the top 20 are: 1: F_4 - subcellular localisation, 2: F_7 - cell cycle and dna processing, 3: F_{10} - metabolism, 4: F_{17} - cell fate, 5: F_{27} - mrna transcription, 6: F_{28} - transcription, 7: F_{29} - unclassified proteins, 8: F_{32} - cellular transport and transport mechanisms, 9: F_{58} - protein fate (folding, modification, destination), 10: L_4 - cytoplasm.

Similarly, there are overlaps between $InfGain$ and the 2-class centroid, $Cntr_0$. However, these sets include many features from the abstracts, and thus are different from those selected by the positive 1-class methods. The 15 common features are: 1: F_4 - subcellular localisation, 2: F_{22} - nucleus, 3: L_3 - nucleus, 4: A_{419} - redund, 5: A_{426} - much, 6: A_{543} - abnorm, 7: A_{613} - comprom, 8: A_{639} - despit, 9: A_{711} - harbor, 10:A_{973} - surprisingli, 11: A_{1002} - subset, 12: A_{1291} - carboxi, 13: A_{1609} - green, 14: A_{2104} - taken, 15: A_{4290} - inviabl.

Interestingly, there are no overlaps in the top 20 features between the 2-class centroid ($Cntr_0$) and the 2-class SVMs ($B = 0$) or between $InfGain$ and the 2-class SVMs.

4. Tests on Reuters

Experiments reported in the previous section show that fringe SVMs tend to perform better than traditional 2-class SVMs on AHR-data. In this section we report some experiments with popular text mining benchmark, Reuters-21578 news-wires, which show similar tendency. For these experiments we used a col-

Table 1. Top twenty features selected by ten feature selection methods. We use the following convention: the letters stand for the data source (A -abstracts, F - function class, P - protein class, I - gene interactions, and L -localisation) and the subscript is the number of the feature. The last row gives mean AROC ± Std of the models used for the model-selection method. We put "−" in front of features with negative weights (2-class SVMs).

	Doc-	Chi-	Mut-	Inf-	$hSVM^1_{B,5000}$		$hSVM^2_{B,5000}$		$Cntr_B$	
Rank	Freq	Squa	Info	Gain	B=1	B=0	B=1	B=0	B=1	B=0
1	F_{95}	F_4	P_{48}	F_4	F_{29}	F_2	F_{29}	L_4	F_4	F_4
2	I_{1150}	F_{29}	P_{92}	F_{22}	F_4	F_{46}	F_4	I_{953}	F_{29}	A_{2104}
3	A_{1045}	F_{10}	P_{110}	L_3	F_{20}	F_{150}	F_{58}	I_{104}	L_3	F_{22}
4	I_{1136}	L_3	I_{61}	A_{2104}	P_{17}	F_{10}	F_{10}	$-F_{10}$	F_{22}	L_3
5	F_{172}	F_{22}	I_{66}	A_{4260}	F_{58}	F_{58}	L_4	$-F_{46}$	F_{10}	A_{4260}
6	F_{39}	F_{28}	I_{83}	A_{543}	L_4	$-I_{104}$	L_3	F_{110}	F_{58}	A_{426}
7	A_{89}	F_{58}	I_{85}	A_{426}	F_{21}	F_{45}	F_7	I_{835}	F_7	A_{639}
8	A_{925}	F_7	I_{95}	A_{711}	F_{10}	F_{38}	F_{22}	I_{210}	F_{17}	A_{543}
9	A_{970}	L_4	I_{533}	A_{1609}	F_{75}	F_{28}	F_{28}	P_{64}	A_{426}	A_{1002}
10	A_{1098}	F_{27}	I_{534}	A_{1002}	F_{86}	$-L_4$	F_{27}	F_{29}	A_{2104}	A_{1609}
11	A_{426}	F_{21}	I_{535}	A_{973}	F_{67}	$-I_{835}$	F_{17}	$-F_{28}$	F_{28}	A_{419}
12	A_{430}	F_{17}	I_{645}	A_{613}	F_{32}	$-I_{351}$	F_{67}	$-F_2$	F_{27}	A_{711}
13	A_{448}	F_{34}	I_{658}	A_{1291}	F_7	P_2	F_{32}	F_{75}	A_{639}	F_{58}
14	I_{1129}	F_{32}	I_{686}	A_{639}	F_{65}	F_{40}	P_{17}	$-F_{45}$	L_4	F_7
15	A_{669}	F_{26}	I_{727}	A_{562}	F_{66}	F_{20}	F_6	$-F_{58}$	A_{543}	A_{1714}
16	A_{586}	F_{15}	I_{838}	F_{27}	F_{64}	F_{32}	F_{21}	$-F_{39}$	F_{32}	A_{1291}
17	I_{1123}	F_{16}	I_{870}	A_{1699}	F_{17}	$-I_{210}$	F_{20}	A_{1721}	A_{1714}	F_{17}
18	I_{1299}	F_6	I_{871}	A_{849}	F_{28}	F_{59}	F_{65}	I_{351}	A_{973}	A_{613}
19	I_{1212}	A_{426}	I_{1050}	A_{1345}	F_{27}	P_{16}	F_{66}	P_{48}	A_{1282}	A_{973}
20	P_{17}	F_{25}	I_{1160}	A_{419}	F_6	F_{39}	F_{26}	$-F_{38}$	A_{1002}	A_{273}
AROC					.61 ± .05	.43 ± .05	.63 ± .04	.42 ± .05	.62 ± .03	.62 ± .04

lection of 12902 documents (combined test and training sets of so called modApte split available from http://www.research.att.com/lewis) which are categorised into 115 overlapping categories. Each document in the collection has been converted to a vector of 20,197 dimensional word-presence feature space (full analogy to the preprocessing of Abstract for AHR-data). Then we gradually removed the most frequent features (highest $DocFreq$ scores) and trained classifiers on random 5% (stratified sample) and then tested on remaining 95% of the data. As usual, the average AROC $\pm Std$ for 20 such tests is shown in Figure 2. Four different target cases were used: the 3rd, the 6th, the 9th and the combined 11th-15th largest categories. The sizes of target classes are shown in the sub-figure titles.

An inspection of plots highlights a few observations:

1. The accuracy of all classifiers is very high when all features are used. As the most frequent features are removed, all SVM models start degenerating, however, the drop in performance for 2-class SVM models is much larger, and 1-class SVM models start outperforming the 2-class models. This behaviour is also present in other categories not shown in Figure 2, so long as the target class is less than 10% of the total data. This trend of better performance with 1-class models is most apparent in $hSVM^1_{B,5000}$, although $hSVM^2_{B,5000}$ also shows similar trends. Thus, when there are many weakly informative features, and the target class is a small fraction of the data set, the fringe classifiers outperform traditional 2-class SVM models.

2. The mean AROC is always > 0.5 indicating that even after feature removal, this data set does not quite have all the properties of AHR-data where 2-class models performed worse than random for many settings of the regularisation constant.

5. Discussion

Related Research. A possibility of single class learning with support vector machines (SVM) has been noticed previously. In particular, Schölkopf et al. [18] have suggested a method of adapting the SVM methodology to 1-class learning by treating the origin as the only member of the second class. This methodology has been used for image retrieval [3] and for document classification [14]. In both cases, modelling was performed using examples from the positive class only, and the 1-class models perform reasonably, although much worse than the 2-class models learned using examples from both classes.

In contrast, in this paper, we show that for certain problems such as AHR-data, positive 1-class SVMs significantly outperform models learned using examples

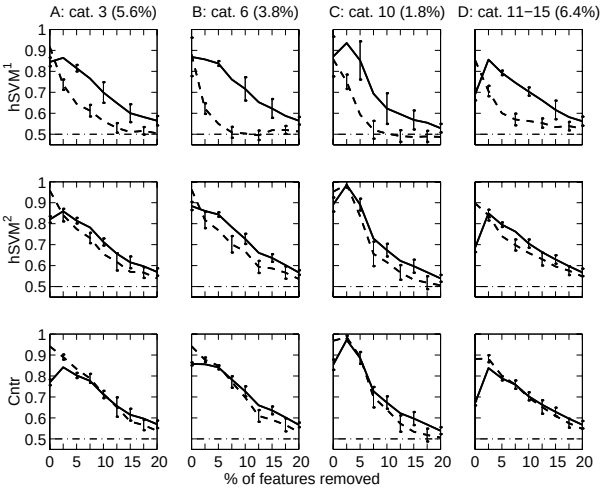


Figure 2. Mean AROC as a function of the % of features removed (with standard deviation envelope). Four different target cases were used: the 3rd, the 6th, the 9th and the combined 11th-15th largest categories. Results are presented for four machines: (1) $hSVM^1_{B,5000}$, (2) $hSVM^2_{B,5000}$ (3) $RN^2_{B,5000}$ and (4) $Cntr_B$. Plots are shown for the positive 1-class ($B = +1$) (solid line) and the balanced 2-class ($B = 0$) (dashed line) modes.

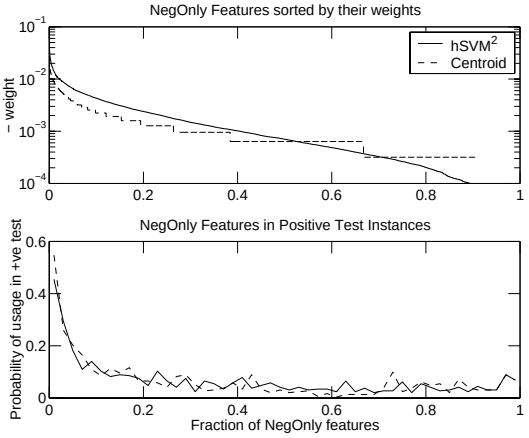


Figure 3. Understanding the influence of sparse high dimensional space on the solutions of 2-class learners. (A) Magnitude of weights for 2-class models for the *NegOnly* (features used only in the negative class in the training set) features in decreasing order of magnitude. (B) Usage of *NegOnly* features in the positive test set. Plots are shown for $hSVM^2_{0,5000}$ (solid line) and $Cntr_0$ (dashed line).

from both classes.

Impact of regularisation constant. In this paper, we have restricted our investigation to $C = 5000$ (the "hard margin" case) and the very low values of C through the $Cntr$ classifier. However, other experiments with different values of C show that the performance of 1-class SVMs, unlike that of 2-class SVMs, is very robust across the whole range of C values [12]. The performance of 2-class SVMs improves as the regularisation constant decreases, and for very low values of C, its performance is roughly equal to that of 1-class SVMs.

Deterioration of 2-class SVMs. We have observed that for AHR-data, fringe SVMs tend to have systematically better AROC than the traditional 2-class SVMs. Typically, the latter deteriorates with increase in the number of features used. In order to gain some insight into this phenomenon, we have compared two 2-class models, a $hSVM^2_{0,5000}$ (test $AROC = 0.39$) and $Cntr_0$ (test $AROC = 0.63$) trained on the same data split. For this training set, we found that there were 14,610 features occurring only in the negative class training instances (*NegOnly features*). Both models allocate non-positive (all negative for $Cntr_0$) weights to such features. Our hypothesis is that for many of these features $hSVM^2_{0,5000}$ allocates excessively low (highly negative) weights, which is an 'easy way' to minimise the margin errors. However, when some of these features occur in positive test examples, they push the scores of these examples excessively into negative direction, which causes a deterioration in the overall performance.

Figure 3 shows results corroborating this hypothesis. In Figure 3A we plot weights allocated to the *NegOnly* features, sorted in the reverse order of their magnitude, for each model separately and for each weight vector normalised to the unit length. Figure 3B shows probability of usage of these features in the positive class test examples (the curves are in fact 50-bin histograms). For both models the usage distribution is very similar and the most popular *NegOnly* features have the most negative weights. However, the weights from SVM for those most popular *NegOnly* features are about twice as large in magnitude as those for $Cntr_0$ model. These weights result in excessive decreasing of scores of some positive test instances leading to deterioration in the overall performance.

Persistent dominance of 1-class SVMs. The above analysis is applicable to a high dimensional feature set. However, we have also observed in Section 3.2 that even in low dimensional spaces, this phenomenon of better performance with one-class learners persists. Our intuitive explanation here is that if the learner uses the minority class examples only, the "corner" (the half space) where minority data resides is properly determined. However, when data from both classes is

used, the minority class is "swamped" by the background class. In such a case the SVM solver seeking a "maximal margin" separation between classes, chooses a direction which is suboptimal in terms of AROC. The strange thing is that heavy discounting of the majority class by a factor $B = 0.99999$ does not rectify this impact completely [11].

Weakly informative features. An alternative explanation for the relatively good performance of fringe classifiers is implied by experiments with Reuters data. We hypothesise that one factor is the relatively "weak" connection between the labels and the features in the case of AHR-data. Since the contrary is true for topic-based classification in Reuters, the superior performance with fringe classifiers is not evident until the most frequent features, which tend to be strongly indicative of the labels for this dataset, are removed (Figure 2). Thus, we may expect fringe classifiers to work well in other real world applications with weak connection between labels and attributes.

Importance of evaluation algorithm for feature selection. An additional point regarding feature selection is that the performance of any dedicated statistical system for that purpose is a function of both, the feature selection method and the learning strategy for evaluation of the selection. For instance, all 1-class SVMs and all centroid learners in Figure 1 perform very well with features selected by *ChiSqua* and *MutInfo*, while all 2-class learners, other than the 2-class centroid, perform poorly with the same features.

6. Conclusion

We have shown that SVMs even for a single kernel can split into a number of different modes, with dramatically different performance. Thus this popular class of learning machines cannot be treated as a monolithic black box, but should be viewed as a rich family of classifiers that need to be carefully tuned if top performance is required.

Further, some easy to implement fringe classifiers, such as centroids and positive 1-class SVMs, often outperform complicated 2-class SVMs. The very good performance of the fringe classifiers is related to sparsity of data and weak links between labels and features, and persists even after aggressive feature reduction. Thus, these classifiers could be used as baseline methods for biomining in general and for machine evaluation of utility of various feature selections. In particular, we recommend the use of centroid classifiers, which are trivial to implement.

Finally, model-based feature selection techniques can provide better results than dedicated feature preselection algorithm, facilitating development of more accurate discriminating models.

ACKNOWLEDGEMENTS

The permission of the Managing Director, Telstra Research Laboratories, to publish this paper is gratefully acknowledged.

References

[1] D. Bamber. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *J. Math. Psych.*, 12:387 – 415, 1975.

[2] R. Centor. The use of ROC curves and their analysis. *Med. Decis. Making*, 11:102 – 106, 1991.

[3] Y. Chen, X. Zhou, and T. Huang. One-class svm for learning in image retrieval. In *Proceedings of IEEE International Conference on Image Processing (ICIP'01 Oral)*, 2001.

[4] M. Craven. The Genomics of a Signaling Pathway: A KDD Cup Challenge Task. *SIGKDD Explorations*, **4(2)**, 2002.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, Cambridge, 2000.

[6] G. Forman. Feature Engineering for a Gene Regulation Prediction Task. *SIGKDD Explorations*, **4(2)**, 2002.

[7] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

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

[9] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the Tenth European Conference on Machine Learning ECML98*, 1998.

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

[11] A. Kowalczyk and B. Raskutti. One Class SVM for Yeast Regulation Prediction. *SIGKDD Explorations*, **4(2)**, 2002.

[12] A. Kowalczyk and B. Raskutti. Exploring Fringe Settings of SVMs for Classification. In *Proceedings of the Fourteenth European Conference on Machine Learning ECML03 (to appear)*, 2003.

[13] M. A. Krogel, M. Denecke, M. Landwehr, and T. Scheffer. Combining Data and Text Mining Techniques for Yeast Gene Regulation Prediction: A Case Study. *SIGKDD Explorations*, **4(2)**, 2002.

[14] L. M. Maneivitz and M. Yousef. One-class SVMs for Document Classification. *Journal of Machine Learning Research*, 2:139–154, 2002.

[15] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, **1**(1), (1986).

[16] B. Raskutti, H. Ferrá, and A. Kowalczyk. Second Order Features for Maximising Text Classification Performance. In *Proceedings of the Twelfth European Conference on Machine Learning ECML01*, 2001.

[17] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, 1983.

[18] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution, 1999.

[19] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2001.

[20] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[21] G. Whaba. Support vector machines, reproducing Hilbert spaces and the randomised GACV. In B. Schölkopf, C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods*, pages 69–88, Cambridge, Ma., 1999. MIT Press.

[22] Y. Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

Using Decision Trees and Support Vector Machines to Classify Genes by Names

Simon Lin[1], Sandip Patel[2], Andrew Duncan[2], and Linda Goodwin[3], {Simon.Lin, Sandip.Patel, Aduncan, Linda.Goodwin}@duke.edu, [1]Bioinformatics Shared Resource, [2]Department of Pharmacology and Cancer Biology, and [3]School of Nursing, Duke University Medical Center, Durham, NC 27710

ABSTRACT[·]

We report a new text classification task originated from our high-throughput genomic labs. Biologists desire to have an automatic tool to separate tens of thousands gene names into two categories: known and unknown ones, based on their gene names. To solve this typical text classification problem, we first evaluated two off-the-shelf machine learning tools. In parallel, we also asked the human-expert to derive a set of rules. Both decision trees (CART) and Support Vector Machines (SVMs) outperform the human-expert rules; the cross-validated error rates are below 1%, and the area under the curve of Receiver Operating Characteristic (ROC) curves reach higher than 0.99. In summary, we describe a new text classification problem that is biologically important. Our investigation indicates that off-the-shelf programs can adequately solve the problem; no special text mining methodology-development is necessary for this task. The classifier we built has been successfully utilized in several internal projects, and it can be applied to many related applications of high-throughput genomics.

Keywords

Gene name, Known gene, unknown gene, Classification and Regression Tree (CART), Support Vector Machines (SVM), expert system, text data mining

1. INTRODUCTION

Genes are named to convey some comprehensible meaning. Simply by its name, biologists can usually tell if a gene is a known one or an unknown one. By classifying the gene into the known category, the expert indicates that the gene was previously studied or characterized. For example, "superoxide dismutase" is a known gene, whereas "ESTs, Weakly similar to MAP-kinase activating death domain" is an unknown one.

With the rapid advance of genome sequencing, millions of gene names have been deposited in the Genbank (http://ncbi.nlm.nih.gov) and other private databases. One particular problem arose from Duke University Comprehensive Cancer Center research labs is to find all the unknown genes from these databases, and then make a specialized microarray just for unknown genes.

This particular problem motivates us to investigate the possibility of classifying genes purely based on gene names. To our knowledge, this problem was not previously reported in the literature. Identifying if the gene is known or unknown also has many other practical applications:

- In computational annotation of genomic sequences, annotators are generally transferring the name and function of a known gene to the un-annotated one. By knowing which genes are already functionally known, we can prevent "null-chaining" by claiming a gene's function and directing it to another unknown gene (Karp *et al.*, 2001).
- In pharmaceutical research, unknown genes are usually given higher priority as a potential novel drug target. Quickly finding out what are the unknown genes in a large screening data set will let the investigator make an informed decision regarding which targets to pursue experimentally (Jones & Fitzpatrick, 1999).

[·] In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

- In analyzing the statistics of genomic databases, we need to know what percentage of the genes are still functionally unknown.
- In the tracking and cleaning of genomic data warehouses, it is crucial to monitor the status of the genes being changed from unknown to known. Updated nomenclature of the genes can expedite biological discoveries by providing information regarding an unknown gene that might not otherwise be considered by human viewing database identifiers.

However, judging if a gene is known or unknown by using a human-expert to look at the name is not only time consuming but is also prone to human errors. Thus, we would like to devise a machine that can mimic human experts in performing the same task. That casts the problem into a typical machine learning task, which usually contains two phases. In phase one, a data set will be acquired from a human expert; i.e., the true answer to the problems are labeled by the expert. In the next phase, a machine learner will be trained. Cross-validation will be used to assess how well the machine 'learns' and to prevent overfitting.

In this paper, we report our investigation of two off-the-shelf machine learning tools -- decision trees (CART) and support vector machines (SVM)-- to classify genes into known and unknown categories by looking at their names.

2. RELATED WORK

Since no previous research has been reported on categorizing genes into known and unknown groups by their names, we reviewed literature in text categorization and machine learning. The problem we are facing resembles a lot with the well studied news classification problem (Kongovi *et al.*, 2002).

In text categorization, it is indicated (Yang & Liu, 1999) that SVM and k-nearest neighbor (kNN) techniques outperform neural networks (NNet) and naïve Bayes (NB) classifiers. Rule-based learners were also studied due to their expressive power (Cohen, 1996). Thus, we first investigate SVM and CART in this paper.

3. METHODS

Knowledge acquisition from the expert

Training data set. We randomly selected 1,624 genes and presented their names to the human expert. The expert was asked to classify the genes into known or unknown categories.

Expert rule-sets. We also asked the expert how he conducted the classification. The expert generated a set of rules that included certain keywords (See Figure 1).

Data model

We chose a much simpler data model than the commonly used term frequency/inverse document frequency weighting (TF-IDF) model (Salton, 1991). Each gene name is represented by a vector of words, where 1 indicates the presence of the word and 0 indicates the absence of the word. A survey of the training data indicates the gene names have a different structure than regular text data. Thus, we do not remove the so called stop words, such as 'and', 'or', and 'is'. However, we do filter out infrequent words and numbers from the training set. A high-pass filter at a frequency of 2 is used. Our classification performance indicates that this simple data model is adequate for this task.

CART

We used the rpart implementation of CART (Breiman, 1993) in the R-statistical package (http://cran.r-project.org). Five-fold cross validation was used to select the complexity (Cp) parameter. Cp parameter controls the size of the tree. Large tree can cause over fitting. Thus, we used the parsimonious "1-SE" rule (Venables & Ripley, 2002) to choose the Cp associated with the largest error rate within one standard deviation of the minimum error rate.

SVM

SVM is a new learning method introduced by Vapnik and coworkers (Vapnik, 1995). We used the SVMlib (Chang & Lin, 2001) implemented in the R-

statistical package. Different choice of kernels--linear, polynomial, radial basic function, and sigmoid -- flexibly maps the input space into a higher-dimensional space where the cases are separated with the maximum margin. Five-fold cross validation was used to select the appropriate kernels.

ROC curve comparison

To compare the two classification methods, we use the ROC analysis (Metz, 1978). Receiver Operator Characteristics (ROC) curves plot sensitivity (Y axis) against 1 minus the specificity (X axis) providing a clear visualization of area under the curve (AUC). The greater the accuracy, the greater the AUC (1.0 is perfect classification). Thus, the 'better' classification method is the one with the most area under the curve.

Outside data set to be classified

We tested the production system with a set of 7,447 genes to be classified. These genes were retrieved from the Affymetrix U74A DNA microarray chip.

4. RESULTS

Training data set acquisition

We acquired a training data set consisting of 1,624 mouse gene names, of which 698 genes are known, and 926 genes are unknown. After filtering out infrequent words, this training data set consists of 522 unique words as attributes, and expressed as a 1,624 by 522 data matrix of zeros and ones. The training data set acquired from the expert is called T0 (see Table 1). During the machine learning process, we found errors in this data set (details are discussed later). The corrected data set is called T1.

Table 1. Training data sets.

	Data Set T0	Data Set T1
unknown	926	929
known	698	695
total	1,624	1,624

Expert rule-set acquisition

The expert summarized his knowledge into a set of rules to look at certain keywords (Figure 1). Performance of the rule-based classifier is summarized in Table 2.

```
1. if ((x$ESTS==1))    {
    y.predicted <- 0}
2. if ((x$HYPOTHETICAL==1)) {
    y.predicted <- 0}
3. if ((x$SIMILAR==1)& (x$TO==1)) {
    y.predicted <- 0}
4. ((x$CDNA==1)& (x$SEQUENCE=1)) {
    y.predicted <- 0}
5. if ((x$RIKEN==1) & (x$CDNA==1)) {
    y.predicted <- 0}
6. if ((x$EXPRESSED==1) & (x$SEQUENCE==1)) {
    y.predicted <- 0}
7. if ((x$DNA==1) & (x$SEGMENT==1)) {
    y.predicted <- 0}
8. if ((x$MUS==1)& (x$MUSCULUS==1) &
    (x$CLONE==1)){
    y.predicted <- 0}
```

Figure 1. Classification rules derived by the expert. For example, rule #1 indicates if there is an "EST" in the gene name, then the predicted class of this gene is unknown (class 0).

Table 2. Confusion table by expert rules using training data set T1.

	true.unknown	true.known
predicted.unknown	923 (56.8%)	10 (0.6%)
predicted.known	6 (0.4%)	685 (42.2%)

Classifier by CART

CART was used to induct the decision tree. To determine the best complexity of the tree, we ran a five-fold cross validation. According to the "1-SE" rule, we chose Cp = 0.0018 (Figure 2), which corresponds to a tree with 9 leaves (Figure 3).

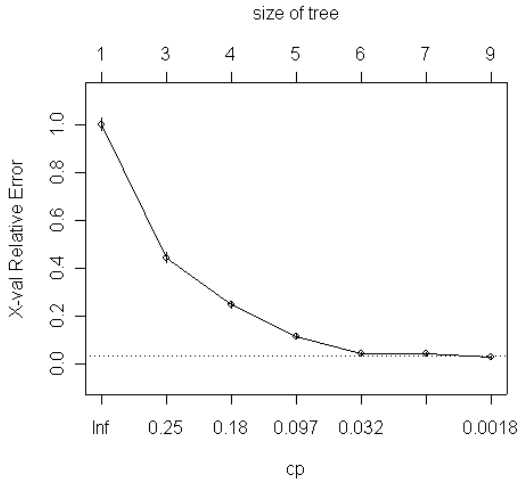


Figure 2. Using five fold cross-validation to determine the best complexity parameter Cp. The dotted line indicates the highest error rate within one SE of the lowest cross-validated training error. Results were obtained using training data set T0.

After cross-validation, we used all training data in set T1 to induce the tree. Results are shown in Figure 3 and Table 3.

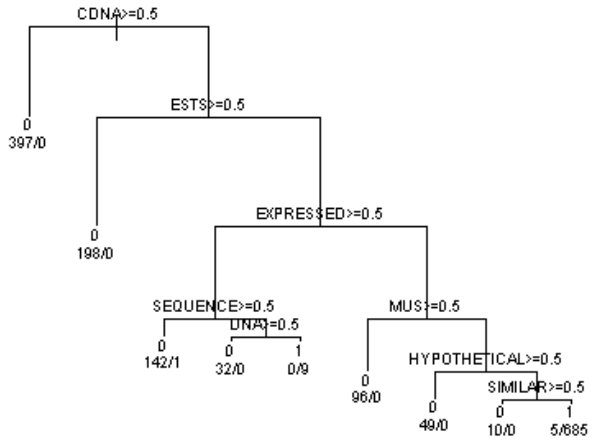


Figure 3. Decision tree by CART analysis. Unknown genes (class 0) are distinguished from know genes (class 1) by answering a series of questions in the flow chart. The purity of each leave is shown below each leave (class0/class1).

Table 3. Confusion table by CART using training data set T1.

	true.unknown	true.known
predicted.unknown	924 (56.9%)	1 (0.1%)
predicted.known	5 (0.3%)	694 (42.7%)

Classifier by SVM

To choose the best kernel for SVM, we ran a five-fold cross validation. A simple linear kernel is the best choice (Table 4). The performance of the SVM classifier is summarized in Table 5.

Table 4. Choice of kernel by SVM using five fold cross-validation (data set T1).

Kernel	Error Rate (%)
linear (gamma=0.0019)	**0.3 ± 0.4**
polynominal (degree=3, gamma=0.0019)	42.8 ± 2.0
RBF (gamma=0.0019)	8.8 ± 5.4
sigmoid (gamma=0.0019)	25.1± 18.9

Table 5. Confusion table by linear SVM using training data set T1.

	true.unknown	true.known
predicted.unknown	928 (57.1%)	1 (0.06%)
predicted.known	1 (0.06%)	694 (42.7%)

Not all training data points are equally important in determining the decision boundary of SVM. "Support vectors" are those critical data points that are close to the separating hyperplane. A list of the supporting vectors is found in Table 6. This data can help us understanding how SVM perform the task, although SVM is largely treated as a 'black-box' classifier.

Table 6. Support vectors indicates those cases close to the decision boundary.

	gene.name	support
1	acetylcholinesterase	1.0000000
2	aconitase 1	1.0000000
3	apelin	1.0000000
4	axin	1.0000000
5	carbonic anhydrase like sequence 1	1.0000000
6	expressed sequence 2 embryonic lethal	1.0000000
7	major urinary protein 2	1.0000000
8	placental protein 6	0.9363062
9	EST AI426782	-1.0000000
10	EST AI447490	-1.0000000
11	ESTs	-1.0000000
12	expressed sequence 2 embryonic lethal	-1.0000000
13	expressed sequence AA408140	-1.0000000
14	expressed sequence AA420392	-1.0000000
15	expressed sequence AA517758	-1.0000000
16	hypothetical protein MNCb 4414	-1.0000000
17	hypothetical protein DKFZp564K0822	-0.9991421
18	RIKEN cDNA 0610007P06 gene	-1.0000000
19	WAVE3	-1.0000000

Machine classifiers detect errors made by experts in the training data set

Using CART and SVM, we found some mistakes made by the human expert in training data set T0, and revised the training data set into set T1 accordingly (Table 7). In the next section, we use the training data set T1 to re-induct the classifiers and compare them.

Table 7. Mistakes made by the human expert in training data set T0.

Gene Name	Label in T0	Label in T1
ESTs Weakly similar to A Chain A Crystal Structure Of The Human Acyl Protein Thioesterase 1 At 1 5 A	known	unknown
expressed sequence AI173355	known	unknown
Mus musculus Similar to hypothetical gene LOC150274 clone MGC 41340 IMAGE	known	unknown

Comparing the three classifiers

Comprehensibility. The expert rules and CART are easily comprehensible. They describe what attributes are important for classification; a flow chart is given for performing the classification. In contrast, SVM reveals the classification from another angle by looking at the critical cases. In that sense, Nishikawa and colleagues (El-Naqa *et al.*, 2002) called SVM a template-matching detector. Table 6 shows the cases on the decision boundary whose 'support' is important to make the decision. They consist of the difficult-to-classify examples on the "border line". The SVM classifier memorizes these cases as critical knowledge to perform the task. In general, we can see the three classifiers acquired the same knowledge such as 'Riken cDNA' and 'hypothetical protein' etc for the classification.

Performance and efficiency. As shown in Table 8, expert rules are the most expensive to obtain. It took more than 10 hours for the expert to generate this set of rules. The production runtime for the three methods are comparable. The error rate in Table 8 is a non cross-validated error rate, since it is impossible to cross-validate the expert rule process.

Table 8. Comparison of three classifiers.

	Expert Rule	CART	SVM
nominal error rate	0.99%	0.37%	0.12%
rule induction time	> 10 hr	< 5 min	< 5 min
production runtime	< 5 min	< 5 min	< 5 min

ROC. Since no threshold structure exists in the expert rules, only CART and SVM are compared here. Both CART and SVM achieve a good balance of sensitivity and specificity. The area under the curve for CART and SVM are 0.9971 and 0.9995, respectively (see Figure 4).

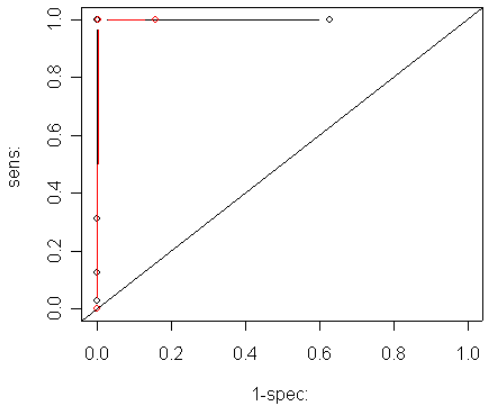


Figure 4. ROC analysis of CART and SVM. Red: CART; Black: SVM. The area under the curve (AUC) of CART and SVM are 0.9971 and 0.9995, respectively.

An application of the machine classifiers

We applied the two machine classifiers to categorize 7,447 genes on the Affymetrix U74A microarray DNA chip.

Table 9. Prediction results of CART and SVM for the 7,447 genes on the Affymetrix U74A microarray chip.

	CART	SVM
predicted.unknown	1747 (23.5%)	1797 (24.0%)
predicted.known	5700 (76.5%)	5650 (76.0%)
total	7447 (100%)	7447 (100%)

The major discrepancy of CART and SVM prediction results in Table 9 is that of 51 genes, where CART classified them as known but SVM classified them as unknown. A manual inspection of these genes indicated that the CART rule is unable to handle cases like "EST C78513" (34 cases) and "DNA Segment Chr 6 human D12S2489E" (12 cases) correctly. Thus, the production performance is in agreement with the evaluations on the training data set. The SVM classifier consistently outperforms the CART classifier in this application.

5. DISCUSSION

We have successfully built a text classification system that is able to capture knowledge from the

expert to perform a gene name classification task. We have shown that the rule-acquisition process from the human-experts is not only time-consuming, but also prone to errors. The machine classifiers have been able to find classification errors by experts in the training data set. Similar experience has also been reported in a leukemia microarray data set (Golub *et al.*, 1999), where a consensus of different classifiers strongly suggests the potential errors made by the human expert (Lin & Johnson, 2002).

In this application domain, we observed that SVM outperforms CART, although the prediction errors of both are acceptably low for production purposes. It can be explained by the capability of SVM to handle high-dimensional data based on Vapnik's statistical learning theory. In contrast, CART utilizes tree nodes to select only a small number of attributes for classification. In text data mining, a variable selection by CART might not be appropriate, because the message in the text is not only conveyed by certain keywords but also by other words in the context. In other words, text data is presented by a dense concept vector where few irrelevant features exist (Joachims, 1998).

We demonstrated an application of this system where we classified all the genes on a mouse U74A chip. It is one of the most commonly used DNA chips in biomedical research. According to Affymetrix, on the U74A microarray chip, the majority of the genes are known ones. However, there is no quantitative description of what this "majority" is. Here we estimate 70% of the genes on U74A are known ones. With our results, we can tell which genes are known or unknown on this chip. This system can help to accelerate the drug development process where priority is given to unknown genes after a microarray screening.

Actually, classification of genes into known and unknown categories can also be achieved by using other information from the database, other than the gene names. For example, if there are publications of a gene, it is a strong indication that this gene is a known one. We started our investigation simply from using gene names, which is the absolutely available information associated with the task, since publications related to genes are often lag-behind in database registration.

6. CONCLUSIONS

We described a new text data mining problem emerged from genomic research. To our knowledge, this gene name classification problem was not reported previous in the literature. However, the problem resembles the classical news categorization problem. Thus, we first assessed off-the-shelf machine learning tools.

To prevent over-fitting, both CART and SVM are deployed on cross-validated training sets. For this task, both methods perform well. Therefore, no new algorithm is necessary to be developed. Our experience indicates that SVM has better capability to handle high-dimensional text data where few irrelevant features exist.

Although the result of our investigation indicates the solution is simple, the impact of this classification tool is non-trivial. It has appropriately solved our biomedical research problems. We have utilized this tool to select unknown genes in several research projects, including microarray design and microarray screening.

7. ACKNOWLEDGMENTS

The authors thank Kim Johnson for motivating discussions.

8. REFERENCES

[1] Breiman L. (1993). "Classification and regression trees," Chapman & Hall, New York, N.Y.

[2] Chang C. C., and Lin C. J. (2001). Training nu-support vector classifiers: Theory and algorithms. *Neural Computation* **13:** 2119-2147.

[3] Cohen W. (1996). Learning rules that classify e-mail. *In* "Proc. Machine Learning in Information Access", AAAI.

[4] El-Naqa I., Yang Y. Y., Wernick M. N., Galatsanos N. P., and Nishikawa R. M. (2002). A support vector machine approach for detection of microcalcifications. *IEEE Transactions on Medical Imaging* **21:** 1552-1563.

[5] Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., Coller H., Loh M. L., Downing J. R., Caligiuri M. A., Bloomfield C. D., and Lander E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286:** 531-7.

[6] Joachims T. (1998). Text categorization with support vector machines: learning with many relevant features. *In* "Proceedings of {ECML}-98, 10th European Conference on Machine Learning", Springer Verlag, Heidelberg.

[7] Jones D. A., and Fitzpatrick F. A. (1999). Genomics and the discovery of new drug targets. *Curr Opin Chem Biol* **3:** 71-6.

[8] Karp P. D., Paley S., and Zhu J. (2001). Database verification studies of SWISS-PROT and GenBank. *Bioinformatics* **17:** 526-32; discussion 533-4.

[9] Kongovi M., Guzman J. C., and Dasigi V. (2002). Text categorization: An experiment using phrases. *In* "Advances in Information Reftrieval", pp. 213-228.

[10] Lin S. M., and Johnson K. F. (2002). "Methods of microarray data analysis : papers from CAMDA '00," Kluwer Academic Publishers, Boston.

[11] Metz C. E. (1978). Basic principles of ROC analysis. *Semin Nucl Med* **8:** 283-98.

[12] Salton G. (1991). Developments in Automatic Text Retrieval. *Science* **253:** 974-980.

[13] Vapnik V. N. (1995). "The nature of statistical learning theory," Springer, New York.

[14] Venables W. N., and Ripley B. D. (2002). "Modern applied statistics with S," Springer, New York.

[15] Yang Y., and Liu X. (1999). A re-examination of text categorization methods. *In* "Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information", ACM Press, New York.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

Visualisation and Analysis of Bibliographic Networks in the Biomedical Literature: A Case Study

Michael Schroeder[1] and Cecilia Eyre[2]

[1] Department of Computing, City University, London, UK, msch@soi.city.ac.uk
[2] Unilever R&D Colworth, Sharnbrook, Bedford, MK44 1LQ, UK, Cecilia.Eyre@Unilever.com

Abstract: In this article, we present a case study on the analysis of the biomedical patent literature. For a set of Unilever's patents we show how to apply visual datamining by applying dimension reduction to document, term, and author data and interactive exploration of 3D scatterplots. We complement this approach by focusing on author and document networks, which we analyse with PSIEYE, a tool for the graph-theoretic analysis of interaction networks (Schroeder et al. 2003). We show how to apply graph-theoretic measures such as sum of distance, connectivity, cluster index and interaction rank to the networks and how to derive maps of the trends and topics in the documents and of the most important authors behind these trends.

Introduction

Biomedical literature grows at a tremendous pace of over 8000 abstracts a month. Due to this size, simple web-based text search of the literature is not yielding the best results and a lot of important information remains buried in the masses of text. Textmining of biomedical literature aims to address this problem. There have been a number of approaches using literature databases such as PubMed to extract relationships such as protein interactions (Blaschke et al. 1999, Thomas et al. 2002), pathways (Friedman et al. 2001), and micro array data (Tanabe, 1999). For a good overview see (Feldman et al. 2003). Mostly, these approaches aim to improve literature search by going beyond mere keyword search by providing natural language processing capabilities. While these approaches are successful in their remit, they do not mimick human information foraging. A typical biologist might pursue a quite sophisticated strategy to find the right papers. To get the most relevant literature on breast cancer and micro arrays, a biologist might enter these keywords into a search engine or PubMed. A quick scan of the best hits and the papers they cite will give a good indication on which labs around the world use micro arrays to investigate breast cancer. Then the user may visit the pages of the labs and look at their publication track record focusing on the prime researcher within the groups and publications in top journals. For these hits, the biologist will then read abstracts and possibly the whole paper.
The important difference of the above information foraging strategy to pure keyword search is that implicitly users navigate networks – networks of documents and networks of authors. The idea of considering network topology for searching has already proved successful for general search engines such as Google. In this paper, we pursue this line of research and apply it to mine a small part of Unilever Bestfoods' patent portfolio. We will use PSIEYE (Schroeder, 2003), a tool for the analysis and visualisation of interaction networks, to analyse a specific author and document network of patent literature. Besides this network approach, we show how classical term co-occurrence data can be visually mined for relevant terms and relationships.
The paper is organised as follows: First, we introduce our application domain of mining patents. Next, we discuss the types of data available and some basic analysis techniques such as clustering and 3D scatter plots to analyse the documents. This motivates our novel approach of document and author network analysis. To this end, we review PSIEYE, a tool for the graph-theoretic analysis of interaction networks (Schroeder et al. 2003) and show and discuss results for the patent analysis.

Mining Biomedical Literature and Patents

Besides the main literature databases such as PubMed there is a wealth of information in patent databases available. Most of these patent databases provide only very simple keyword search facilities. However, one also provides the possibility to submit sequences and search patents that have mentioned similar sequences. To fully use the potential of these literature databases a more thorough analysis is required. (Breitzman and Mogee 2002) have reported on the many applications of patent analysis and have described a number of patent analysis software tools. VantagePoint is one of these tools and is capable of data and text mining. They also list a couple of features that VantagePoint displays, the

ability to make lists and co-occurrence matrices and it is noted that co-occurrence techniques may be used to identify areas of competence. A company's patent portfolio can for example reveal key personnel, technology areas of interest and inventor networks. It must be realised that any patent portfolio is dependent on publication policies and how prolific each inventor is.

(Porter at al 2002) report that VantagePoint is a useful tool for bibliographic analyses of up to 20000 references for text, numerical and graphical depiction. By visualising large data sets it is possible to use broader searches which may in turn enable topical linkages to be seen, that would otherwise have been lost. In order to produce maps Porter et al select keywords of interest which are then visualised. The software does not allow 'drilling down' from a topical view to increasing detail. A different approach has been taken by (Buter and Noyons 2001), who present maps based on co-occurrence of phrases where the map is intended to provide an interface into a large set of bibliographic references, in the region of 100000 records. Furthermore, their maps have one significant capability not currently displayed by VantagePoint and it is the possibility of showing a dynamic version of a map, where they animate the changes in position of the clusters over time. In order to aid the strategic direction of a business it is useful to follow temporal shifts of science and technology trends from literature and patent publications, as a backward looking guide. Another software is ArrowSmith (http://arrowsmith.psych.uic.edu/cgi-test/arrowsmith_uic/start.cgi), which is based on the work by (Swanson 1987 and Swanson and Smalheiser 1998) and enables the user to find scientific cross-category links by looking at titles in PubMed. In ArrowSmith, the user provides two topics, A and C, of interest and ArrowSmith fills in the gaps to relate these two topics. It does so by identifying common terms from the two topics, by finding titles containing the words A and B, and B and C, respectively. Thus it can be implied that A relates to C via B, the first example published was the discovery that articles on Raynaud's disease and articles on eicosapentaenoic acid when considered together implied that Raynaud patients may benefit from dietary fish oils rich in this acid. Two years later clinical trials proved this hypothesis to be correct. Obvious drawbacks of this tool are that it can only be used on publications in PubMed and that it only works on titles, as there can be a large number of other sources of interest as well as the opportunity of utilising the knowledge in the whole paper is missed. For multi-national corporations innovation has to grow at ever increasing rates in order to remain competitive. Discovery of cross-category links (such as A, B ,C) is a very attractive route to increase efficiency in research. To these authors there are currently no tools available on the market that can deal with these kinds of issues, using any source, any text and with visualisation possibilities that include showing the bigger picture as well as zooming into greater detail.

Case Study: Unilever Patents

In this paper, we will use a case study consisting of a set of patents by Unilever to compare some existing approaches with our novel approach of mining document and author network. The authors have decided to carry to this case study on a small dataset that is well known to one of the authors in order to be able to determine the effectiveness of the text analysis and subsequent visualisation. Thus the dataset contains 59 patents, originating from a search made in the Micropatent database, all by the author Norton at Unilever. While most approaches aim to pinpoint the most relevant documents, we aim to map out the topic and trends implicit in a set of documents. For the former task, documents are usually represented as term vectors. For the latter task, we use a complementary approach: We want to map out and cluster relevant terms from the documents and thus we represent terms as document vectors. This is possible using VantagePoint via a co-occurrence matrix of terms by document with subsequent visualisation. However, this visualisation is limited to represent only a very small (or a selection of a) dataset to a maximum of ca 30-50 clusters in a 2D scatterplot. More clusters make CPU time excessive and the resultant visualisation difficult to interpret when using the current version of VantagePoint. Additionally, we obtain a term vector for each author. The term vector represents a profile for each author containing all the relevant terms an author used in the abstract of the patents he or she co-authored. Given the two matrices, how can we analyse them? First of all, let us consider the analyses that VantagePoint provides (see Figure 1). We applied dimension reduction with principal component analysis (see Figure 2) to each document reducing the matrix of the above document vectors and *term vectors, respectively,* to the three factors, which cater for most of the variance in the data. Next, we visualised the resulting 3D matrix as a 3D scatterplot using SpaceExplorer (Schroeder et al. 2001).

Consider Figure 3. It shows distinct clusters of terms, which map out a topic and describe a number of patents, which can then be further analysed. The user can navigate freely in the 3D scatterplots and zoom in focussing on details, thus providing for focus and context. The clusters contain terms such as

thickener composition, emulsion, viscosity, protein, pourable etc. Eight patents mention protein. The context in which protein is mentioned can not be determined from the scatterplot. However, the term protein is in a small cluster a significant distance away from the main cluster. This distance indicates that the context in which the term protein is mentioned in a context which is different to the context of the majority of terms in the main cluster. It is at this point it is necessary to be a content expert, in order to be able to postulate what the visualisation is suggesting. It can now be speculated that the outlying point protein is mentioned in the context as an ingredient as opposed to most of the patents in the large aggregate which are assumed to have a context concerning processing and science and technology. By analysing (manually reading) all patents in this dataset this has been confirmed to be correct, i.e. the distances in the plot has accurately visualised differences in the context terms are placed in. . Furthermore, as the terms thickener composition and emulsion are so closely linked it is postulated that both terms are mentioned in the same abstracts. This was confirmed in the two patents mentioning thickener composition (nine patents mentioned the term emulsion). The distance between the two clusters is suggested to be due to a texture difference according to each abstract. The cluster further down is not a pourable texture, but rather a stronger texture, in comparison to the cluster above that concerns pourable textures. This was found to be mostly correct, with the majority of patents in the lower cluster being spreads patents (one pourable product) and vice versa for the pourable cluster (two spread products). The example of this particular trend can quickly be interpreted by a content expert, i.e. that there is a range of textures. By looking at the products in the market place it is clear that Unilever utilises a range of textures for its products. Thus, the implication of this is that by carrying out these types of analyses it is possible to estimate what is going on in our competitive environment. Of course, the other side of the coin is that our competitors can postulate what we are doing, as we are leaving similar marks by our publications, whether papers or patents.Consider Figure 4. As mentioned above, we complement the analysis of terms, by an analysis of authors. Representing each author as a term vector of terms used in any of his/her patents, we can apply dimension reduction with PCA to this high-dimensional data and represent the result as a 3D scatterplot. Additionally, the size of the spheres in the scatterplots indicates the number of patents. The screenshot on the left of Figure 4 clearly indicates that Norton, Brown, and Applequist are the most prolific authors with 59, 32, and 9 patents, respectively. The patent abstracts analysed in this paper were patents published by Norton between January 1974 and June 2003, i.e. Norton is an author in every patent. Brown and Norton were Norton's two most prolific co-authors during this time period. From the text analysis and visualisation above it can now be implicated that these authors have contributed to work that relates to the same type of products. Furthermore, by understanding the science and technology behind these products it is now also possible to hypothesise these authors areas of expertise. Clark and Cain, which are the ouliers in Figure 4 on the right, are two authors which distinguish themselves from the remaining authors in that their abstracts contain the phrases edible plastic/plastified dispersion. The two authors are closely associated as Clark is a co-author in two of Cain's 3 patents, and Clark only has two patents in this data set.

While the above analysis is different to most classical text mining efforts in that it aims to map out topics to understand trends and to represent authors to identify the most prolific authors and their areas of expertise from the terms used in their abstracts, it nonetheless focuses on classical visual datamining techniques with dimension reduction and scatterplots. In the next section we shift the focus towards the analysis of the document and author networks.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

		Abstract (NLP) (Phrases)	1	2	3	4	5	6	7	8	9	10	11	12
		# Records	59	32	16	13	9	6	6	4	4	3	3	3
Inventor(s) (Cleaned)	# Records	1	NORTON, Ian, Timothy	BROWN, Charles, Rupert, Telford	JONES, Malcolm, Glyn	UNDERDOWN, Jeffrey	APPELQVIST, Ingrid, Anne, Marie	MALONE, Mark, Emmett	HOMAN, Jennifer, Elizabeth	DANIELS, Stephen, Charles	Wilding, Peter	Cain, Frederick William	ARONSON, Michael, Paul	Campbell, Iain James
1	20	aqueous phase	20	7	5	5	2				1	1		1
2	14	present invention	14	3	4	2					1	2		
3	13	dispersed phase	13	11	8	2	9	6	6					
4	12	invention	12	3	3	4						1		1
5	12	present	12	7	1	1	2			2			3	1
6	11	continuous fat phase	11	1	3	2						2		
7	10	fat	10	6	3	3				4	2			
8	9	polysaccharide	9	4		4								1
9	9	continuous aqueous phase	9	6	7	4	4	6	6					
10	9	spreads	9	3	1	4								
11	9	water	9	4	3	2				2				
12	9	emulsion	9	6	5		2	4	4		1			1

Figure 1: Screenshot of VantagePoint. Across shows authors and their number of records in this dataset. Down are the NLP phrases extracted from each abstract. At the intersection you find the number of times the phrase has been used by the specific author. By selecting this number you get access to the relevant abstracts.

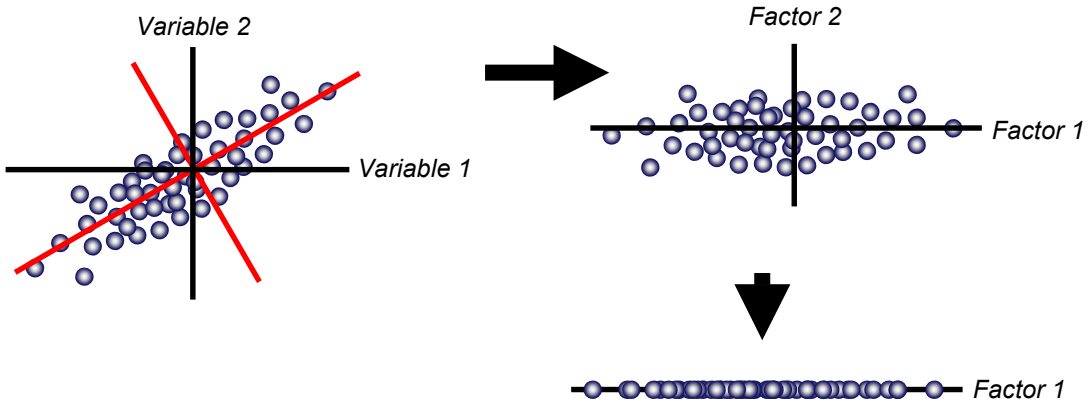


Figure 2: Dimension reduction with Principal Component Analysis (PCA). Documents are represented as term vectors and thus represent a point in a high-dimensional space. PCA rotates the points in space so that the factor catering for most of the variance is aligned to the first axis, the factor explaining most of the remaining variance is mapped to the second axis etc. By considering only the first 2 or 3 factors the original high-dimensional data has been mapped to a 2 or 3D space, which can be visualised as scatterplot.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

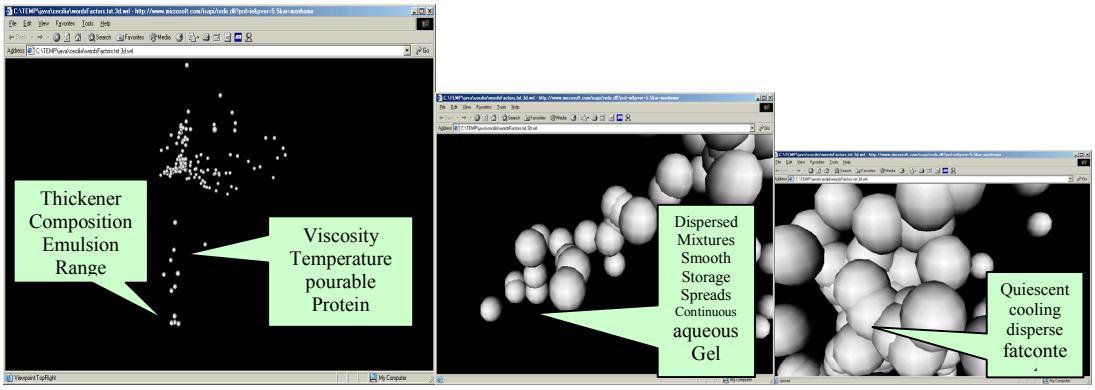


Figure 3: To map out the topic of a set of documents, terms are represented as document vectors, PCA is applied and the result is visualised as 3D scatterplot, which can be interactively explored. The leftmost figure shows two distance term clusters consisting e.g. of terms such as "thickener composition", "emulsion", "range" and "viscosity", "temperature", "pourable", "protein". The terms are mentioned in 2, 9, 2, 4, 6, 3 and 8 patents, respectively.

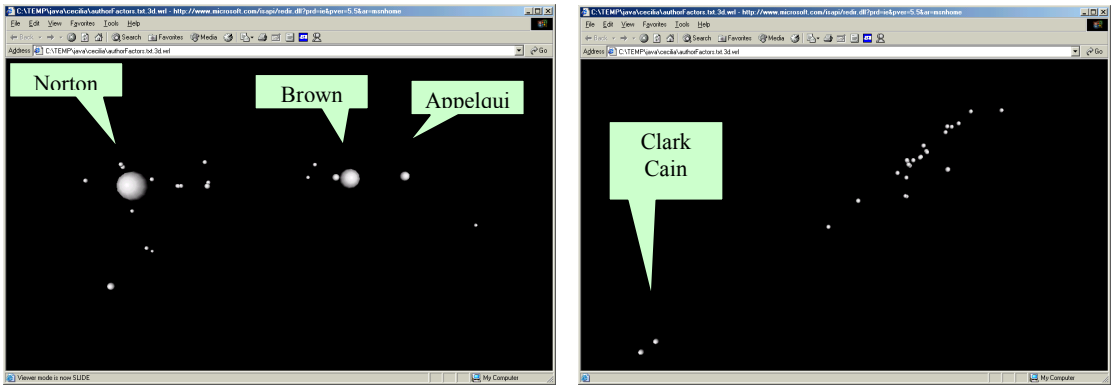


Figure 4: For the patent data set we represent authors as term vectors according to the terms used in the patents they co-authored. We applied PCA to these high-dimensional term vectors and visualised the result as 3D scatterplots. The size of the spheres on the left indicates the number of patents. The screenshot on the left clearly indicates that Norton and Brown are the most prominent authors in the analysed dataset. On the right, the two outliers labelled Clark and Cain indicate that these authors have closely collaborated on topics different from the rest of the authors.

Analysing Author and Document Networks

One approach to mine text data focuses on networks of documents. Google is the best example of how additional information on link structures can be useful to rate the relevance of a document. We pursue a similar line of research by focusing the literature analysis on networks of documents, but also on networks of authors. The networks can be defined in a number of ways. In an author network, there can be a link between two authors if they co-authored a paper or if one cited the other. Similarly, a document network may contain a link between two documents if one cites the other or if a number of crucial terms co-occur between the two documents. Given such large networks how can we analyse and visualise them? To this end, we deploy PSIEYE (Schroeder et al. 2003), an integrated network visualisation and analysis tool. It implements a number of graph-theoretic measures, which rate proteins according to number of interaction partners, overall network location, and local interaction density (cluster index). PSIEYE complements these measures with a novel parameter, interaction rank, which treats interaction as a Markov process, and combines aspects of connectivity and cluster index. Interaction rank pinpoints important nodes given the topology of the whole network.

Consider Figure 5 with a screenshot of PSIEYE. The central panel shows the co-authorship graph for the patent data. The panel is linked by brushing to the list on the left, which displays the authors sorted

by number of co-authors, cluster index, eccentricity, interaction rank, etc. The panel in the middle implements a fish-eye view to maintain a context and the ability to focus on parts of the network. The right panel can display web pages with additional detailed information for an author such as e.g. the patents he or she co-authored. As the graphs are usually big and difficult to clearly represent it is important to swiftly move from an overview to a detailed view with only a few nodes selected by criteria such as number of co-authors. This is achieved by the semantic zoom, which allows to remove nodes from the display and thus focus on a few selected ones, a significant improvement to the visualisation capabilities of vantagePoint. Overall, PSIEYE fulfils all requirements set by Schneiderman's visual information-seeking mantra of overview first, zoom and filter, then details-on-demand (Card, Mackinlay, Shneiderman; 1999).

Broadly, PSIEYE ranks nodes using measures relating to the location of nodes within the network and measures of interactivity. There are two measures of vertex centrality in a network. A vertex's *eccentricity* is the distance to the farthest vertex in the network. The *center* of the network are the vertices, which have the smallest eccentricity. A vertex's *sum of distances* to all other vertices in the network is a measure related to eccentricity. Similar to the center, the *barycenter* are the vertices with the smallest sum of distances. In contrast to eccentricity, the sum of distance averages over all vertices. In Figure 8 on the left, the colour indicates the sum of distance and the lowest sum of distance is achieved by Norton, who therefore forms the barycentre of the network.

Regarding interactivity, PSIEYE provides three measures: connectivity, cluster index, and interaction rank. A vertex's *connectivity* is the number of interaction partners it has. In Figure 6, it has been applied together with semantic zoom to focus on the authors with the greatest number of co-authors. The left shows the five most cooperative authors. Interestingly, this most highly interactive group of authors closely cooperates with each other as they form a clique, i.e. all of them have co-authored patents with each other. Overall, these authors can be considered as the core and most important, at least most prolific, authors in the network. The right figure shows an example of visual querying. The colour is mapped to the number of co-authors. The colour scale ranges from blue (many co-authors) to red (few co-authors).

The connectedness of a node's direct neighbourhood is measured by *cluster index* (Watts and Strogatz 1998), which is defined as the number of interactions between a vertex's neighbours divided by the total number of possible interactions between them. A cluster index of 0 means that none of a vertex's neighbours interact, whereas 1 indicates that they all interact with each other. In Figure 7, the left figure shows the author network coloured by cluster index, where blue indicates a high cluster index and red a low one. Norton, who has many co-authors, does not score well in terms of cluster index, as not all of his co-authors have co-authored patents. The large number of authors with a moderate number of co-authors score well in terms of cluster index indicating also the teams, in which the work has been carried out.

Both connectivity and cluster index have shortcomings: connectivity does not consider interactions in a vertex's neighbourhood; cluster index favours low connectivity vertices. Interaction Rank addresses these shortcomings. Interaction rank treats networks as Markov processes: If two vertices are linked in the network, then there is a certain probability they will interact at any one time. Using the simplification of an unweighted interaction network we get a $1/|N(v)$ chance for vertex v to interact with a neighbour w, where $|N(v)|$ is the size of the set of v's neighbours. If we enumerate all vertices from 1 to n, we can capture this Markov process as a matrix M. The interaction rank is defined as the steady state of M. In general, the interaction rank is the better, the more interaction partners there are and the better connected a vertex's neighbourhood is. Thus interaction rank combines aspects of connectivity and cluster index on a global scale.

Consider Figure 8 on the right. Similar to author networks we can consider term and document networks. The figure shows such a term network based on co-occurrence. The network puts terms such as process, shear, dispersion, edible dispersion, another aspect, concentration level, continuous fat phase, weight, advantage, invention, mouth, water, agents, gel setting temperature, least two, low fat content, present invention into the same context. It is clear that only some of these terms are useful, while other are of general nature and should be removed from the analysis. The colour of the nodes represents there interaction rank. Blue indicates a high interaction rank, red a low one. In this respect, process and preparation are the defining terms for this term cluster.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics*, held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

Finally, PSIEYE allows one to select individual nodes and to explore their neighbourhoods. The right figure in Figure 7 shows an example for such visual querying. It shows that Aronson co-authored patents with 5 authors., who in turn co-authored with each other, thus also leading to a good cluster index of Aronson.

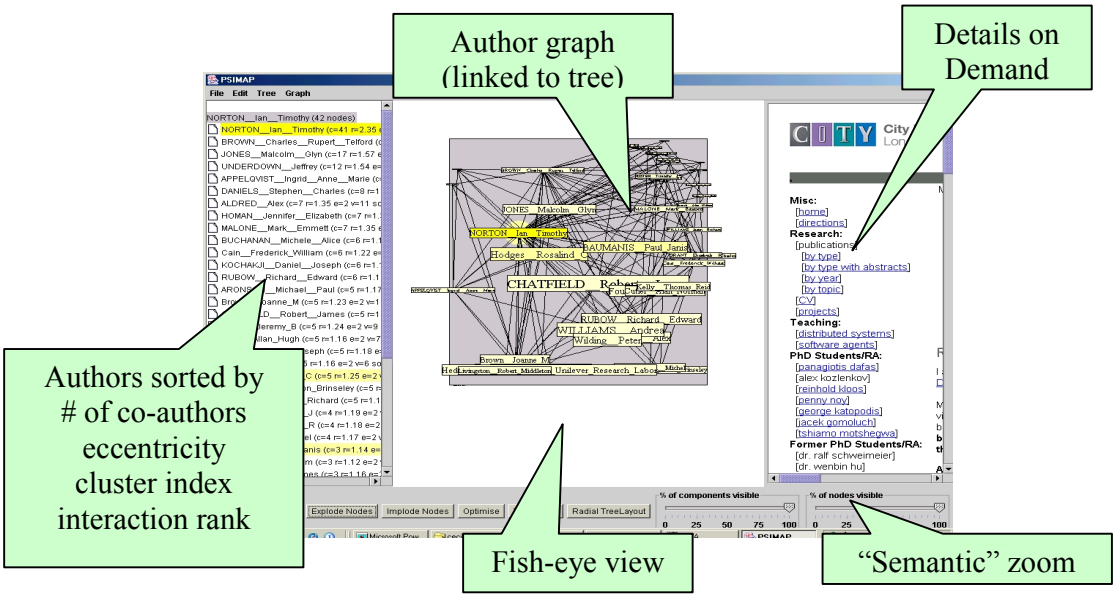


Figure 5: A screenshot of PSIEYE.

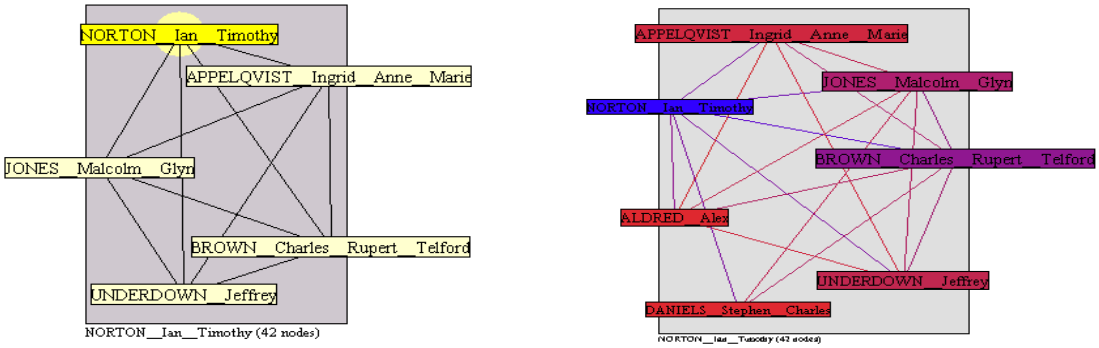


Figure 6: The authors with most co-authors have been selected using PSIEYE's semantic zoom feature. The left shows the 5 most cooperative authors, who also closely cooperate with each other as they form a clique, i.e. all of them have co-authored patents with each other. The right figure shows an example of visual querying. The colour is mapped to the number of co-authors. The colour scale ranges from blue (many co-authors) to red (few co-authors).

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics*, held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

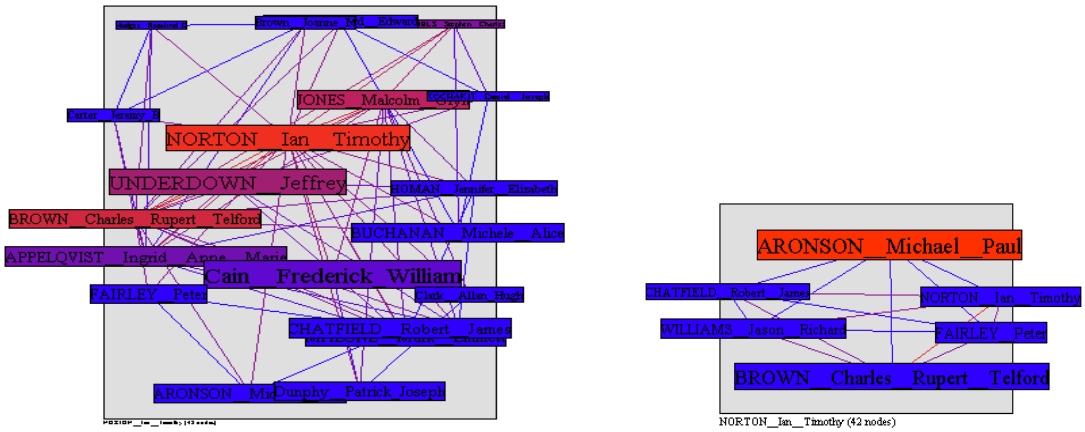


Figure 7: Cluster index measures the connectivity of a node's neighbourhood. 0 indicates that none of the neighbours interact, 1 means all interact. The left shows the author network coloured by cluster index, where red indicates a low cluster index and blue a high one. The right figure shows an example for visual querying. For a selected node its neighbourhood can be visualised. It shows that Aronson co-authored patents with 5 authors, who in turn co-authored with each other, thus also leading to a good cluster index of Aronson.

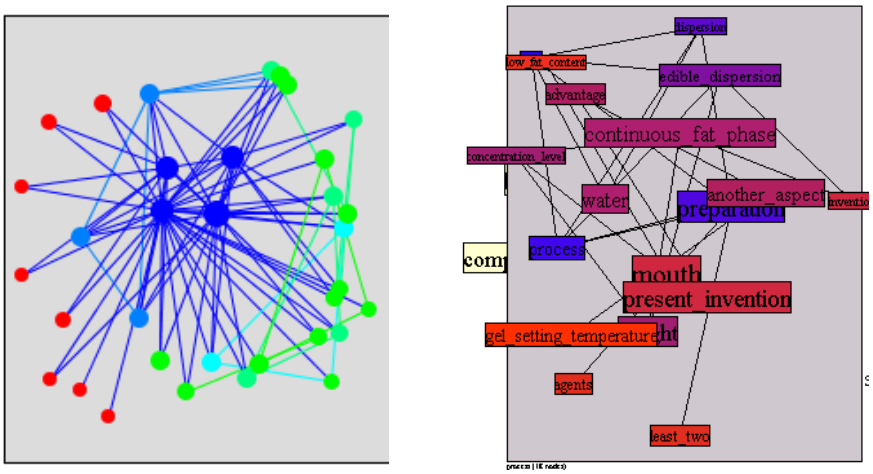


Figure 8: **Left**: The whole author network with colour indicating the sum of distance to all other authors. The colour grading ranges from blue (low) via green/yellow (medium) to red (high). Norton is the node with the lowest sum of distance and this forms the barycentre of the whole network. **Right**: Similar to author networks we can consider term and document networks. The figure shows such a term network based on co-occurrence. The network puts terms such as process, shear, dispersion, edible dispersion, another aspect, concentration level, continuous fat phase, weight, advantage, invention, mouth, water, agents, gel setting temperature, least two, low fat content, present invention into the same context. It is clear that only some of these terms are useful, while other are of general nature and should be removed from the analysis. The colour of the nodes represents there interaction rank. Blue indicates a high interaction rank, red a low one. In this respect, process and preparation are the defining terms for this term cluster.

Conclusion

With the tremendous growth of the biomedical literature, text mining has become paramount. In this paper we have presented a case study, in which we applied visual datamining to a selected set of Unilever patents. The goal of this effort is to map out current topics and developments by identifying term clusters in recent patents and to identify the most relevant authors behind these emerging trends. To achieve this goal, we first applied dimension reduction with principal component analysis to terms represented document vectors and authors represented as term vectors. We then visualised the result as 3D scatterplots, which we interactively explored. We complemented this approach by focusing on term

and author networks. We mined these networks using PSIEYE, which provides a number of graph-theoretic measures besides the visualisation of the networks.
A number of conclusions can be drawn: First, the focus on networks of authors is an important emphasis in text mining analysis as research is a social process. It is also important that author networks have special properties (Watts and Strogatz 1998), which require appropriate measures for analysis. To this end, cluster index (Watts and Strogatz 1998) and interaction rank (Schroeder et al. 2003) are important concepts beside the basic measures such as connectivity and eccentricity. Second, it is important to stress that visualisation (Card, Mackinlay, Shneiderman 1999) is an important part of our mining effort. Visual datamining puts the human back into the loop and makes datamining an interactive exploratory process aimed at supporting hypothesis formulation. In particular, tools need to support this process by providing means for showing overviews, to filter/zoom, to display detail-on-demand, to visually querying, to link views, etc. (Card, Mackinlay, Shneiderman 1999). SpaceExplorer (Schroeder et al. 2001) and PSIEYE (Schroeder et al. 2003), which were used in this case study, both provide these capabilities. Third, expertise in the scientific/technical field under investigation is necessary in order to interpret views and draw conclusions/make hypothesis about what the analysis is suggesting.
The case study provides valuable insights on how to approach visual datamining of bibliometric networks. However, there are a number of limits. In our current work, we relied on tools such as VantagePoint, whose natural language processing capabilities are limited. While e.g. temperature and temp can be equated by stemming, other terms are equivalently used in different communities (e.g. the gene names YOR141c and ARP8 are equivalent) and cannot be equated. Most important, it will be important to distinguish types of words and apply templates in parsing such as process (e.g. heating) is applied to a substance (e.g. water) , which is in a certain state (e.g. liquid). There are text analysis tools on the market that have gone some way towards aiming to solve this issue, but huge manual input are generally needed.

References

(Blaschke C. et al. 1999) Automatic extraction of biological information from scientific text: protein-protein interaction. *Proceedings of the AAAI conference on Intelligent Systems in Molecular Biology*, 60-7
(Breitzman and Mogee 2002) Journal of Information Science, 28(3), 2002, pp187-205
(Buter and Noyons 2001) Scientometrics, 51, pp55-67
(Card, Mackinlay, Shneiderman 1999) Readings in Information Visualization: Using Vision To Think Morgan Kaufmann
(Feldman R. et al. 2003) Mining the biomedical literature using semantic analysis and natural language processing techniques. *Biosilico* 1(2):69-79
(Friedman C. et al. 2001) GENIES: a natural language processing system for the extraction of molecular pathways from journal articles. *Proceedings of the International Confernceon Intelligent Systems for Molecular Biology,* 574-82
(Porter, A.L; Kongthon, A; Lu, J-C 2003) Scientometrics, vol 53, no 3, (2002), pp351-370)
(Schroeder, Gilbert, van Helden, Noy, 2001) Approaches to Visualisation in Bioinformatics: from Dendrograms to Space Explorer. In *Information Sciences: An International Journal* 139(1):19-57
(Schroeder et al. 2003) PSIEYE: A tool for the graph-theoretic analysis of protein interaction networks. Submitted
(Swanson, D.R 1987). Two medical literatures that are logically but not biographically connected. *J.Amer.Soc.Inform.Sci*. 38(4), 228-233
(Smalheiser, N.R, Swanson, D.R 1998) Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. *Comput.Method.Program.Biomed*, 57(3), 149-153
(Tanabe, L. et al. 1999) MedMiner: internettext-mining tool for biomedical information, with application to gene expression profiling. *BioTechniques* 27, 1210-7
(Thomas, J. et al. 2002) Automatic extraction of protein interactions from scientific abstracts. *Proceedings of the Pacific Symposium on Biocomputing*, 538-49
(Watts and Strogatz 1998). Collective dynamics of small-world networks. Nature}, vol 393 pp440-2,

Towards Recognizing Domain and Species from MEDLINE Publications*

Alexander K. Seewald[1]

Austrian Research Institute for Artificial Intelligence, Freyung 6/VI/7,
A-1010 Vienna, Austria alexsee@oefai.at

Abstract. In text mining for bioinformatics, one important bottle-neck is the availability of high-quality tagged corpora. We introduce a novel approach to learn extraction patterns from pre-classified but untagged corpora, which are easier to generate automatically. We apply our approach to two datasets derived from SWISS-PROT plus associated MEDLINE references. In both experiments a Ripper-like rule learner, JRip, is competitive to all other learners; outputs a manageable number of understandable rules; and performs comparably to a human domain expert investigating the same task. Based on our results, we note weaknesses and strengths of both human model and machine learning approaches, which indicates that they have distinct areas of expertise. Our approach may be used to generate initial rulesets for information extraction, to be iteratively refined by domain experts; or as a stand-alone approach with some losses in precision.

1 Introduction

In text mining for bioinformatics, one important bottle-neck is the availability of high-quality tagged corpora. Creating a pre-tagged corpus entails high workload for domain experts, but a corpus for a specific domain can usually not be directly transferred to other domains. Disagreement between domain experts even for basic issues such as extent of protein and gene names [4] further complicate the issue.

In this paper, we propose and evaluate an alternative approach to information extraction from pre-classified, but untagged corpora; in our case created automatically from the SWISS-PROT and MEDLINE databases. We reformulate the information extraction problem as learning problem where each distinct class represents a unique slot value to be extracted from the document. Thus, we are also able to learn synonyms and utilize partial evidence for slot values. On the other hand we require that the full list of values for a given slot is known a priori. Also, the scalability of our approach towards very high number of slot values remains to be investigated.

We will proceed to show that our approach is competitive to state-of-the-art approaches such as Support Vector Machines and Decision Trees; results in a

* In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003

manageable number of easily understood extraction rules for each slot value and even performs competitively to a human domain expert investigating the same extraction task. The automatic approach and the domain expert each has its own area of expertise: domain experts are better at creating rules with high precision at cost of lower recall, while the automatic approach is biased towards creating rules with lower precision and higher recall. Still, the automatic approach gives better models than our domain expert in a third of cases; and generally manages the trade off between precision and recall much better.

2 Databases

SWISS-PROT [5, 1] is one of the largest protein databases. All its entries are created by biologists and are continually updated, extended and corrected. Thus the quality of the entries is considered to be quite high, which is not yet the case with automatically generated databases. On the other hand, the number of available examples is much smaller but still sufficient for our experiments. SWISS-PROT is biased towards better described proteins; also, the attached MEDLINE references are selected to be representative of each entry. Both of this may lead to overestimation of true performance, which should be kept in mind; but this bias is in our opinion a small price to pay for such a wealth of data with excellent quality.

We obtained a recent snapshot of the SWISS-PROT database, consisting of 121,745 entries. We also obtained all referenced MEDLINE entries, yielding 83,044 documents. For our experiments, we focus on the OS field which encodes taxonomic information in the form of organism, or species, where the protein is present. We are interested in predicting domain and species of an organism from the associated MEDLINE documents.

All in all, there are 7,803 distinct species referenced in our SWISS-PROT snapshot, on average 1.1 ± 0.3 per entry. It is thus not unlikely that a protein appears in more than one species; however, since then the learning problem is not well defined, we removed these entries. We also removed those entries without MEDLINE references (10.6%), and used only the first referenced MEDLINE entry[1] in each of the remaining cases, on the premise that it is the most relevant document. 104,747 entries remain after these prior selections, each consisting of a species value and an associated MEDLINE publication. These form the basis for our experiments.

From each MEDLINE publication, we chose to use title, abstract and MeSH terms. Throughout this paper, we use word occurrence vector representations.[2] We removed all characters except whitespace, lower- and uppercase letters, numbers, the dash (-) and the prime ('). Each continguous sequence of non-

[1] On average, each SWISS-PROT entry references 1.9 ± 2.0 MEDLINE entries.

[2] I.e. one binary attribute for each word which encodes if it appears in the text (1) or not (0) – a widely used representation in text mining, which of course loses information on word sequence.

Table 1. Results for predicting species domain. Acc.CV shows two-fold cross-validation on 5% of data; Acc.Test shows performance of trained model from 5% data on the remaining 95%. Approximate execution times are also given.

Classifier	Acc.CV	Exec	Acc.Test	Exec
`ZeroR` (baseline)	49.8%	1.4s	47.6%	0.2s
`OneR`	85.3%	1m	84.2%	0.1s
`NaiveBayes-K`	93.6%	5m	93.3%	176s
`J48`	96.2%	36m	96.4%	0.33s
`PART`	96.4%	36m	96.8%	0.4s
`JRip`	97.0%	108m	97.5%	0.1s
`Logistic`	97.4%	285m	96.4%	7s
`SMO-RBF`	97.5%	17m	97.6%	842s
`SMO`	97.7%	2m	97.8%	2.6s

whitespace characters framed by whitespace[3] is considered a word. For simplicity, we combined all words from title, abstract and MeSH terms into a single word vector.

3 Species Domain

As preliminary step, we investigated the task of predicting the species domain, or kingdom – one of Archaea, Bacteria, Eukaryota or Virus – from MEDLINE documents. We chose to restrict our word occurrence vector representation to the most-frequent 1044 words, which form the input attributes for our four-class learning problem.

Initially, we used 5% of our examples with two-fold cross-validation[4]. For validation, all classifiers were retrained on 5% data and evaluated on the remaining 95% for validation. A selection of common classifiers from WEKA[5] was chosen: `ZeroR` is a simple baseline classifier which always predicts the most common class from training data, independent of the specific example to be classified; `OneR` is a simple classifier which learns one rule based on a single attribute's values; `NaiveBayes` is another well-known classifier based on the Bayes Theorem for conditional probability; `J48` is a decision tree learner based on C4.5R8, `PART` corresponds to c4.5rules and generates rulesets from all paths within a C4.5 decision tree; `JRip` is a rule learner similar in spirit to the commercial rule learner Ripper; `SMO` and `SMO-RBF` are support vector machine implementations with linear resp. Radial-Basis-Function kernels. Space restrictions prevent us from giving more details on strengths and weaknesses of the various approaches

[3] The character before the beginning and after the end of text is also considered whitespace.

[4] This is equivalent to splitting the data into two equally sized halves (retaining class distributions); using one part for training, the other part for testing; then swapping the sets, repeating train/test and averaging over the two test-set results.

[5] The source code of WEKA is available at `www.cs.waikato.ac.nz/~ml/weka`

Table 2. Results for predicting top twenty species. Acc.CV shows two-fold cross-validation on the complete dataset. Approximate execution time is also given.

Classifier	Acc.CV	Exec
`ZeroR` (baseline)	19.0%	12s
`OneR`	26.5%	52m
`NaiveBayes-K`	76.4%	10h
`JRip`	88.9%	312h
`SMO`	89.3%	29h

- however, as we shall see, most of them perform similarily, so we can afford to focus on a single rule-based approach, `JRip`.

Results can be found in Table 1. We see that more than half of our classifiers perform similarily.[6] According to runtime, we see that the support vector implementation `SMO` is both much faster than `JRip` and slightly better; however, for the purposes of communicating our models to domain experts, `JRip` is much better suited. For illustration, Figure 2 shows the model which was obtained by `JRip`.

4 Species Top Twenty

We then decided to predict the top twenty largest species appearing in SWISS-PROT, 42.1% of all entries. This gives us 43,761 examples. We chose to restrict ourselves to the top 3,834 most frequent words as input attributes for our learning algorithms. For this task, we also obtained a domain expert's model for comparison. The expert utilized multi-word patterns, or phrases.[7] To ensure a fair comparison with our approach, we counted the number of rule matches and chose the rule with maximum number of matches. Ties were resolved in favor of the more common class according to training data.[8]

We chose a subset of classifiers from the previous experiment, based on considerations of runtime, accuracy and understandability. Experimental results are found in Table 2. `JRip` performs again slightly worse than `SMO`. Again, we prefer `JRip` because of its more understandable and small rule set (172 rules for twenty classes), even if its training takes an order of magnitude longer.

For a more detailed analysis, we compared each species separately, see Table 3. Maximum values are marked **bold** in this table. We can see at once that

[6] It deserves mention that the very simple model by `OneR`, *IF document contains word 'Bacterial' => Bacteria ELSE Eukaryota*, already improves significantly over the baseline.

[7] Due to the limits of word vector representations information on word sequence is lost and thus multi-word patterns cannot be learned with our approach, which may create a significant disadvantage for our model.

[8] We expect `JRip` to use similar optimizations.

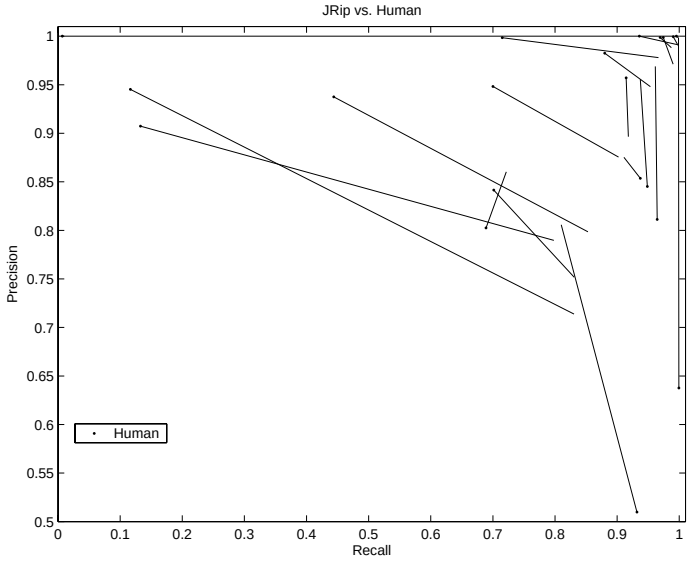


Fig. 1. Results for predicting top twenty species. X shows recall, Y shows precision. The dot (.) shows the domain expert's result, the other end of the line shows `JRip`'s result. Each line corresponds to one species.

`JRip` performs better in terms of recall and F-measure[9], while the domain expert's model usually offers higher precision.

Figure 1 shows the same data in graphical form. In most cases, the domain expert improves precision at the cost of recall, i.e. the lines point[10] to the top left. Since domain experts have told us that they emphasize precision over recall, these results are not surprising. However, in seven cases `JRip` offers better precision than the human model; and in some cases the domain expert's model yields very low recall – once even less than 1%. It should be mentioned that this reflects shortcomings of the domain expert's model rather than the domain expert's judgement.

`JRip` also manages the trade off between precision and recall better, which can be seen by the F-measure being higher in three-fourths of cases. Average precision, recall and F-measure are also uniformly higher and the standard deviation of these values are uniformly lower than those of the domain expert, which also confirms this observation.

[9] The F-measure is a simple combination of recall and precision, i.e. $2 * \frac{r*p}{r+p}$.

[10] The dot at one end of the line shows the performance of our domain expert. We consider the lines to point towards this dot.

Table 3. Results for predicting top twenty species. p shows precision, r shows recall, F shows the F-measure, $2 * \frac{r*p}{r+p}$. On the left, we see results by `JRip`, on the right those from the domain expert's model. Best values are shown in **bold**. Average and standard deviation for all columns is also given.

Species Name	JRip			Human		
	p	r	F	p	r	F
Homo sapiens (Human)	**80.57**	81.00	**80.79**	50.99	**93.18**	65.91
Xenopus laevis (African clawed frog)	75.17	**83.10**	**78.94**	**84.15**	70.14	76.51
Escherichia coli	**96.90**	96.13	**96.51**	81.13	**96.45**	88.13
Caenorhabditis elegans	**87.53**	91.09	89.27	85.37	**93.73**	**89.36**
Haemophilus influenzae	99.04	**99.83**	99.43	**99.94**	99.03	**99.49**
Arabidopsis thaliana (Mouse-ear cress)	97.80	**96.66**	**97.23**	**99.85**	71.49	83.32
Bos taurus (Bovine)	71.38	**83.05**	**76.77**	**94.52**	11.64	20.72
Bacillus subtilis	98.82	**98.74**	**98.78**	**99.87**	96.93	98.38
Archaeoglobus fulgidus	**100.00**	**100.00**	**100.00**	**100.00**	0.69	1.36
Mus musculus (Mouse)	78.97	**79.81**	**79.39**	**90.73**	13.26	23.15
Mycobacterium tuberculosis	99.85	**99.77**	**99.81**	**100.00**	99.55	99.77
Salmonella typhimurium	89.65	**91.81**	90.71	**95.71**	91.45	**93.53**
Synechocystis sp. (strain PCC 6803)	99.13	**99.67**	**99.40**	**100.00**	93.57	96.68
Pseudomonas aeruginosa	97.12	**99.02**	98.06	**99.87**	97.43	**98.64**
Saccharomyces cerevisiae (Bakers yeast)	**95.55**	93.73	**94.63**	84.52	**94.85**	89.39
Methanococcus jannaschii	**99.87**	99.87	**99.87**	63.76	**99.93**	77.85
Gallus gallus (Chicken)	79.85	**85.31**	**82.49**	**93.75**	44.39	60.25
Rattus norvegicus (Rat)	**86.04**	**72.16**	**78.49**	80.24	68.90	74.14
Schizosaccharomyces pombe (Fission yeast)	94.81	**95.35**	**95.08**	**98.25**	87.99	92.84
Drosophila melanogaster (Fruit fly)	87.55	**90.23**	**88.87**	**94.83**	70.00	80.55
Max	7	**15**	**16**	**14**	5	4
Avg	**90.78**	**91.82**	**91.23**	89.87	74.73	75.50
± stdDev	**9.31**	**8.35**	**8.62**	13.22	32.03	28.56

5 Related Research

[6] introduces a system to generate extraction patterns from untagged, but preclassified corpus by exhaustively generating all extraction patterns, which are then manually selected. Our approach is related in that we extract patterns based on preclassified documents without a manually tagged corpus. No manual inspection of extraction patterns are necessary for our approach which makes it more efficient.

[7] introduces a support vector machine classifier to classify sub-cellular location. In their case, the machine learning approach still performs significantly worse than the best manually created rulesets from [3]. SVM models are also hard to visualise and understand, while our approach generates understandable rulesets of manageable size.

[2] also investigate the prediction of sub-cellular location, among other tasks. While using a pre-tagged corpus at first, they later resort to weakly labeled training instances, generated from online databases. Their approach is somewhat

```
(archaeon) => domain=A (163.0/0.0)
(Archaeal) and (!Bacterial) => domain=A (92.0/0.0)
(Halobacterium) => domain=A (22.0/3.0)
(archaebacterium) and (Bacterial) => domain=A (7.0/0.0)
(Methanobacterium) => domain=A (6.0/2.0)
(Archaea) and (!Proteins) => domain=A (2.0/0.0)
(Viral) => domain=V (351.0/18.0)
(Bacterial) and (!Animal) => domain=B (1665.0/14.0)
(Bacterial) and (!RNA) and (!cerevisiae) => domain=B (211.0/10.0)
(!Animal) and (Escherichia) and (!Proteins) and (!Fungal) and (!cDNA) => domain=B (26.0/2.0)
(!Animal) and (bacteria) and (!cDNA) => domain=B (19.0/3.0)
(strain) and (!Fungal) and (!Proteins) and (!2) => domain=B (17.0/1.0)
(!Animal) and (cyanobacterium) => domain=B (9.0/1.0)
(Bacteria) and (!Animal) => domain=B (6.0/1.0)
(Frames) and (operon) => domain=B (4.0/1.0)
(Salmonella) => domain=B (2.0/0.0)
(Streptomyces) and (!at) => domain=B (5.0/0.0)
(Anabaena) => domain=B (3.0/0.0)
(bacterium) => domain=B (5.0/1.0)
(Bacillus) and (!Animal) => domain=B (5.0/1.0)
(pneumoniae) => domain=B (2.0/0.0)
 => domain=E (2534.0/20.0)
```

Fig. 2. JRip's model for species domain. (word) encodes word occurrence and (!word) word non-occurrence. E.g. second rule reads like this: If the (title, abstract, MeSH terms) of a MEDLINE entry contains *Archaeal* and not *Bacterial*, predict domain archaea (domain=A). This rule is corrent for 92 examples and incorrect for none (92.0/0.0). The last line is the default rule, which is chosen if no other rule matches.

similar to ours, in that our training instances are also weakly labeled (i.e. we do not know where *exactly* the required information is present in the text). They also investigate using relational learning for this task, and find that it improves precision.

6 Conclusion

We have reformulated a problem of information extraction within BioInformatics as learning problem. More specifically, we have aimed to extract organism names (domain and species) from MEDLINE documents related to proteins.

The reported results show that our approach is competitive to a model by a human domain expert, both having distinct areas of expertise: Humans are better at creating rules with high precision at cost of lower recall, while our approach is well suited to create rules with lower precision and higher recall. This is in accordance with our domain expert's preference for precision. Still, in a third of cases our approach yields models with better precision than the domain expert's models. The focus on precision over recall may impair the domain expert's ability for a reasonable trade off between precision and recall[11] while our approach manages the trade off fairly well.

[11] Recall as low as 0.7% was observed, for small or even nonexisting improvements in precision

Our approach returns an easily understandable ruleset of manageable size[12] and could either be used for generating initial rulesets for iterative refinement by domain experts; or as a stand-alone approach with some losses in precision.

We intend to investigate the scalability of our approach to thousands of slot values in the future; and also see how well it performs with data from other sources. We will also look into relational learning approaches and support vector machines with string kernels; and other interesting problems within the field BioInformatics.

Acknowledgements

This research was supported by the European Union under project no. QLRI-2002-02770 (BioMinT). The Austrian Research Institute is supported by the Austrian Federal Ministry of Education, Science and Culture. We want to thank our domain expert Violaine Pillet from the Swiss Institute of Bioinformatics for sharing with us her extraction rules for SWISS-PROT species.

References

1. Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, Nucleic Acids Research, 31(1):365-370.
2. Craven M., Kumlien J. (1999) Constructing Biological Knowledge Bases by Extracting Information from Text Sources, Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99).
3. Eisenhaber F., Bork P. (1999) Evaluation of human-readable annotation in biomolecular sequence databases with biological rule libraries, BioInformatics (15) 7-8, pp.528-535.
4. Franzen K., Eriksson G., Olsson F., Asker L., Liden P., Coester J. (2002) Protein names and how to find them, International Journal of Medical Informatics, Vol. 67/1-3, Special Issue on NLP in Biomedical Applications, pp.49-61.
5. O'Donovan,C., Martin,M.J., Gattiker,A., Gasteiger,E., Bairoch,A. and Apweiler,R. (2002) High-quality protein knowledge resource: SWISS-PROT and TrEMBL. Briefings in Bioinformatics, *3*, 275–284.
6. Riloff,E. (1996) Automatically Generating Extraction Patterns from Untagged Text, in Proceedings of the 13th National Conference on Artificial Intelligence, AAAI Press/MIT Press, Cambridge/Menlo Park, pp.1044-1049.
7. Stapley B.J., Kelley L.A., Sternberg M.J.E. (2002) Predicting the Sub-Cellular Location of Proteins from Text Using Support Vector Machines, Pacific Symposium on Biocomputing (7), pp.374-385.

[12] 22 rules for species-domain, see Figure 2; 172 rules for species-top20.

Evaluating Protein Name Recognition: An Automatic Approach*

Alexander K. Seewald[1]

Austrian Research Institute for Artificial Intelligence, Freyung 6/VI/7,
A-1010 Vienna, Austria alexsee@oefai.at

Abstract. In some domains, named entity recognition might be considered a solved problem. This does not hold for biological text mining, where protein and gene name recognition are still open research problems [4, 6]. In this paper, we compare two current approaches to the problem of protein name recognition, KeX [5] and Yapex [4]. Unlike manual evaluation which relies on domain experts' judgement concerning position and extent of all relevant names and entails a high workload, our comparison methodology is fully automatic. Our results agree with previous manual evaluations of KeX and Yapex which validates our approach.

1 Introduction

While for some tasks named entity recognition may be considered a solved problem, this is not the case for protein name recognition in biological text mining. Although useful results can be achieved with a fixed static dictionary [1, 8] thousands of new papers appear daily and no fixed dictionary is expected to be accurate for long. Current implementations of protein name recognizers rely on heuristics and proprietary text processing, and do not yet learn, although they are quite able to recognize previously unseen protein names. Machine learning approaches to this problem are also upcoming[1]. In this paper, we focus on two approaches which are already available, KeX[2] [5] and Yapex[3] [4].

A reasonable approach to compare protein name recognizers is to measure them against the gold standard – a domain expert which marks up all protein names in a set of test papers. While this is not completely unproblematic – domain experts may sometimes disagree to the extent of protein names or what kinds of names are considered valid [4] – it entails a high workload which is better put to more productive use. Therefore, we propose a fully automatic approach to comparison, using the full SWISS-PROT [7, 2] database and associated MEDLINE[4] publications as approximation to the gold standard. Our approach allows

* In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics,* held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003

[1] GAPSCORE, http://bionlp.standford.edu, which we look forward to investigate.

[2] The KeX source code has been made freely available by the authors at http://www.hgc.ims.u-tokyo.ac.jp/service/tooldoc/KeX/intro.html

[3] Yapex is available via web form http://www.sics.se/humle/projects/prothalt/yapex.cgi and utilizes a commercial tagger which is subject to licensing.

[4] MEDLINE is a bibliographic database owned by the U.S. National Library of Medicine and can be searched via PubMed: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

to utilize several orders of magnitude more text for evaluation than manual approaches.[5] While we cannot offer specific answer keys, and our approach is only based on positive examples of protein names, our results agree with an earlier comparison of KeX and Yapex by classical means.

2 Databases

SWISS-PROT is one of the largest proteomics databases. All its entries are created by biologists; continually updated, extended and corrected.

For our experiments, we obtained a recent snapshot of the SWISS-PROT database, consisting of 121,745 entries. We also obtained all referenced MEDLINE entries, yielding 83,044 documents. For our experiments, we focus on the DE field which encodes protein names and synonyms; and use it as standard against which different protein name recognizers are measured, using the referenced MEDLINE entries as input. Since MEDLINE publications are expected to reference other proteins as well, we also measured each recognizer against a full list of proteins names generated from the whole SWISS-PROT database.

All in all, there are 95,982 unique protein names and synonyms referenced in our SWISS-PROT snapshot. This is less than the total number of entries, so not all proteins have an unique name. Non-specific entries such as 111 kDa protein, which refers to any protein with a specific molecular weight, explain this discrepancy. On average, each SWISS-PROT entry includes 2.34 ± 1.52 protein names and synonyms.

From each MEDLINE publication, we chose title and abstract. For KeX, we compiled KeX and executed it locally; for Yapex we are obliged to Kristofer Franzen, who supervised the processing at their site. Both KeX and Yapex are quite fast; parsing all 83,044 MEDLINE documents took about a working day, i.e. 8-10h. Thus, parsing all new MEDLINE entries in real-time seems feasible with either of these approaches.

Since SWISS-PROT is biased towards better characterized proteins[6]; and the referenced MEDLINE publications are usually more representative that a random selection, our approach may overestimate the true performance of protein name recognizers. Even in that case, a relative comparison is still meaningful.

3 Experiments

Extracting the protein names from KeX and Yapex output is quite simple – both use html-like tags which enclose protein names and parts. While KeX always outputs a single level of tags, those by Yapex are sometimes recursively nested in multiple levels. Therefore we chose to use the full word sequence within the outermost level of tags for Yapex, to prevent duplication of parts of protein names which may bias our evaluation. We match each protein name from KeX and Yapex separately against SWISS-PROT.

Matching recognized protein names to SWISS-PROT protein names is less obvious. Inspired by [4], we have considered three matching schemes:

[5] We *do* indirectly rely on SWISS-PROT curators painstakingly collecting protein names – so even our approach heavily relies on reusing human expertise.

[6] For example, we were unable to find the yeast prion proteins Rnq1p, Sup35p and Ure2p

Table 1. This table shows results for KeX and Yapex with different comparison methodologies, as well as vs. all of SPROTs protein names or vs. only those which are associated to a given MEDLINE publication. Better values are shown in **bold**.

	all SPROT		only SPROT refs	
	Yapex	KeX	Yapex	KeX
Strict	**0.202**±0.401	0.097±0.296	**0.077**±0.267	0.038±0.192
PNP	**0.606**±0.423	0.529±0.374	**0.328**±0.426	0.221±0.346
Sloppy	0.732±0.443	**0.775**±0.420	**0.415**±0.493	0.354±0.478

- *Strict*, i.e. matching of the whole string. We observed slight differences in capitalization and therefore chose to use case-insensitive matching.
- *Protein Name Parts* (*PNP*), i.e. a degree of match between 0 and 1 as ratio of those words within the recognized protein which are matched to any words in a given SWISS-PROT template. Word matching is done both with lowercase and uppercase first letter to account for differences in capitalization. We rely on the protein recognizer to emphasize word boundaries, which we consider to be whitespace, or parentheses wrapped in whitespace.
- *Sloppy*, i.e. a match is counted even if only a single word matches. This is equivalent to $PNP>0$.

The *Left* and *Right* schemes from [4] are not applicable since we have no information on the position of a probable match. Furthermore, our protein list is not exhaustive, so we have no complete data on non-matches.

Along an orthogonal dimension, we compared each protein name recognizer against a full list of all unique SWISS-PROT protein names (*all SPROT*); or only against those SWISS-PROT names which are associated with the given MEDLINE publication (*only SPROT refs*). The latter is considered to be more specific – it is expected that at least one entry from this list appears in every MEDLINE publication. This is almost the case for Yapex with 0.81±1.93 entries per publication; and less so for KeX with 0.46±1.33. Therefore, not all synonyms are recorded in SWISS-PROT, which agrees with our domain experts stated opinion that the protein lists from SWISS-PROT are not exhaustive.

Table 1 gives the results. We give averages and standard deviations over the match values from all recognized proteins, based on the three matching schemes. A match is considered 1 and no match 0 for *Strict* and *Sloppy*; and a ratio between 0 and 1 for *PNP*. We see that *Sloppy* is the only matching scheme where KeX performs comparable to Yapex, which was also found in [4]. Under the other matching schemes, Yapex performs better. If we consider *only SPROT refs*, Yapex always performs better than KeX. The high standard deviation for most entries indicates that there are large fluctuations in the proportion of correct matches from abstract to abstract which may bias the arithmetic average[7], so we now take a closer look at the specific match values from *PNP*.

We take a closer look at the distributions of *PNP* match values in Figure 1. Alas, the most obvious patterns are just artefacts of ratio scores – short protein names mean a smaller number of possible scores, since each word can only match or not. Since there are many short protein names, the values tend to cluster near simple ratios, such as $\frac{1}{2}, \frac{1}{3}, \frac{2}{3}$ and so on. For longer protein names, the number of possible scores increases, so longer names tend to distribute over a wider area and are more susceptible to disappear into the background.

[7] Using the median would probably have been more appropriate.

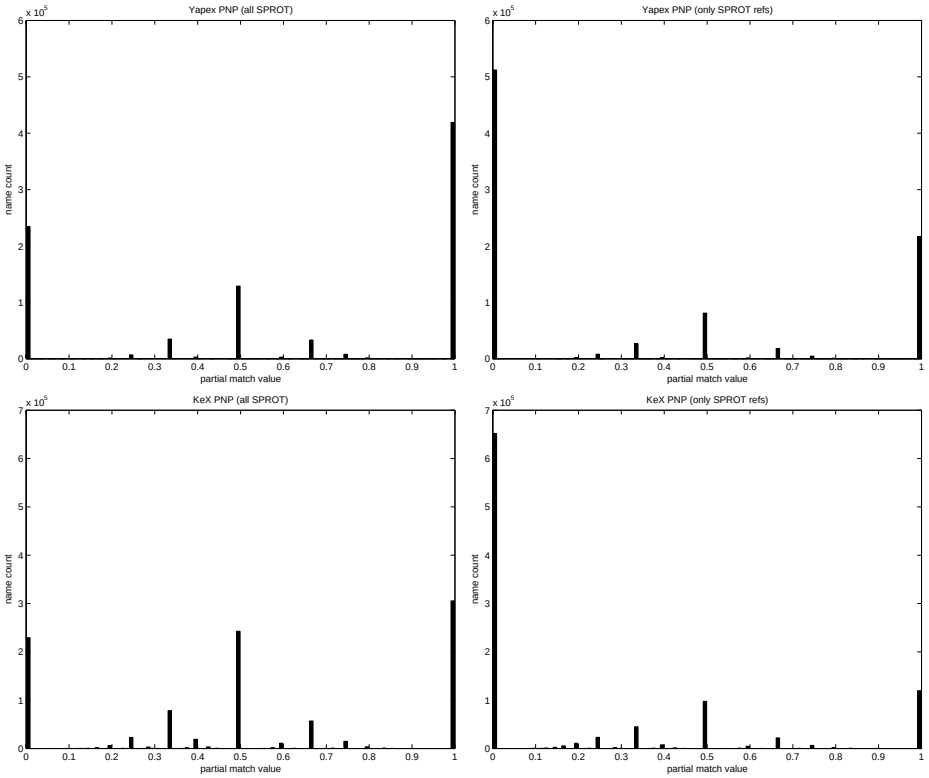


Fig. 1. This figure shows histograms for the PNP comparison: Yapex on top, KeX on the bottom. The figures on the left show results vs. all of SPROT; those on the right, vs. only references SPROT entries.

There *are* some patterns, though. The match values for KeX are biased towards smaller values, indicating that its protein names contains superfluous parts and are thus on average too long, which was also found by [4]. The former is also indicated by the average length of recognized proteins: for Yapex, this is 1.59 ± 0.95 which for KeX it is much higher at 2.17 ± 1.56. For Yapex, the values are quite symmetric in their arrangement between match values 0 and 1.

4 Related Research

[4] gives an excellent overview on the challenges and issues of protein name recognition; and also compares Yapex to KeX on manually annotated data. We found their discussion of comparison methodologies for protein names quite enlightening.

[5] introduces a system to extract protein names, which has been extended towards KeX. They report excellent results, which have been called in question by [3, 4], and incidentially also by our work here.

5 Conclusion

We have applied an automatic comparison methodology on two protein name recognizers. We were able to validate some conclusions from an earlier manual comparison of the same two recognizers by [4], concerning the comparable performance of KeX and Yapex when compared via *Sloppy*, and the overlong matches of KeX.

We look forward to compare all current approaches within the same methodology, and also investigate in more detail the strengths and weaknesses of different approaches. Ultimately, we hope to create large, unbiased training sets for protein name recognition tasks with this approach, complementary to manually generated training sets.

Acknowledgements

This research was supported by the European Union under project no. QLRI-2002-02770 (BioMinT). The Austrian Research Institute is supported by the Austrian Federal Ministry of Education, Science and Culture. We want to thank Kristofer Franzen for supplying the results for Yapex, so we did not have to push 83,044 MEDLINE publications through their web form; and to the team of [5] for making their system publicly available, although it does have some minor bugs.

References

1. Blaschke, C., Andrade, M.A., Ouzounis, C., Valencia, A. (1999) Automatic Extraction of biological information from scientific text: protein-protein interactions, in 7th International Conference on Intelligent Systems in Molecular Biology, (ISMB'99), Heidelberg, pp.60-67.
2. Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, Nucleic Acids Research, 31(1):365-370.
3. deBrujin, B., Martin, J. (2000) Protein name tagging, presented as a poster at the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00)
4. Franzen K., Eriksson G., Olsson F., Asker L., Liden P., Coester J. (2002) Protein names and how to find them, International Journal of Medical Informatics, Vol. 67/1-3, Special Issue on NLP in Biomedical Applications, pp.49-61.
5. Fukuda, K., Tsunoda, T., Tamura, A., Takagi, T. (1998) Toward information extraction: identifying protein names from biological papers, in Pacific Symposium on Biocomputing, pp.705-716.
6. Hanisch, D., Fluck, J., Mevissen, H.-T., Zimmer, R. (2003) Playing Biology's Name Game: Identifying Protein Names in Scientific Text, Proceedings of the Pacific Symposium on Bioinformatics (PSB), 2003.
7. O'Donovan,C., Martin,M.J., Gattiker,A., Gasteiger,E., Bairoch,A. and Apweiler,R. (2002) High-quality protein knowledge resource: SWISS-PROT and TrEMBL. Briefings in Bioinformatics, *3*, 275–284.
8. Ono, T., Hishigake, H., Tanigami, A., Takagi, T. (2001) Automated extraction of information on protein-protein interactions from the biological literature, Bioinformatics, 17(2), 155–161.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics*, held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

Evaluating Context Features for Medical Relation Mining

Špela Vintar[♦], Ljupčo Todorovski[♠], Daniel Sonntag[♣], Paul Buitelaar[♥]

[♦]Faculty of Arts, University of Ljubljana, *spela.vintar@guest.arnes.si*
[♠]Department of Intelligent Systems, Jožef Stefan Institute, *ljupco.todorovski@ijs.si*
[♣]DaimlerChrysler AG, Research & Technology, Ulm, *daniel.sonntag@daimlerchrysler.com*
[♥]DFKI GmbH – German Research Center for Artificial Intelligence, *paulb@dfki.de*

Abstract

The paper describes a set of experiments aimed at identifying and evaluating context features and machine learning methods to identify medical semantic relations in texts. We use manually constructed lists of pairs of MeSH-classes that represent specific relations, and a linguistically and semantically annotated corpus of medical abstracts to explore the contextual features of relations. Using hierarchical clustering we compare and evaluate linguistic aspects of relation context and different data representations. Through feature selection on a small data set we also show that relations are characterized by typical context words, and by isolating these we can construct a more robust language model representing the target relation. Finally, we present graph visualization as an alternative and promising way of data representation facilitating feature selection.

1. Introduction

Finding previously unknown information in large text collections is undoubtedly the greatest challenge of Text Mining, and biomedicine remains one of its most interesting domains of application. This is primarily due to the potentially very broad impact of biomedical findings, but also to the extensiveness of electronic knowledge sources (e.g. UMLS and Medline), "waiting" to be exploited in an innovative way integrating natural language processing and machine learning techniques.

Using linguistic analysis and medical thesauri, we introduce multiple levels of semantic annotation which help us narrow our search to selected medical concepts or semantic types. Despite all this explicitly or implicitly available knowledge, the identification of semantic relations, such as *substance A* ***treats*** *disease B*, remains a non-trivial task. Of course, the Semantic Network of the Unified Medical Language System (UMLS) already defines 54 domain-specific relations between the 134 available semantic types, which enables us to identify instances of UMLS relations in texts. However, applying the Semantic Network relations to medical abstracts shows that those relations are often too generic, ambiguous or incomplete. Also, for many knowledge modelling or information extraction tasks 54 different relations are too much, as the boundaries between, say, *associated_with* and *interacts_with* tend to be blurred. We therefore seek for better ways of identifying selected domain-specific relations in medical texts, and since we believe that meaningful relations between concepts are verbalized in some way or another, the aim is to identify the context features that most reliably point to a certain semantic relation and learn the most effective way of representing them. By context features we mean in particular the linguistic environment of a pair of concepts, which we explore at different levels, including pure tokens, selected part-of-speech classes and semantic classes.

The first indicator of a possible semantic relation is when two concepts co-occur more frequently than would be expected by chance. We use a corpus of medical abstracts obtained from Springer (a subset of Medline) to extract pairs of co-occurring concepts, which we generalize according to MeSH-tree membership at level 0. In order to explore context features in a controlled environment we use manually compiled lists of pairs of MeSH-tree leaves, which according to the medical expert very probably represent a specific semantic relation. Thus, for the relation *treats* the medical expert provided a list of over 100 pairs, such as D13|C23. Similar lists were compiled for 3 other relations, *location_of, causes* and *analyzes.*

Using these data sets and the semantically annotated corpus, we seek to answer the following two questions: Firstly, which context features most reliably characterize a relation or help us distinguish between possible relations, and secondly, which data representation and mining algorithm works best in grouping MeSH-pairs according to the relation they represent.

2. Related work

We are aware of several approaches to mining semantic relations from text for various applications, e.g. ontology construction, information retrieval and knowledge discovery,

much of the latter in the biomedical domain. Approaches to ontology construction are primarily focused on discovering taxonomic or non-taxonomic relations between concepts, for example by learning association rules [12] or by concept clustering combined with grammatical relations [2]. In contrast to a typical ontology building scenario, we exploit an already existing ontology (UMLS) to identify our concepts and semantic classes and then focus on specific (i.e. labelled) medical relations for potential ontology enrichment.

A more supervised line of research aims to find relations via lexico-syntactic patterns, e.g. {NP}, *especially* {NP}, which would match pairs of hypernyms as in *European countries, especially France* [10], [9], [6], [1].

For the purposes of knowledge discovery in medicine even unlabelled associations or statistical correlations may prove useful for hypothesis generation, as Swanson's experiments show [15]. Later work by Weeber et al. [16] proposes a more sophisticated model of automatic hypothesis generation from a medical corpus, which already integrates some linguistic processing and semantic annotation. Finding specific medical relations, such as *X causes Y*, was initially attempted through tables of pattern-matching rules based on co-occurrences of MeSH-classes [4]. Rosario et al. [13] use MeSH in a similar way to determine semantic relations within noun compounds. Our approach also uses pairs of co-occurring MeSH-classes, however instead of providing patterns or rules we try to learn the context that determines a particular relation.

3. Linguistic and semantic processing

To obtain concept co-occurrence data and contextual features we use two corpora of medical abstracts: the MuchMore Springer bilingual corpus[1] of ca. 9,000 abstracts in German and English and a subpart of the Ohsumed corpus of 22,000 abstracts in English. Both corpora were linguistically and semantically processed using tools developed within the MuchMore[2] projecton cross-lingual information retrieval in the medical domain. Linguistic processing plays an important role in the accuracy of semantic annotation, where we identify medical terms and map them to UMLS concepts. Linguistic processing included tokenization, part-of-speech tagging and lemmatization; for the latter the morphological lexicon was extended to include medical terminology.

The main semantic resource for the medical domain is UMLS (Unified Medical Language System)[3], a multilingual database of medical terms, concepts, definitions and semantic relations. UMLS consists of 3 major parts: Metathesaurus, Semantic Network and Specialist Lexicon. The Metathesaurus is essentially a large termbank listing medical terms and expressions and assigning a language independent code to each term (CUI – Concept Unique Identifier). Since UMLS is being developed as an integrated system unifying various medical thesauri and sources, it also includes the mappings of CUIs to some of these more specific thesauri. Thus, one of such core sources is MeSH (Medical Subject Headings), a thesaurus organizing medical knowledge into 15 top tree nodes, each of which is marked with a letter and subdivided into branches. For example, A stands for Anatomy, B for Organisms, C for Diseases etc.

In our semantic annotation we identify medical terms and label them with codes (CUIs) from the UMLS Metathesaurus. These are mapped further to semantic types (TUI – Type Unique Identifier) as well as to MeSH codes corresponding to the nodes in the MeSH tree hierarchy. Although the 134 semantic types defined by the UMLS are also hierarchically ordered, we opted for using MeSH descriptors instead, because these transparently show the position of a certain concept within the MeSH tree structure. They also allow us to choose the desired level of abstraction simply by climbing to higher-level tree nodes. Thus, if a text contains the medical term *anorexia nervosa,* it will be assigned the concept code C0003125 and the MeSH descriptor F03.375.100, which can then be abstracted to F03 – Mental Disorders.

4. Text Mining methods

4.1 Hierarchical clustering

Clustering is an unsupervised learning method [15]. Given data about a set of instances, a clustering algorithm creates groups of objects following two criteria. Firstly, instances are close (or similar) to the other instances from the same group (internal cohesion) and secondly, they are distant (or dissimilar) from instances in the other groups (external isolation).

A particular class of clustering methods, studied and widely used in statistical data analysis are hierarchical clustering methods [15]. The hierarchical clustering algorithm starts with assigning each instance to its own cluster, and iteratively joins together the two closest (most similar) clusters. The distances between instances are provided as input to the clustering algorithm.

[1] http://muchmore.dfki.de/resources1.htm
[2] http://muchmore.dfki.de
[3] http://www.nlm.nih.gov/research/umls/

The iteration continues until all instances are clustered into a single cluster. The output of the hierarchical clustering algorithm is a hierarchical tree of clusters or dendrogram that illustrates the order in which instances are joined together in clusters. In the final step of the hierarchical clustering algorithm, clusters are obtained by cutting the dedrogram into sub-trees: elements in each sub-tree form a cluster. Cutting the same dendrogram at different heights produces different number of clusters. The optimal "cut point" that produces clusters with maximal internal cohesiveness and minimal external isolation from a given dendrogram is where the difference between heights of two successive nodes in the tree is maximal.

4.2 Data representation and distance measures

In our experiments, instances are MeSH-pairs. Each MeSH-pair (e.g. A1|C23) is described by 300 most frequent contextual features, which were observed to be nouns and verbs. Two different data representation were used to represent the feature vectors. The first data representation is relative frequency, i.e., frequency of context words relative to the frequency of the observed instance in the corpora. The second is simple binary true/false representation where only presence/absence of words in the context of the observed MeSH-pair is considered.

Another parameter that may influence the success of clustering is the measure of distance between instances. Apart from the standard Euclidean and Manhattan distances, which can be used with both the relative frequency and binary data representation, we also tested a distance measure based on the Jaccard coefficient (1) for measuring similarity between binary vectors interpreted as sets of words (X and Y):

$$JD(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}. \quad (1)$$

4.3 Evaluation of clustering

We used a metric for evaluation of clustering that is based on comparison of the set of clusters, obtained in the experiments (*candidate clustering*), with the *reference clustering*, provided by human expert. Namely, for each instance in the dataset human expert provides the relation represented by the pair (e.g., *treats* or *location_of*). Then, let $R = \{R_1, R_2\}$ [4] be the reference clustering and $C = \{C_1, C_2, \ldots, C_n\}$ be the candidate clustering. The measure of quality of C with respect to R can be defined as:

$$Q_R(C) = \frac{\max_{i,j,i \neq j}(O_{1i} + O_{2j})}{N},$$

where $O_{ki} = |R_k \cap C_i|$ measures the overlap between clusters and N is the number of instances in the data set. The quality measure assesses the classification accuracy of the classifier that assigns semantic relations to MeSH-pair instances based on the obtained clustering C. The range of the quality measure is *[0,1]*. The quality of 1 is obtained when the candidate clustering C is identical to the reference R.

5. Identifying context features for relation mining

5.1 Evaluation of context features

Starting from the hypothesis that a frequently co-occurring pair of MeSH classes indicates the existence of a semantic relation and that this relation is somehow expressed through language, we wish to determine the context features that help us identify and label the relation. Many linguistically motivated approaches to relations have focussed on verbs as vehicles of relationships between syntactic elements, however in medical texts we observe that nominalizations are frequently used instead of verbal forms, as in *The treatment of [disease X] through [substance Y] proved successful.* We therefore tested the following possible features occurring within the same sentence:

- all tokens *(tokens)*,
- all verbs *(verbs)*,
- all nouns *(nouns)*,
- all other concepts *(cuis)*.

Context	tl_old	tl_new
tokens	0.5454	0.5642
nouns	**0.6591**	0.6703
verbs	0.6363	**0.6872**
cuis	0.5681	0.5195

Table 1: Selecting context

For each of the above, frequency data was collected from the corpus for our sets of manually labelled pairs and used for hierarchical clustering. Table 1 shows a comparison of clustering accuracy for all of these settings on two different data sets (*tl_old* and *tl_new*); the data representation selected for the above comparison was binary.

It is clear from the above that among these settings, nouns and verbs perform best, we therefore used nouns and verbs as context for all further experiments. We also experimented with the

[4] In this paper we deal with the task of distinguishing between two relations only, so the reference clustering consists of two clusters only.

number of attributes used to describe each pair, which resulted in an optimal cut-off point of 300 for most frequent verbs and nouns.

5.2 Data representation and distance measure

Table 2 shows the evaluation of data representation and distance measures for three different data sets:

- tl_old: 49 MeSH-pairs representing the relations *treats* and *location_of,*
- tl_new: 287 additional MeSH-pairs representing the relations *treats* and *location_of,*
- ac: 89 MeSH-pairs representing the relations *analyzes* and *causes.*

The distance measures tested were Euclidean (euc), Manhattan (man) and Jaccard (jacc).

Data set	Data representation	Distance measure	Score
tl_old	binary	euc/man	0.5227
		jacc	0.4318
	relative frequency	euc	0.5454
		man	**0.5682**
tl_new	binary	euc/man	0.5225
		jacc	**0.7528**
	relative frequency	euc	0.6910
		man	0.7303
ac	binary	euc/man	**0.7368**
		jacc	0.5131
	relative frequency	euc	0.6973
		man	0.6578

Table 2: Data representation and distance measures

5.3 Learning typical contexts

For some purposes, for example information retrieval, the results as given by Table 2 might be sufficient to distinguish between two or more relations on the basis of context. However, in all above scenarios the clusters are still very fuzzy. Using the optimal split into clusters indeed produces two clusters most of the time, but the quality of the clusters remains between 70 and 80%. Therefore, in order to obtain a clearer view of which features best function as distinctive and whether it was possible to generalize these findings, an experiment involving supervised feature selection was performed. For the data set *tl_old* of 49 manually selected MeSH-pairs representing the relations *treats* and *location_of*, the context words were automatically weighted according to their occurrence with either *treats*-pairs or *location_of*-pairs. Then, only words that were found to occur with one of the relations significantly more often than with the other were kept as context words, others were omitted. Recursively testing this controlled context on the same data set left us with 4% of the initial context words, a list of 290 distinctive context words for the selected relation pair. Table 3 lists the results obtained with 40%, 20% and 4% of the context words respectively.

To test whether this list was distinctive only for the data set it had been produced with or for all data sets representing the same relation pair, the new data set *tl_new* of 190 MeSH-pairs representing the relations *treats* and *location_of* was constructed. The context features were now no longer most frequent verbs and nouns but the list of 290 distinctive words, for which data was obtained from the larger corpus, Ohsumed, and clustered. The last two lines of Table 3 show the results of this controlled-context experiment *(tl_new 0.04)* compared with the uncontrolled-context result given above *(tl_new W/O 1.0)*. The improvement is significant, which shows that contextual features learned on a small data set can be generalized to a larger data set of the selected relation pair.

Data set	Filtering threshold	Score
tl_old	0.40	0.6888
	0.20	0.6888
	0.04	0.8889
tl_new	0.04	**0.8212**
	W/O (1.0)	0.5225

Table 3: Selecting context words

6. Graphical representation

The graphical representation is used for visualising the data and for inducing new data instance vectors that serve as alternative input for the ML algorithms. We provide a powerful data engineering facility while projecting the data onto a two-dimensional grid, since the 2D graph format visualises potentially interesting structures. The spatial co-ordinates form new input for learning schemes apart from those obtained using regular attribute selection and discretization methods in higher-dimensional attribute vector spaces.

We use an information-preserving mapping from vector data attributes to graphical properties where all attribute values are reflected in the corresponding graph [3], [5]. Most of these properties represent input to a graph layout optimisation algorithm: We use an automatic graph layout with layout constraints and an objective function based on aesthetic criteria that serves well for displaying semantic proximity if the graph structure is well designed.

Most of the global graph properties represent input to the graph layout optimisation algorithm. Every

data instance is presented as an undirected graph with all data attributes as vertices. The target concept is a special vertex, with special shape, colour and size. The shape and the colour are perceptual attributes for better visual data inspection, while the size of the node reflects the special status of the target node as a special attribute.

Each target node is connected with an edge to every attribute. The ratio of the target node size and attribute node accounts for the fact that one target node is always connected to several attribute nodes. The attribute value (e.g. relative frequency) is projected onto a discrete numerical value representing the preferred edge length that is also input to the layout algorithm. The algorithm accepts integer values between 80 and 400. The mapping function was defined in a way to map the attribute values inversely proportional on this interval. This means that a high attribute value sets a low preferred edge length on the edge between the target node and the attribute node reflecting high term/word weight as (semantic/spatial) proximity between attribute and target.

When laying out a graph, nodes are considered to be physical objects with mutually repulsive forces, like protons or electrons. The connections between nodes also follow the physical analogy and are considered to be metal springs attached to the pair of nodes. These springs produce repulsive or attractive forces between their endpoints if they are too short or too long. The layouter simulates these physical forces and rearranges the positions of the nodes in such a way that the sum of the forces emitted by the nodes and the edges reaches a (local) minimum [7], [8].

6.1 Clustering coordinates

Using graph objects and their visualization as an alternative way of representing our data and evoking an automatic layout algorithm (and tuning the layout parameters) now produces two-dimensional vectors of co-ordinates, which can be clustered with the same algorithm as before and the Euclidean distance. Using the co-ordinates as new input vectors is a special kind of dimensionality reduction method inspired by visualisation techniques for analysing meaning [17]. Table 4 shows the clustering results for all 3 data sets.

In general, the performance of this method is roughly comparable to the statistical method, slightly better with some data sets and slightly worse with others. The visualisation shows that clusters can better be separated visually than automatic clustering reflects.

It is clear however that any semi-automated or interactive approach to knowledge discovery would benefit from a graphical representation, both for data/parameter selection and evaluation. The graphical representation may act as additional view of the text data to reveal new data characteristics that can be visually explored by the domain expert and quantified by graph properties. In particular, to find relationships between term distribution models and graphical representations may help to characterize how informative a word is [11].

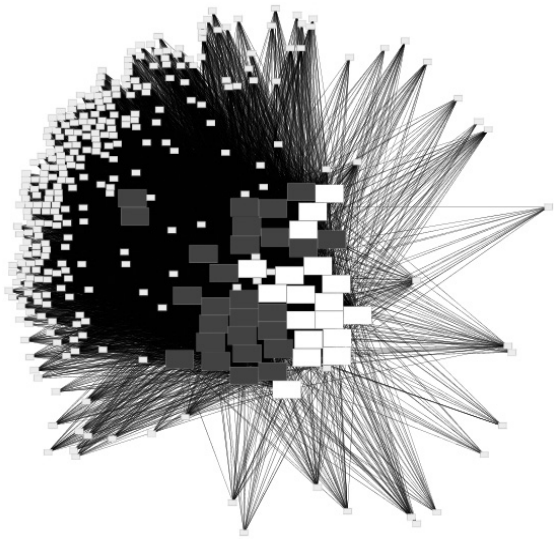
Data set	Score
ac_rf_ne_404	(4) 0.4473
ac_rf_ne_smart	(3) 0.5131
tl_new_smart40	(4) 0.4213
tl_old_smart400	(2) 0.5454
tl_old_smart_rf	(4) 0.5909

Table 4: Visualization and clustering coordinates

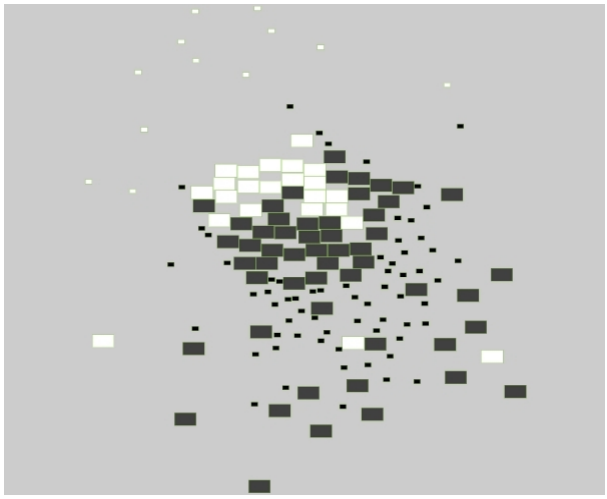
6.2 Feature selection through visualization

Knowing that semantic relations can be identified by their context, graphical representation can also be used as an alternative way of selecting distinctive features, e.g. typical words. Pictures 1 (ac_rf_ne_smart) and 2 (tl_old_smart_rf) below show the distribution of typical features on the two-dimensional grid and their correspondence to the formation of visual clusters. Large black and white boxes represent instances of the two relations, *analyzes* and *causes*, and the small boxes represent context words. It can be seen very clearly how typical context words "pull" instances into the white or black cluster.

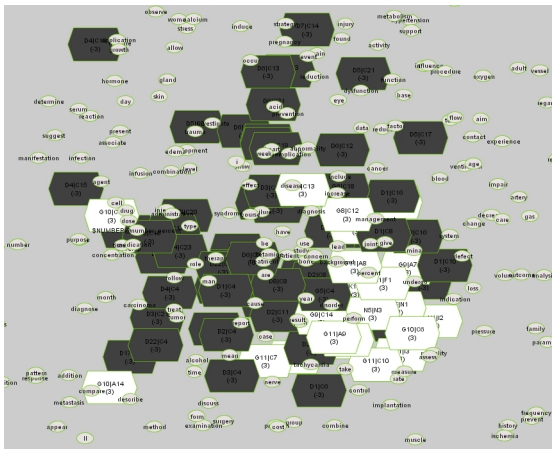
Picture 3 (ac_rf_ne_404) shows the term cloud for *analyses-causes*. In this constellation we allow for overlapping nodes, which leaves less constraints for the resulting layout. Although the score for this dataset was low, one can see that the different relations are apparent. Interestingly, typical stop words (be, have, are, patient) are positioned in the barycentre of the graph.



Picture 1: Distinctive features for *treats - location_of*



Picture 2: Distinctive features for *analyzes - causes*



Picture 3: Term Cloud for *analyses-causes*

7. Conclusions

Starting from the hypothesis that semantic relations are realized in texts through identifyable context features, the goal of the described experiments was to design a methodology to model relations and determine the parameters that distinguish relations. After evaluating different data representations we propose a method for feature selection on a relatively small data set of 49 manually selected relation instances, which was found to perform well also on a larger set of relation instances. In further work we will explore the interaction between general statistical methods, vector-based representations and graph representations. The result could be used as decision support for text-mining algorithm selection or be combined with the outcome of a text-mining algorithm on the original data.

Although all our test sets were limited to two relations, the approach can be easily generalized to an arbitrary number of domain-specific relations. The evaluation of the approach on distinguishing between more than two relations is another direction for further work.

By learning context models of medical semantic relations, new unlabelled instances can be classified and thus identified in texts. A particularly important application of relation extraction is in document retrieval, where a query may be pruned or expanded according to the target relation. On the other hand, collecting new relation evidence from large text collections can also be used for the purposes of enriching the UMLS.

In order to test the usability of the methods we propose, context models should be constructed for all target relations and evaluated in a classification task. Finally, we also envisage transferring this approach to a proper knowledge discovery task by expanding the context of a relation to larger text sections or entire documents or search for document parts where the vocabulary and thus the context of a relation instance shifts from one instance to another.

Acknowledgements

This research has in part been supported by EC/NSF grant *IST-1999-11438* for the MUCHMORE project.

References

1. Agichtein, E.; Gravano, L. (1999) Snowball: Extracting relations from large plain-text collections. Columbia University Computer

Science Department Technical Report CUCS-033099.
2. Bisson, G.; Nédellec, C.; Cañamero, D. (2000) Designing clustering methods for ontology building: The Mo'K workbench. In: Ontology Learning ECAI 2000 Workshop, Berlin.
3. Bradley, J.; Rockwell, G. (1994) What Scientific Visualisation Teaches us about Text Analysis, ALLC/ACH '94, Paris
4. Cimino, J.J.; Barnett, G. O. (1993) Automatic Knowledge Acquisition from MEDLINE. Methods of Information in Medicine, 32(2): 120-130.
5. Engelhardt, Y. (1997) Toward a Unified Visual Representation of Documents and Concepts, CODATA Euro-American Workshop on Visualisation of Information and Data.
6. Finkelstein-Landau, M.; Morin, E. (1999) Extracting semantic relationships between terms: Supervised vs. unsupervised methods. In: International Workshop on Ontological Engineering in the Global Information Infrastructure, 71-80.
7. Frick, A.; Ludwig, A.; Mehldau, H. (1994) A Fast Adaptive Layout Algorithm for Undirected Graphs (Extended Abstract and System Demonstration) Proc. DIMACS Int. Work. Graph Drawing, GD
8. Fruchterman, T.M.J.; Reingold, E.M. (1991) Graph drawing by force-directed placement. Software-Practice and Experience, 21.
9. Gildea, D.; Jurafsky, D. (2000) Automatic Labeling of Semantic Roles, ACL 2000 Hong Kong, pp. 512-520.
10. Hearst, M.A. (1998) Automated discovery of Wordnet relations. In: Fellbaum, Ch., ed., WordNet: An Electronic Lexical Database, pp. 131-151, MIT Press.
11. Lee, L.; Pereira, F. (1999), Distributional Similarity Models: Clustering vs. Nearest Neighbors, 37th Annual Meeting of the Association for Computational Linguistics, pages 33-40.
12. Maedche, A.; Staab, S. (2000) Discovering Conceptual Relations from Text. In: W. Horn (ed.) ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, IOS Press, Amsterdam.
13. Rosario, B.; Hearst, M.A.; Fillmore, Ch. (2002) The descent of hierarchy, and selection in relational semantics. In: Proceedings of ACL '02, Philadelphia, PA.
14. Hartigan, J.A. (1975) Clustering Algorithms. Wiley, New York.
15. Swanson D.R.; Smalheiser N.R. (1997) An interactive system for finding complementary literatures: A stimulus to scientific discovery. Artificial Intelligence 91:183–203. `http://kiwi.uchicago.edu`.
16. Weeber, M.; H. Klein, A. R. Aronson, J. G.; Mork, L. Jong-van den Berg; R. Vos (2000) Text-Based Discovery in Biomedicine: The Architecture of the DAD-system. In: The American Medical Informatics Association 2000 Symposium.
17. Widdows, D.; Cederberg, S.; Dorow, B. (2002) Visualisation Techniques for Analysing Meaning. Fifth International Conference on Text, Speech and Dialogue, Brno, Czech Republic, September 2002, pages 107-115.

Emerging Pattern Based Projected Clustering for Gene Expression Data

Larry T. H. Yu
Department of Computing
Hong Kong Polytechnic University
Hunghom, Kowloon, Hong Kong
(852) 2766 4901
csthyu@comp.polyu.edu.hk

Fu-lai Chung*
Department of Computing
Hong Kong Polytechnic University
Hunghom, Kowloon, Hong Kong
(852) 2766 7289
cskchung@comp.polyu.edu.hk

Stephen C.F. Chan
Department of Computing
Hong Kong Polytechnic University
Hunghom, Kowloon, Hong Kong
(852) 2766 7259
csschan@comp.polyu.edu.hk

ABSTRACT
In this paper, we propose a clustering method to group the high dimensional gene expression into clusters that are easy to understand by biologists. The importance of study gene expression data is that it gives us opportunities to understand those biological processes within organisms at molecular level. By increasing our knowledge through information comes from the data, many genetic related diseases would be cured. However, gene expression data is high dimensional in nature and it is not easy to manipulate and understand. In view of their successes in solving different data mining problems, the emerging patterns (EPs) and projected clustering techniques are integrated for effective clustering of gene expression data. The key concept of the resulted EP-based projected clustering (EPPC) algorithm is to introduce the readability and strong discriminatory power of EPs in the dimension projection process of the projected clustering so that the readability of the projected clusters can be improved. Encouraging experimental results were obtained.

Keywords
Data mining, emerging patterns, projected clustering, bioinformatics, gene expression data.

1. INTRODUCTION
Knowledge discovery from the gene expression data by clustering techniques is hot, especially in the cancers related studies. However, the high dimensionality of the gene expression data and the required readability of clustering results give challenges to most of the existing data mining methods. We propose a new clustering method called "Emerging Pattern based Projected Clustering" (EPPC) to form easy understandable clusters for gene expression data and cancer correlation studies.

Bioinformatics is one of the most popular research areas since biological scientists started to open the molecular black box in living cells with the advanced biotechnologies [12]. Biological scientists are now not limited to study the large components of a cell by electronic microscopes. They also started to discover the secret of life at molecular level. One of the most significant projects, U. S. Human Genome Project [13], started in 1990. In this year, scientists have determined the sequences of the 3 billion chemical base pairs that make up human DNA already. Those determined sequences are used in identify genes in human DNA and various kind of researches. Although we have learned many new insights from the sequence, there is still tons of knowledge waiting for us to discover. The advancements in biotechnologies generate a lot of data and the use of computer becomes one of the essential components in biological data analysis.

Bioinformatics is an interdisciplinary subject that focus on the use of computational techniques to assists the understanding and organization of the information associated with biological macromolecules. Not limited to the raw DNA sequences, there are various types of data, such as protein sequences data, macromolecular structure data, genomes data, gene expression data and so on. They provide unpredictable opportunities to researchers and most of them are available freely on the internet. In recent years, data mining techniques are frequently used in the correlation studies of gene expression data and cancers.

Gene expression patterns are different between different tissues. In normal tissues, every cell of our body contains a full set of chromosomes and identical genes. However, only a fraction of these gene are turned on, called expressed [11], during the protein synthesis. Those expressed genes determine the unique properties of different types of cells and those mRNA [11] that the cell can produce. By study the types and amounts of mRNA produced by a cell, scientists can determine those genes that are expressed and obtained the gene expression context under different environments. By using the microarray technology, we can retrieve large amounts of gene expression data from a single experiment and many genetic related studies become possible. For example, scientists are now using the context of gene expression levels to discover the causes of cancers since there are differences in gene expression between normal and cancer tissues.

In *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics*, held in conjunction with ECML/PKDD. Dubrovnik, Croatia, 2003.

* Corresponding author

The high dimensional nature of gene expression data is inherited by the physical properties of the microarray experiments. In a single experiment, large number of genes in a tissue is examined under different conditions. For example, there exist microarray experiments that can examine about 40,000 genes from 10 samples under 20 different conditions in one single experiment [4]. Therefore, the gene expression data always consists of large number of numerical attributes, i.e. gene expression values, but limited records

On the other hand, gene expression data always contain class information. Microarray experiments are still expensive in terms of both time and cost, they are only being conducted to investigate some of the biological significant properties. The data itself always contains predefined class labels of important biological meanings, e.g., normal and cancerous tissues in cancer related data [1].

In summary, we are challenging by the flood of gene expression data as well as their complexity. High dimensionality is the major challenge to those existing mining algorithms in the knowledge discovery process. The sparseness of data in the high-dimensional space, referred as dimensionality curse, causes the meaningfulness of proximity or clustering being questioned [3]. Moreover, most of the available datasets contain very limited number of records compared with the number of available attributes. For example, the "Subtype of childhood leukemia" dataset [10] from St. Jude Children Research Hospital contains more than 12,000 genes but only 327 samples. It is very difficult for us to obtain a good approximation of the real world from the data. Last, but not the least, the readability and understandability are important to these biological related studies.

Pattern association and clustering are two data mining techniques that are now frequently applied in the field of cancer and gene expression correlation studies, but they cannot satisfy our needs. They are expected to discover the cancer causing gene expression patterns for different diagnosis purposes, e.g., identifying the development of cancers in earlier stages and proposing useful treatment plans [8]. However, the usefulness of the gene expression data analysis obtained by traditional data mining techniques is still questionable due to the aforementioned dimensionality curse problem. Furthermore, clusters formed by conventional measures of tightness of data points often lack of practical meaningful support and they are not easy to understand and become applicable in the knowledge discovery process in biology domain. In short, techniques for gene expression data analysis needed to tackle the high dimensionality of data and provide easy understandable analysis results.

Projected clustering method is designed to tackle the problem of dimensionality curse. It overcomes the sparsity of data points by projecting them into lower dimensional subspace. The strength of projected clustering is that it eliminates the limitations of feature selections that may not always be possible to prune off too many features without any information loss [3]. However, the resulting clusters are not easy to understand, particularly to biologists.

Emerging patterns (EPs) [5] are specific patterns that their occurrences have significant differences between different partitions of a dataset. For example, EPs may only exist in cancerous tissues but they are absent from normal tissues. Unlike the frequent patterns in ordinary association analysis, EPs have high discriminatory power in nature and they have been proved to be useful in classification problem [6]. EPs are also easy to understand because they are just the collections of attributes in dataset and this property is especially important for bioinformatics applications. However, the volume of EPs generated is very large for high dimensional gene expression data [7]. In order to maintain the efficiency in using EPs, we have to use top EPs instead of full set of EPs in applications. Although the accuracy in the classification studied in literatures is still high [6], [9], the robustness of EPs for the new data is still questionable. Using large number of top EPs in application can absolutely improve the robustness but the efficiency will become lower and it should not be an ultimate solution.

In this paper, our target is to group the gene expression data points into clusters that are easy to understand. We make use of the domain knowledge and patterns generated from different classes in the gene expression data to form clusters. Our clustering strategy is closely related to projected clustering [3], a technique used to cluster high dimensional data. In the past, emerging patterns and projected clustering techniques were used independently in solving different types of problems since they are strong in different domains. In this paper, we propose to integrate them for effective clustering of gene expression data. The key concept of our approach is to introduce the readability and strong discriminatory power of EPs in the dimension projection process of the projected clustering so that the readability of the projected clusters can be improved. The rest of the paper is organized as follows. The use of EPs is briefly described in section 2. In Section 3, we present our EP-based projected clustering approach. The dataset used and the experimental results are reported in section 4. The final section concludes the paper.

2. USE OF EMERGING PATTERNS

Emerging patterns (EPs) were first introduced by Dong and Li [5]. They are defined as itemsets whose supports increase significantly, larger than the threshold value called the growth rate (ρ), from one dataset (D1) to another (D2). There are several types of EPs, such as Jumping EPs (JEPs), plateau EPs and so on. All of them have different properties and they are applied to different problems. They are easy to understand because they are only collections of data attributes. For example, a cancerous EP [7]:

$$\{\text{gene(K03001)} \geq 89.20\} \text{ and } \{\text{gene(R76254)} \geq 127.16\} \text{ and } \{\text{gene(D31767)} \geq 63.03\}$$

is a pattern that only occurs in cancerous tissues but not in normal tissues.

Another advantage of EPs is their discrimination power in classification. EPs capture the significant differences of attribute's values that exist in different partitions of dataset. So, EPs naturally identify those collections of attributes of data points that are close within the same cluster but far away to others. Yet another advantage about EPs is the border based mining algorithms [5] which can mine a complete set of EPs efficiently even the support value of EPs is low. It is very important for the bioinformatics applications because patterns with low occurrence are still important. For example, cancer classification problem is targeted to find out cancer tissues whose occurrence is relatively

low when compared those normal ones. Other patterns, such as traditional frequent patterns from association analysis, are potentially having problems in their extraction process.

In terms of the discrimination power, the EPs with highest growth rates are most preferred. Therefore, the JEPs, whose growth rate is infinity, are being employed in our proposed method. However, the number of JEPs is still numerous [7]. For the sake of simplicity and efficiency, we use the top 20 cancerous tissues JEPs and the top 20 normal tissues JEPs mined [9] from 35 top-ranked genes by entropy method in this study.

3. EP-BASED PROJECTED CLUSTERING

Before proceeding to describe our EP-based projected clustering (EPPC) algorithm, we introduce some notations and definitions. Let N be the total number of data points and n_i be the number of data points in cluster C_i. Assume that the dimensionality of full data space D is equal to d and the dimensionality of projected space D_i of cluster C_i is equal to d_i, where $d_i \leq d$. Let $X_i = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ be the set of data points in cluster C_i, $\vec{p}_j = \{x_{j1}, x_{j2}, \ldots, x_{jd_i}\}$ be the projected point and $\vec{s}_i$ be the centroid, i.e. $\vec{s}_i = \sum_{j=1}^{n_i} \vec{p}_j / n_i$. Finally, the (projected) distance of projected point $\vec{p}_j$ to the center of cluster C_i is written as $Pdist(\vec{p}_j, \vec{s}_i, D_i)$.

As mentioned in Section 2, the EPs have high discrimination power in nature. Set of EPs can be considered as collections of features that are highly discriminated between different classes and highly correlated within the same class. Therefore, we propose to employ EPs as guidance in dimension projection for the projected clustering instead of using variants of singular value decomposition (SVD) techniques in order to improve the interpretability of clustering results and consequently the readability for the biologists. In our experiments, the top JEPs [9] having the highest discrimination power were employed. All of them have infinity growth rate and highest occurrence in one of the partitions of the datasets.

3.1 EP-based Projected Clustering Algorithm

Finding projected clusters can be established as a two-fold problem [2] [3]. First of all, we have to locate the cluster's center and then to find out the projected dimensions for the corresponding clusters. The proposed EP-based projected clustering algorithm includes three phases similar to [3], namely, initialization, iteration and refinement phase. In general, the initialization phase is to pick the initial cluster seeds for the iteration phase. In the iteration phase, data points are assigned to different clusters and the projected dimensions of those newly formed clusters are being evaluated. The iteration phase continues until the number of user specified clusters are obtained. Once the set of good cluster seeds is obtained, the refinement phase will start and all the data points will be reassigned to those cluster seeds obtained by the iteration phase to form the final clusters. These three phases of the proposed EPPC algorithm are detailed in Figure 1.

```
Algorithm EPPC (k0, k, E){
  Initialization phase
  Pick k0 > k initial cluster seeds randomly from
  the dataset;
  Set no. of current cluster to no. of initial cluster;
  for each cluster {
    Set the cluster dimension to full dimensionality
  }
  Iterative phase
  While no. of current cluster > user requirement {
    Assign the data points to the nearest cluster seeds;
    Determine the cluster dimensions associated to each
    cluster;
    Merge the closest clusters and obtain the new seed for
    the newly merged cluster;
    Update the no. of current cluster;
  }
  Refinement phase
  Reassign the data points to the set of good seeds
  obtained from iteration phase;
  Determine the cluster dimensions associated to
  each cluster;
  Return the projected clusters with cluster seeds,
  corresponding dimensions and data points;
}
```

Figure 1. EPPC algorithm.

3.1.1 Initialization phase

In this phase, the number of final clusters is assigned by the users. We randomly pick k_0 initial cluster seeds from the dataset, where k_0 should be several times larger than k, and the projected dimensions of all initial seeds are initialized to the full dimensions of the dataset initially.

3.1.2 Iteration phase

The goal of the iteration phase is to improve the quality of the cluster seeds iteratively in order to find the best clusters. There are three operations in this phase.

3.1.2.1 Assignment operation

There should be k_c cluster seeds in the current iteration. In this operation, the data points in the dataset are assigned to their closest seed. We used the distance metric, such as city block distance or Euclidean distance, to measure the distance between the data points and cluster seeds under those projected dimensions, i.e. the projected distance. After the partitions are formed, the centroids of each partition are evaluated and they are used as the new seeds in the next iteration. This procedure is illustrated in Figure 2.

3.1.2.2 Dimension projection operation

Those partitions formed by the assignment operation consist of a set of data points. In this operation, the projected dimensions of each projected cluster are evaluated by their own data points. For each partition, we examine its data points and find those EPs embedded. The EPs with most frequent occurrence are chosen and they act as the collection of projected dimensions for that particular partition. This procedure is described in Figure 3.

3.1.2.3 Merging operation

The closest pair of clusters is merged together to form a new cluster. They are obtained by evaluating the average distance between the union of data points and new cluster seed of merged clusters. The smaller the average distance, the closer the pair of clusters. The details of this operation can be found in Figure 4.

3.1.3 Refinement phase

The resulting cluster seeds obtained from the iteration phase are then used to form the final clusters by assigning all the data points to them again. The goal of this phase is to ensure that all data points are assigned to the closest cluster seeds after the final cluster seeds are found.

```
Algorithm Data_Point_Assignment {
   for each data point {
      for each cluster {
         Determine the projected distance between
         the data points and current seeds;
      }
      Add the data points to their nearest cluster;
   }
   Remove cluster from the set if it is empty;
   Set the centroids of those projected clusters as the new
   cluster seed;
   Return the cluster seed and data set in projected clusters;
}
```

Figure 2. Data point assignment algorithm.

```
Algorithm Dimension_Projection {
   for each cluster {
      Find the EPs having most frequent occurrence among
      the data points in the cluster;
      Find the corresponding attributes that make up the EPs;
      Set the projected dimensions to those collection of
      attributes;
   }
   Return the dimensions for the projected clusters;
}
```

Figure 3. Dimension projection algorithm.

```
Algorithm Cluster_Merging {
   Find the closest pair of clusters from the set of
   existing clusters;
   Merge the data points of the two clusters;
   Find the projected dimensions of the unified data points;
   Find the new seed of the unified data points;
   Evaluate the radius of the merged clusters;
   Merge the closest pair of clusters such that the
   radius of the merged clusters is minimal;
   Return the set of new cluster seeds;
}
```

Figure 4. Cluster merging algorithm.

4. EXPERIMENTAL RESULTS

The dataset we used consists of 2000 gene expression values of 40 tumor and 22 normal colon tissues samples [1] and it is publicly available at http://microarray.princeton.edu/oncology/affydata/index.html. Since the original dataset consists of huge number of attributes and not all of them are useful in separating samples into different classes, the dataset is first reduced its size by the entropy discretization method. The reduced dataset consists of 35 top-ranked genes in the dataset as mentioned in [9]. In order to facilitate the study of the relationship between different genes and the tissue types, each gene expression value of the reduced dataset is normalized before clustering. The mean of the normalized gene expression values is 0 while the standard deviation is 1.

In this paper, we report the clustering performance of our proposed EPPC algorithm. Different number of initial and final clusters combination are used and three most popular distance metrics, namely, Euclidean, City block and City segmental distance, are employed. All samples in the dataset are used in the experiments and the error of the resulting cluster is calculated as:

In order to minimize the effect from different initializations, each combination of experiment settings was simulated 50 times and the average error rate is listed in Tables 1 – 3.

Table 1. Clustering error based on Euclidean distance

Initial Cluster Num	Final Cluster Num.								
	.	4	8	12	16	20	24	28	32
	8	0.221	0.135						
	16	0.249	0.167	0.152	0.110				
	24	0.243	0.165	0.151	0.135	0.140	0.095		
	32	0.272	0.177	0.136	0.135	0.113	0.122	0.123	0.084
	40	0.257	0.174	0.137	0.116	0.103	0.092	0.102	0.096

Table 2. Clustering error based on City block distance

Initial Cluster Num	Final Cluster Num.								
	.	4	8	12	16	20	24	28	32
	8	0.288	0.129						
	16	0.304	0.230	0.222	0.107				
	24	0.311	0.233	0.213	0.206	0.198	0.092		
	32	0.284	0.240	0.197	0.182	0.177	0.166	0.162	0.080
	40	0.318	0.237	0.199	0.170	0.154	0.145	0.138	0.114

Table 3. Clustering error based on City segmental distance

Initial Cluster Num	Final Cluster Num.								
	.	4	8	12	16	20	24	28	32
	8	0.195	0.129						
	16	0.188	0.157	0.141	0.107				
	24	0.193	0.152	0.136	0.120	0.117	0.093		
	32	0.209	0.167	0.143	0.123	0.114	0.100	0.089	0.076
	40	0.197	0.169	0.147	0.119	0.096	0.085	0.076	0.074

From the tables above, the City segmental distance gives the smallest cluster errors. It was observed that in every final cluster, the number of projected dimensions varies due to different EPs being used in their own projections. Thus, better clustering subspaces were identified for better classification. It can also been

seen that the decrease in clustering error as a result of increasing the number of final clusters from 4 to 8 is most significant. It is because there may have quite a number of sources causing the cancer to develop. In our clustering context, more than two natural clusters exist in those gene expression data and using 8 final clusters gives a much better result than using 4. However, if the number of the final clusters is larger than the number of natural clusters in the data, the further increase in the number of clusters may not be so important and thus the improvement in the clustering error becomes limited. Although we only have two class labels obtained from the microarray experiments, we can guess the number of natural clusters from our clustering results and such kind of information is potentially useful in providing some directions for further biological studies.

In order to demonstrate the effectiveness of the EPPC algorithm, the K-means algorithm implemented by a public domain package called NetLab3.2 was used to cluster the same set of data. Again, the simulation was repeated for 50 times for each individual setting. As shown in Table 4, the performance of the K-means algorithm is not as accurate as ours.

Table 4. Clustering error using K-means algorithm

Avg Error	Cluster Num.			
	8	16	24	32
	0.321	0.270	0.233	0.190

In traditional clustering algorithms, the resulting clusters give us information on those data points that are similar under a particular distance metric. Projected clustering algorithm gives us more, i.e., the similarity of data points is high in a particular set of dimensions. But many of them, such as the resulting cluster dimensions obtained by the principle components [3], are not easy to understand. However, our proposed algorithm can give us information that is easier to understand. In our gene expression data experiments, the projected clusters found by the EPPC algorithm are characterized by a collection of attributes. An example is shown in Table V where cluster 2 consists of 6 cancerous tissues that are similar in gene expression values with respect to two suspected genes M76378 and T47377.

Table 5. Examples of resulting clusters obtained by EPPC

	No. of sample	Tissue Type	Cluster Dimension				
			Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
Cluster 1	7	Normal	H51015	R10066	U32519	T47377	Z50753
Cluster 2	6	Cancer	M76378	T47377			
Cluster 3	10	Cancer	H08393	M76378			

5. CONCLUSIONS

In this paper, we propose and discuss the integration of emerging patterns and projected clustering in high-dimensional data analysis. The experimental results not only demonstrate the possibility of using emerging patterns in dimension projection of projected clustering, but also show that the resulting clusters are more readable to end users. Such a feature is very important to many bioinformatics applications. In our future work, we will examine how to extract more information from emerging patterns in order to improve the accuracy of our EPPC algorithm.

6. REFERENCES

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings Of The National Academy of Sciences of the United States of America 96.10 (1999): 6745-6750.

[2] C.C. Aggarwal, C. Procopiuc, J.L. Wolf, P.S. Yu, and S.P. Jong. Fast algorithms for projected clustering. SIGMOD Record1999 ACM SIGMOD International Conference on Management of Data ACM, 1999. 61

[3] C.C. Aggarwal, and P.S. Yu. Redefining clustering for high-dimensional applications. IEEE Transactions on Knowledge and Data Engineering IEEE, 2002. 210

[4] A. Brazma, A. Robinson, G. Cameron, M. Ashburner. One-stop shop for microarray data - Is a universal, public DNA-microarray database a realistic goal?. Nature 17 Feb. 2000: 699-700.

[5] G. Dong , J. Li. Efficient mining of emerging patterns: discovering trends and differences. Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining San Diego, California, United State: 1999. 43-53.

[6] G. Dong, X. Zhang, L. Wong, and J. Li. CAEP: classification by aggregating emerging patterns. Discovery Science. Second International Conference, DS'99. Proceedings (Lecture Notes in Artificial Intelligence Vol.1721) Ed. Arikawa, S.; Furukawa, K. Berlin, Germany Germany: Springer-Verlag, 1999. 30

[7] J. Li, L. Wong. Emerging Patterns and Gene Expression Data. Proceedings of 12th Workshop on Genome Informatics Tokyo, Japan: 2001. 3-13.

[8] J. Li and L. Wong. Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. Bioinformatics 18.5 (2002): 725-734.

[9] J. Li and L. Wong. Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns (Corrigendum). Bioinformatics 18.10 (2002): 1406-1407.

[10] J. Li, H. Liu, J.R. Downing, A.E. Yeoh, and L. Wong. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. Bioinformatics 19.1 (2003): 71-78.

[11] NCBI. A Science Primer. 2003 (http://www.ncbi.nlm.nih.gov/About/primer/index.html)

[12] National Cancer Institute. Understand CGAP. 2003. (http://press2.nci.nih.gov/sciencebehind/cgap/cgap01.htm)

[13] U.S. Department of Energy Office of Science. Human Genome Project Information. 2003 (http://www.ornl.gov/TechResources/Human_Genome/home.html)