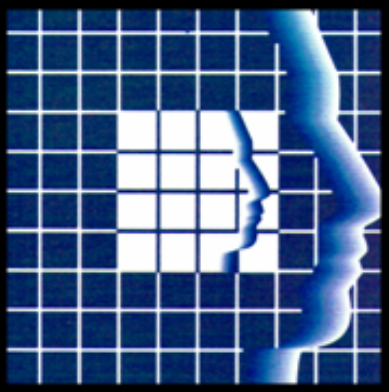


AIME 2007

www.aimedicine.eu/aime07

Working Notes

IDAMAP 2007

Intelligent Data Analysis in bioMedicine and Pharmacology

Carlo Combi and Allan Tucker

July 8, 2007

IDAMAP 2007 Intelligent Data Analysis in bioMedicine and Pharmacology Artificial Intelligence in Medicine 2007, Amsterdam

Carlo Combi and Allan Tucker (Chairs)

1 Introduction

Welcome to IDAMAP 2007, the twelfth workshop on Intelligent Data Analysis in Biomedicine and Pharmacology, held in conjunction with the eleventh conference on Artificial Intelligence in Medicine 2007 in Amsterdam.

The IDAMAP workshop series is devoted to computational methods for data analysis in medicine, biology and pharmacology that present results of analysis in the form communicable to domain experts and that somehow exploit knowledge of the problem domain. Such knowledge may be available at different stages of the data-analysis and model-building process. Typical methods include data visualization, data exploration, machine learning, and data mining. This year's IDAMAP will spend specific, although not exclusive, attention to classification and feature selection methods.

Gathering in an informal setting, workshop participants will have the opportunity to meet and discuss selected technical topics in an atmosphere which fosters the active exchange of ideas among researchers and practitioners. The workshop is intended to be a genuinely interactive event and not a mini-conference, thus ample time will be allotted for general discussion.

2 Program

The scientific program includes a selection of long papers and short papers presented throughout the workshop with the following themes:

- *Probabilistic and Bayesian Analysis*
- *Feature Selection / Reduction and Visualisation*
- *Classification and Filtering*
- *Temporal Datamining / Information Retrieval*

We are delighted to have an invited talk from Dr Jose Peña of Linköping University, Sweden who is an expert in probabilistic models and feature selection, particularly in the field of gene expression data.

3 Program Committee

- Ameen Abu-Hanna, Academic Medical Center, Amsterdam, The Netherlands
- Riccardo Bellazzi, University of Pavia, Italy
- Carlo Combi, University of Verona, Italy (chair)
- Janez Demsar, University of Ljubljana, Slovenia
- Michel Dojat, Universite Joseph Fourier, Grenoble, France
- Dragan Gamberger, Rudjer Boskovic Institute, Croatia
- Werner Horn, Medical University of Vienna, Austria
- John H. Holmes, University of Pennsylvania School of Medicine, USA
- Jim Hunter, University of Aberdeen, UK
- Elpida Keravnou-Papaeliou, University of Cyprus, Cyprus
- Matjaz Kukar, University of Ljubljana, Slovenia
- Pedro Larranaga, University of the Basque Country, San Sebastian, Spain
- Nada Lavrac, J. Stefan Institute, Slovenia
- Xiaohui Liu, Brunel University, UK
- Peter Lucas, Radboud University Nijmegen, The Netherlands
- Marco Masseroli, Politecnico of Milan, Italy
- Silvia Miksch, Danube University Krems, Austria
- Lucila Ohno-Machado, Harvard Medical School and M.I.T., Boston, USA
- Niels Peek, Academic Medical Center, Amsterdam, The Netherlands
- Marco Ramoni, Harvard Medical School, Boston, USA
- Steve Rees, Aalborg University, Denmark
- Paola Sebastiani, Boston University School of Public Health, USA
- Yuval Shahar, Ben-Gurion University of the Negev, Israel
- Stephen Swift, Brunel University, UK
- Allan Tucker, Brunel University, UK (chair)
- Blaz Zupan, University of Ljubljana, Slovenia

4 Acknowledgements

We would like to thank the invited speaker, all of the paper authors and all of the members of the program committee. We would also like to thank Ameen Abu- Hanna (Workshop and Tutorial Committee Chair), and the members of the AIME Local Organization Committee, Niels Peek, Gita Guggenheim, Ellen Lustenhouwer-Werring and Winston Gary Tjon Sjoe Sjoe (Webmaster), for their support in the organization of IDAMAP 2007.

IDAMAP 2007 - Scientific programme

Sunday, July 8th, 2007

9:00 am **Opening of IDAMAP Workshop**
Carlo Combi and Allan Tucker

9:15 am **Invited presentation**
Jose Pena

10:00 am **Paper session: *Probabilistic and Bayesian Analysis***

*** *A Millinghoffer, G Hullam and P Antal*
On inferring the most probable sentences in Bayesian logic

*** *E Peeling, A Tucker and PAC 't Hoen*
Discovery of local regulatory structure from microarray gene expression data using Bayesian networks

* *M van Gerven*
Tensor Decompositions for Probabilistic Classification

10:50 am **Break**

11:20 am **Paper session: *Feature Selection / Reduction and Visualisation***

*** *S Swift, A Tucker and M Hirsch*
Improving the Performance of Consensus Clustering Through Seeding: An Application to Visual Field Data

*** *C Fuchsberger, C Chan, S Ongarello, M Sips, I Feuerstein, A Pelzer, G Bonn, G Bartsch and H Klocker*
Visual Feature Selection in Biological Time-Series for Mass Spectrometry based Biomarker Discovery

* *D Klimov and Y Shahar*
Intelligent Visualization of Temporal Associations for Multiple Time-Oriented Patients Records

* *F Portet, F Gao, J Hunter and R Quiniou*
Reduction of Large Training Set by Guided Progressive Sampling. Application to Neonatal Intensive Care Data

12:30 pm **Lunch**

IDAMAP 2007 - Scientific programme

Sunday, July 8th, 2007

2:00 pm **Paper session: *Classification and Filtering***

*** *O Luaces, F Taboada, GM Albaiceta and A Bahamonde*
The Weight of Variable Groups for the Prediction of Probability of Survival in ICU Patients

*** *M Kukar, D Tzikas and A Likas*
Using Kernel Based Classifiers for Reliable Predictions in Medical Diagnostics

*** *V Moustakis, ML Laine, L Koumakis, G Potamias, L Zampetakis and BG Loos*
Modeling Genetic Susceptibility: a case study in periodontitis

*** *N Peek, M Verduijn, E de Jonge and B de Mol*
An empirical comparison of four procedures for filtering monitoring data

3:20 pm **Break**

3:40 pm **Paper session: *Temporal Datamining / Information Retrieval***

*** *R Azulay, R Moskovitch, D Stopel, M Verduijn, E de Jonge and Y Shahar*
Temporal Discretization of medical time series - A comparative study

*** *R Moskovitch, D Stopel, M Verduijn, N Peek, E de Jonge, and Y Shahar*
Analysis of ICU Patients Using the Time Series Knowledge Mining Method

*** *S Andreassen, A Zalounina, L Leibovichi and M Paul*
Learning susceptibility of a pathogen to antibiotics using data from similar pathogens

* *D Schmidt, G Lindemann and T Schrader*
First Steps towards an Intelligent Catalogue within the Open European Nephrology Science Center – OpEN.SC

* *MJ O'Connor, RD Shankar and AK Das*
Reusable Semantic Web-based Methods to Query Temporal Patterns: Application to Clinical Trials Management

* *C Larizza and P Ciccarese*
An Extensible Software Framework for Temporal Data Processing

* *G Tusch, M O'Connor, T Redmond, R Shankar and A Das*
SPOT – Utilizing Temporal Data for Data Mining in Medicine

* *L Sacchi, R Bellazzi, S Quaglini, A Sinico and G Moroni*
Temporal Rules to Predict Renal Flares in Lupus Nephritis

5:30 pm **Closing**

6:00 pm **Canal Tour**

Timing of presentations:

Invited talks: 35 minutes + 10 minutes discussion
Long presentations (***): 15+5 minutes
Short presentation (*): 8+2 minutes

Invited presentation

Three Feature Selection Problems (with Solutions)

Jose M. Peña
Department of Physics, Chemistry and Biology
Linköping University, 58183 Linköping, Sweden
jmp@ifm.liu.se, www.ifm.liu.se/~jmp, +46 13 281651

Abstract

I will discuss feature selection for three different goals commonly pursued in bioinformatics and biomedicine: Learning the a posteriori class distribution, learning the Bayes classifier, detecting any feature carrying information about the class. I will show that the optimal feature set changes depending on the goal. I will describe algorithms for solving the three problems and discuss their consistency and assumptions. I will also show some results on gene expression data.

Paper session:

Probabilistic and Bayesian Analysis

On inferring the most probable sentences in Bayesian logic

András Millinghoffer, Gábor Hullám and **Péter Antal***
Department of Measurement and Information Systems
Budapest University of Technology and Economics

Abstract

Vast literature and accumulating data in biomedicine creates a challenging problem. On the one hand, the literature and the curated ontologies, as a huge logical knowledge base, offer tremendous amount of raw factual knowledge. On the other hand, the biological data and its Bayesian probabilistic modeling bring in inherent uncertainty about models, model properties, or predictions. The fusion of such textually oriented logical knowledge and complex probabilistic models prompted active research, including research on probabilistic first-order logic or on more powerful probabilistic models.

In the paper, we introduce a method for fusing logical knowledge bases and complex, multivariate distributions inducing probability for first-order sentences. We present an extended first-order logic language with predicates and functions oriented towards graphical models. Furthermore, within this framework, we formulate the concept of "most probable sentences", which is a first-order generalization of the "most probable explanation" problem. We characterize the approaches, present the statistical analysis of the problem, and describe integrated "search-and-estimate" methods. Finally we report preliminary results for a real world medical problem.

1 Introduction

Whereas the new high throughput measurements are transforming biomedicine into a data rich science, the pace of the growth of the biomedical literature is breathtaking as well. The integrated usage of biomedical literature and statistical data poses many challenges, such as (1) the application of text mining methods for domain exploration, (2) the construction of priors based on the literature for Bayesian statistical data analysis, and (3) the literature based interpretation and evaluation of data analysis.

In the paper we introduce a novel framework for the semi-automated support of these problems. The proposed Bayesian logic (BL) framework fuses the factual knowledge bases with the result of Bayesian statistical data analysis. We present a formal semantics and the main inference problem. We overview approaches to this inference problem, its statistical aspects, and computational methods.

The paper is organized as follows. In Section 2 we overview works on probabilistic logic. Section 3 presents the formal definition of the semantics of the hybrid BL knowledge base (i.e., including logical and probabilistic parts). In Section 4 we introduce the basics of the proposed knowledge representation language and demonstrate its support for domain exploration, prior construction and evaluation. Section 5 defines the main type of inference in this BL framework and approaches to this inference problem including Monte Carlo methods and search techniques. Section 6 discusses the consequences of using Monte Carlo estimates. Finally Section 7 describes two algorithms and Section 8 illustrates their usage in the the area of ovarian cancer.

2 From Bayesian network features to probabilistic logics

To formalize the proposed hybrid knowledge, we adopt the Bayesian framework and use the Bayesian network model class (see e.g. [4; 11] and [20]). Bayesian networks (BNs) use directed acyclic graphs (DAGs) to represent a probability distribution and optionally the causal structure of the domain. In an intuitive causal interpretation, nodes represent the uncertain quantities, edges denote direct causal influences, defining the model structure. Local probabilistic models are assigned to each node quantifying the stochastic effect of its parents (causes). The descriptors of the local models give the model parameters. The widespread popularity of Bayesian networks follows from the fact that this representation addresses jointly three autonomous levels of the domain: the causal model, the probabilistic dependency structure, and the distribution over the uncertain quantities.

Two fundamental approaches emerged for the successful application of such complex models in biomedical domains with relatively scarce amount of data: Bayesianism and the usage of model properties (i.e. features). Indeed, the conceptualization of the posterior $p(G|D)$ over the set of structures ($\mathcal{G}$) as a probabilistic knowledge base was proposed from the beginning of the field [5; 6]. Practical methods for the joint application of Bayesianism and model properties

*The authors thank Tadeus Dobrowiecki for his helpful comments.

were proposed in [10; 12; 18]. Given a DAG structure G of a Bayesian network over the set of domain variables V, we define a *structural feature* $F(G)$ as a property of the structure G. In the Bayesian approach the posterior of the values $\{f_1, \ldots, f_k\}$ of $F(G)$ given data D_N can be computed as:

$$p(f_i|D_N) = \sum_{G:F(G)=f_i} p(G|D_N), \qquad (1)$$

i.e. the posterior of a structural feature taking a certain value f_i is the sum of the posteriors of possible DAGs where the structural feature takes the specified value. Such features include directed edges with binary values (indicating the presence or absence of the edge) or parental sets with values representing the possible subsets of variables.

The common approach to BN feature learning assumes that the set of feature functions provides a computationally tractable (i.e., linear or quadratic in the number of variables) characterization of the overall domain (e.g., see [10]). The other approach provides a more complete view on certain properties of the model (e.g., focusing on classification [2]).

However, the knowledge base view of the posterior allows a richer application. For the fusion of factual (free-text) and uncertain knowledge based on data, we proposed a method summarized in Section 3 to embed analytically intractable posteriors into a logical knowledge base [3; 2].

Related work can be grouped as research on probabilistic logics and on the generalization of Bayesian networks towards first-order logic (FOL) (for a recent overview see e.g. [7]). One of the early works in the first group attempted to combine logic and probability [14], which defines the probabilistic knowledge base from elementary probabilistic building blocks. The BLOG (Bayesian Logic) language and Markov logic networks are also members of the first-order probabilistic logic family [9; 19]. The former uses the concept of a probability distribution over a set of possible worlds, while the latter defines a first-order knowledge base which specifies a ground Markov network given a set of constants representing objects in the domain. The concept of Relational Bayesian networks [15] is another possible approach. The main idea is to represent every predicate with a node in the network and to assign a probability formula to them describing their conditional probability distribution.

Following the proposed possible world interpretation from [14], we specialize this general approach for the sake of practical applicability. We restrict the knowledge base to a voluminous factual part consisting of established ontologies and papers from the domain and to an uncertain part defined by an arbitrary distribution over Bayesian network structures with fixed set of domain variables. Furthermore, we define a language with Bayesian network oriented functions.

3 The hybrid probabilistic-logical semantics

Though the mentioned representations attempt to unify the expressive power of logic and probability, in their final forms they implement a pure probabilistic semantics, while logical elements play role only at the model construction phase. In our work, we propose a representation which encapsulates different knowledge sources, some with probabilistic, some with logical semantics.

The probabilistic and the causal interpretations of BNs ensure that structural features can express a wide range of relevant concepts based on conditional independence statements and causal assertions [20; 21]. To enrich this approach with subjective domain knowledge via free-text annotations, we introduced the concept of Probabilistic Annotated Bayesian Network knowledge base (PABN-KB) [2].

Definition 1 *A Probabilistic Annotated Bayesian Network Knowledge Base K for a fixed set $\boldsymbol{V}$ of discrete random variables is a first-order logical knowledge base using standard graph, string and BN related predicates, relations and functions. Let G represent a target DAG structure including all the target random variables. The knowledge base includes free-text descriptions for the subgraphs and for their subsets. We assume that the models of the knowledge base differs only w.r.t. G (i.e. there is a bijection $G \leftrightarrow M$) and the distribution $p(G)$ is available.*

For a sentence α in K, its probability is defined as the expectation of its truth function

$$\mathrm{E}_{p(M|K)}[1(\alpha, M)] = \sum_{G} 1(\alpha, M(G))p(G|K), \qquad (2)$$

where $1(\alpha, M)$ denotes the α's truth-value in the model M and $M(G)$ denotes the model defined by G. This hybrid approach defines a distribution over the set of models $\mathcal{M}$ by combining a logical knowledge base with a probabilistic model. The logical knowledge base describes the certain knowledge in the domain defining a set of models (legal worlds) and the probabilistic part ($p(G)$) expresses the uncertain knowledge over these worlds.

Note that the logical knowledge base usually excludes a priori certain structures G, so only an unnormalized distribution is available. However, this is not a serious restriction, since $p(G)$ usually is an unnormalized posterior.

4 A graphical model oriented FOL language

The basic probabilistic semantics of Bayesian networks supports the formalization of two kinds of inference: (1) predictive inference means the computation of probabilities conditional on the known instantiation of some variables, i.e. it can be used for the evaluation of isolated observation cases, (2) in case of parametric inference, we formulate statements about the overall domain (i.e. the parametrization and structure of the model).

Feature learning is also a subset of this latter case, hence the query language over a PABN-KB consists of statements about the parametric, or more typically the structural features of the model. Questions may contain grounded or quantified closed sentences as well, e.g. *"the Markov Blanket[1] (MB) of variable X consists of the variables $Y_1, ..Y_N$"* (ground sentence), or *"the Markov Blanket of variable X contains at least N variables"*.

[1]The Markov Blanket of a variable is the set which probabilistically isolates it from the rest of the model. In a Bayesian network it consists of the parents and the children of the node, and the other parents of the children [20].

The shortcoming of such a purely "model-based" language is that its sentences may be difficult to interpret. The usage of the previously introduced annotations may help users better understand the result of the query. If we suppose for example that every variable belongs to one of a set of disjoint classes, the statement could be *"The Markov Blanket of variable X contains at least one member of every class"* ($\forall class \in Classes, \exists Y \in class : Y \in MB(X)$). Of course these queries may contain any binary relation, as in our case $member$ ($\in$) over the annotations.

The above first-order language can be further extended by using external knowledge sources (e.g. publication repositories). These can be incorporated into the knowledge base by including references of their elements in the textual annotations, then relations of these outer sources can be applied to the elements of our models as well. The occurence of terms within scientific publications can supplement queries, an example for this could be: *"The Markov Blanket of variable X contains at least one such member of every class which was cited in publications before date D"*.

Summarizing, the elements of such a first-order logic oriented towards annotated graphical models (AGM-FOL) are the following:

- Possible worlds are Bayesian network structures, the probability of each is defined as the posterior of the model given observation data $p(G|D)$. Any logical predicate can be applied to these models, forming statements about their structural features, e.g. Markov blankets.
- This probabilistic foundation can be supplemented by any kind of logical knowledge sources, like ontologies or publication repositories.
- There are free-text or structured (e.g. XML) annotations assigned to model elements, through which first-order statements can be formulated about the models and outer elements are linked to models.
- Sentences can be built from functions and predicates, such as standard graph and set relations over model elements (like $directededge$, $directedpath$, $MarkovBlanket$, etc.) and string functions over annotations (like $contains$),

5 The problem of the most probable sentences

Probabilistic ABN-KB defines the truth value (i.e., its probability) of any given target well-formed first-order formula in K. Frequently, however, we are interested in a larger set of target sentences in K, for example if the grounding of the closed sentence is subject to change. This suggests the following definition.

Definition 2 *Let $\mathcal{S}$ denote the set of well-formed first-order, closed formulas in an annotated graphical model oriented FOL language L. The problem of the* most probable sentences *(MPS) consists of the identification of the K most probable sentences $\mathcal{S}_K \subseteq \mathcal{S}$ and optionally the computation of their truth values according to Eq.2. If only groundings are subject to change in sentence templates defining $\mathcal{S}$, then we call it the* most probable groundings *problem. Using a set based loss function to quantify the selection of $\mathcal{S}_K$, we talk about the* sentence subset selection *problem.*

This definition generalizes the *most probable explanation* problem (MPE) and the *feature subset selection* problem (FSS) (e.g., see [20] and [16] respectively). The MPE problem aims at identifying the most probable instantiations of target variables given certain evidences (in this case the formalization of the annotated Bayesian network knowledge base is different, because it is defined over values instead of structures). The FSS problem aims to select the most relevant input (i.e., independent) variables in a conditional modeling framework.

Note that it is possible that a sentence α is a tautology or entailed by the factual knowledge base K, so it has probability 1. Indeed, the purpose of the so-called target set is to restrict the set of examined sentences to the relevant ones for a particular question in the domain, which are not fully determined by the factual part of the knowledge base.

The approach to the MPS problem depends on two factors. The first is the number of models $|\mathcal{M}|$ and the distribution over them, because in lack of analytic solution, Monte Carlo (MC) methods or possibly Markov Chain Monte Carlo (MCMC) methods have to be applied to approximate Eq. 2. The second factor is the number of target sentences $|\mathcal{S}|$, because search techniques have to be used in case of a high cardinality of $\mathcal{S}$. Table 1 summarizes the classification of approaches.

Table 1: Approaches to the MPS problem depending on the number of the models and of the target sentences.

	$\|\mathcal{S}\|$ is small	$\|\mathcal{S}\|$ is large
small $\|\mathcal{M}\|$	exact computation for all sentences (Alg. 1)	search over sentences using exact score
large $\|\mathcal{M}\|$	MC estimation for all sentences (Alg. 2)	search over sentences using MC estimated score (Alg. 3)

For small $|\mathcal{M}|$ and $|\mathcal{S}|$ we can enumerate all the possible/allowed sentences, and evaluate their probability exactly according to Eq. 2, as shown in Alg. 1.

Algorithm 1 Exact computation of the posterior of every sentence

Require: PABN-KB(K, $p(G)$), $\mathcal{S}$
Ensure: Exact posteriors: $\forall s \in \mathcal{S}$: $p(s)$
  for all $s \in \mathcal{S}$ **do**
    for all $G \in \mathcal{G}$ **do** {enumerate worlds}
      if $\mathcal{M}(K \wedge G) \neq \emptyset$ **then**
        if s true in $K \wedge G$ **then**
          $p(s)$ += $p(G)$

More efficient algorithms are reported in Section 7.

6 On the sample complexity of MPS

Before the computational considerations of the search problem, let us investigate the statistical consequences of using MC estimates for the truth values (i.e., probabilities) of the sentences in a MPS problem (i.e., estimates of the expectations in Eq. 2). Specifically, we investigate the effect of the cardinality of the target sentences $|\mathcal{S}|$ on the mean error of the selected sentences $\mathcal{S}_K \subseteq \mathcal{S}$. Initially, we assume that an i.i.d. data set D_N is available containing N samples from the target distribution specified in the PABN-KB K.

The relative frequencies of the sentences $s_i \in \mathcal{S}$ are denoted with $\hat{\mathcal{P}}_i$. The loss of reporting the sentences selected by the $|\mathcal{S}|$ dimensional binary vector I is

$$L(I) = L(\mathcal{P}, I) = \sum_i I_i L(s_i), \text{ where } L(s_i) = 1 - \mathcal{P}_i, \tag{3}$$

and $\hat{L}(I), \hat{L}(s_i)$ denote the corresponding estimated losses based on $\hat{\mathcal{P}}$. The decision rule $\delta(D_N) = I_N^*$ is defined as $I_N^* = \arg\min_{I \in \mathcal{I}^K} L(\hat{\mathcal{P}}(D_N), I)$, where $\mathcal{I}^K$ denotes the set of $|\mathcal{S}|$ dimensional binary vectors with exactly K ones. Let $I^* = \arg\min_{I \in \mathcal{I}^K} L(\mathcal{P}, I)$ denote an optimal set (i.e., it selects the empirically[2] most probable sentences). The error is defined as $\frac{1}{K} L(I_N^*) - \frac{1}{K} L(I^*)$. Then analogously as in classifier selection [8],

$$\begin{aligned}
&\tfrac{1}{K} L(I_N^*) - \tfrac{1}{K} \min_{I \in \mathcal{I}^K} L(I) \\
&= \tfrac{1}{K} (L(I_N^*) - \hat{L}(I_N^*) + \underbrace{\hat{L}(I_N^*) - \hat{L}(I^*)}_{\leq 0} + \hat{L}(I^*) - L(I^*)) \\
&\leq \tfrac{1}{K} (L(I_N^*) - \hat{L}(I_N^*) + \hat{L}(I^*) - L(I^*)) \\
&\leq \tfrac{1}{K} |L(I_N^*) - \hat{L}(I_N^*)| + |\hat{L}(I^*) - L(I^*))| \\
&\leq 2 \max_{s \in \mathcal{S}} |p(s) - \hat{p}_N(s)|.
\end{aligned} \tag{4}$$

It means that if we can estimate uniformly well the probabilities of the sentences, then we can bound the error of the selected sentences. Using the Hoeffding inequality [8], we get for ϵ accuracy and δ confidence

$$\begin{aligned}
&p(|\frac{1}{K} L(I_N^*) - \frac{1}{K} \min_{I \in \mathcal{I}^K} L(I)| \geq \epsilon) \\
&\leq p(\max_{s \in \mathcal{S}} |p(s) - \hat{p}_N(s)| \geq \epsilon/2) \leq 2|\mathcal{S}| e^{-N\epsilon^2/2} \leq \delta,
\end{aligned}$$

which shows that the sample complexity is

$$N \geq 1/\epsilon^2 (2 \log(2|\mathcal{S}|) + 2 \log(1/\delta)). \tag{5}$$

Furthermore, the expected average error of the selected sentences can be bounded as follows using the inequality $\mathrm{E}[Z] \leq \sqrt{\frac{\log(ce)}{2N}}$ (which holds if $p(Z \geq \epsilon) \leq ce^{-2N\epsilon^2}$ for all $0 \leq \epsilon$ and some $0 \leq c$) [8]:

$$\mathrm{E}_{p(D_N)}[L(I_N^*) - L(I^*)] \leq \sqrt{\frac{\log(2|\mathcal{S}|) + 1}{N/2}}. \tag{6}$$

[2] We use the empirical term w.r.t. the stochastic simulations as well.

This shows that the sample complexity and the expected error is proportional to the logarithm of the cardinality of the set of target sentences $|\mathcal{S}|$. Note that here the cardinality of the set for selection $|\mathcal{S}|$ is independent of the sample size N.

This result was derived assuming an i.i.d. sample from the target distribution. Analogic results can be derived using MCMC variants of the Hoeffding inequality (e.g., see [13]).

7 The search-and-estimate scheme: finding relevant query instantiations

The exact computation of the probabilistic quantity of Eq 2 is usually not feasible by Alg. 1, because of the cardinality of the model space. Hence, instead of an exhaustive enumeration a sampling method (typically an MCMC method, see [11]) can be applied. The sampling (i.e. *estimation*) can be incorporated into the whole algorithm in several ways, considering its relation to the other main component, namely the *search* part.

Estimation posterior of every sentence If the cardinality of the search space (i.e. the possible instantiations and forms of the sentence skeleton) is tractable, the sampling can be placed within the search cycle, i.e. the probability of each examined atomic sentence is calculated separately, using a "dedicated" estimation run.

The main advantage of this approach is that the convergence and the confidence issues can be handled separately, sentence by sentence. On the other hand its drawbacks are that it presumes that the relevant sentence instantiations are available a priori.

Algorithm 2 Estimation posterior of every sentence

Require: PABN-KB(K, $p(G)$), $\mathcal{S}$
Ensure: Estimated posteriors: $\forall s \in \mathcal{S}$: $\hat{p}(s)$
  for all $s \in \mathcal{S}$ **do**
    N=0, $G = G_0$
    MCMC initialization
    repeat {MC sampling of worlds ($G \in \mathcal{G}$)}
      if $\mathcal{M}(K \wedge G) \neq \emptyset$ **then**
        if s true in $K \wedge G$ **then**
          $\hat{p}(s)$ += 1
      N++, G=MC-sampling
    until std.error($\hat{p}(s)$)$< \delta$
    Normalize $\hat{p}(s)$ with N

A variant of the above can have the sampling cycle done off-line, i.e. the sequence of the visited structures is created prior to the search phase and then is reused for every considered sentence. The scheme of this method is the same as of the previous one, offering a trade-off between the time and the space complexity. In this latter, the sampling phase has to be performed only once , but the storage of the sample has to be solved.

Estimation and search for the most probable sentences Though previously the sampling phase was placed within the search, this relation can be reversed: we perform one

"large" sampling cycle, during which we update the estimates of the examined sentences. The main advantage of this approach is that we do not have to make any considerations about the sentences prior to the estimation phase: each visited structure indicates a set of sentences, i.e. those which are $true$ in the current world. These sentences are then collected in a list, updating their cumulative probabilities in each estimation step. This method eliminates sentences with zero probability in a natural manner. In fact this approach can be conceived as a two phased sample-then-search method with a special search method exploiting the estimation steps and using increasing prefixes of an offline sample to decrease time complexity (see Alg. 3).

Algorithm 3 Estimation and search of sentences

Require: PABN-KB(K, $p(G)$), $\mathcal{S}$, L
Ensure: Estimated posteriors: $\forall s \in \mathcal{S}_K$: $\hat{p}(s)$
  N=0, $G = G_0$
  MCMC initialization
  repeat {MC sampling of worlds ($G \in \mathcal{G}$)}
    if $\mathcal{M}(K \wedge G) \neq \emptyset$ **then**
      $\mathcal{S}_K$ = $\mathcal{S}_K \cup$GetValidSentences(K,G)
      for all $S \in \mathcal{S}_K$ **do** {MCMC update}
        if s true in $K \wedge G$ **then**
          $\hat{p}(s)$ += 1
      N++, G=MC-sampling
      if $L < |\mathcal{S}_K|$ **then**
        $\mathcal{S}_K$=PruneToMostProbables($\mathcal{S}_K, L$);
  until std.error($\hat{p}(s)$)< δ

7.1 Informed search for sentences

The third "estimation and search" method leaves open the question of finding promising sentences (i.e., the implementation of the function $GetValidSentences(K, G)$). If the set of true instantiations in the extended knowledge base $K \wedge G$ can be computed efficiently, the result set is generated "exhaustively". If this set grows too large, then prior domain knowledge can be applied to direct the search (i.e., to select the most promising subset of sentences to be included in the Bayesian statistical update).

The pruning of the set of estimated sentences in $\mathcal{S}_K$ can be controlled in two ways. In the problem of the Most Probable Sentences, it is based on the expected probability of the sentences (i.e. sentences with a probability under a certain threshold will be dropped). In the Sentence Subset Selection problem, the overall loss of the set(!) of the estimated sentences has to be optimized (e.g., the loss function can express the overall probability and diversity of the set).

8 Results

To illustrate the above concepts, we have performed test queries in a the domain of ovarian cancer using 35 variables [1]. The posterior probabilities were estimated using 782 records. As auxiliary logical knowledge source we used a database containing 62716 references of articles returned by the PubMed system to the query *"ovarian AND (cancer OR tumor OR tumour OR mass)"* by April 2007.

The implemented system consisted of the following components: a standard C++ engine storing the Bayesian network models and controlling the MCMC methods, the auxiliary knowledge sources were stored in MySQL databases connected to the central engine through C++ wrappers. For the definition and the evaluation of the predicates we used the SWI Prolog engine.

The estimations were performed using Alg. 3. We compared two MCMC methods, the DAG-based sampling proceeded directly on complete structures as described in [12], and the ordering-based MCMC [10] (using 10^3 burn-in steps, 30000 samples, with maximum 4 parents and CH priors).

The tested predicate was the following:

"What is the probability that the MB set of the variable Pathology will contain at least one variable from each of the 5 variable classes and contain at least Y of the 5 variables of the class 'Vascular'?".

The most probable grounding (i.e., MB sets) are reported in Table 2.

Table 2: The four most probable Markov blanket sets of Pathology compatible with the target query (the probabilities are 0.062, 0.052, 0.035, and 0.033, and 0.039 0.026 0.023, and 0.021 respectively). The G and the $\prec$ symbols denote the sets from the DAG-based and the ordering-based MCMC methods. Variables $FamHist$, $HormTherapy$, $Parity$, $PMenoAge$, Age, $PMenoY$, $PillUse$, $FHOvCa$, and $Pain$ are never selected, and the variables $Bilateral$, $Ascites$, $PapFlow$, $WallRegularity$, $Shadows$, RI, $TAMX$ are always selected, so they are not reported.

	G_1	G_2	G_3	G_4	$\prec_1$	$\prec_2$	$\prec_3$	$\prec_4$
Meno	0	0	0	0	1	1	1	0
CycleDay	1	0	1	1	0	0	0	0
Volume	0	1	1	0	1	1	1	1
Fluid	1	1	1	1	0	0	0	0
Septum	0	1	1	0	1	1	1	1
ISeptum	1	0	1	0	0	0	0	0
Papillation	1	1	1	1	1	1	0	0
PSmooth	0	1	1	0	0	0	0	0
Locularity	0	0	0	1	1	1	1	1
Echog.	1	0	1	1	1	1	1	1
ColScore	0	0	0	0	1	1	1	1
CA125	0	0	0	0	1	1	1	1
PI	0	1	0	0	1	1	1	1
PSV	1	1	1	1	1	0	0	1
Hysterectomy	1	1	0	1	1	1	1	1
Solid	1	1	1	0	1	1	1	1
FHBrCa	1	0	0	1	1	1	1	1

9 Conclusion

In the paper, we introduced a method for fusing logical knowledge bases and multivariate distributions inducing probability for first-order sentences. We developed and illustrated an extended first-order logic language with predicates and functions oriented towards graphical models. We formulated the concept of the "most probable sentences" within this framework, which is a first-order generalization of the "most probable explanation" problem in Bayesian networks. We analyzed the sample complexity and the ex-

pected error for this problem, and characterized the computational approaches.

A novel system was introduced which encompasses and integrates the components of logical and probabilistic inference: different MCMC sampling schemes and wrappers for the factual knowledge sources. The separate components (e.g. prolog and SQL engines, C++ MCMC samplers) were integrated using a wrapper written in C++. The system was tested in the field of ovarian cancer, and currently we are deploying it to explore the genetic background of rheumatoid arthritis and asthma.

The results have shown that the method is capable of reporting relevant statements about a domain, however, the formulation of "good" questions is not trivial. For the sake of better usability the knowledge engineering aspects of the methodology have to be considered, e.g. by constructing query schemes which possibly yield interesting/relevant results.

An other extension of the introduced representation could target the use of hierarchical Bayesian networks as possible worlds, because many domains have a hierarchic/modular structure. Such domain can better be modeled by some extensions of Bayesian networks (e.g. Object-Oriented Bayesian Networks [17]). The hierarchical description can be formalized as a stochastic graph-grammar, which defines the probability of how the modules can be derived from each other. The availability of such a description allows the extension of the uncertain part of the proposed hybrid knowledge base from Bayesian network structures to include their hierarchical derivation.

References

[1] P. Antal, G. Fannes, Y. Moreau, D. Timmerman, and B. De Moor. Using literature and data to learn Bayesian networks as clinical models of ovarian tumors. *AI in Med.*, 30:257–281, 2004.

[2] P. Antal, G. Hullám, A. Gézsi, and A. Millinghoffer. Learning complex bayesian network features for classification. In *Proc. of third European Workshop on Probabilistic Graphical Models*, pages 9–16, 2006.

[3] P. Antal and A. Millinghoffer. A probabilistic knowledge base using annotated bayesian network features. In *Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*, pages 1–12, 2005.

[4] J. M. Bernardo. *Bayesian Theory*. Wiley & Sons, Chichester, 1995.

[5] W. L. Buntine. Theory refinement of Bayesian networks. In *Proc. of the 7th Conf. on Uncertainty in Artificial Intelligence (UAI-1991)*, pages 52–60. Morgan Kaufmann, 1991.

[6] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[7] J. Cussens. *Statistical Relational Learning*, chapter Logic-based formalisms for statistical relational learning. MIT Press, Cambridge, MA, 2007.

[8] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, Berlin, 1996.

[9] P. Domingos and M. Richardson. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[10] N. Friedman and D. Koller. Being Bayesian about network structure. In *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence(UAI-2000)*, pages 201–211. Morgan Kaufmann, 2000.

[11] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, London, 1995.

[12] P. Giudici and R. Castelo. Improving Markov Chain Monte Carlo model search for data mining. *Machine Learning*, 50:127–158, 2003.

[13] P. Glynn and D. Ormoneit. Hoeffding's inequality for uniformly ergodic markov chains. *Statistics and Probability Letters*, 56:143–146, 2002.

[14] J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.

[15] Manfred Jaeger. Relational bayesian networks. *Proc. of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-1997)*, 1997.

[16] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[17] D. Koller and A. Pfeffer. Object-oriented Bayesian networks. In Dan Geiger and Prakash P. Shenoy, editors, *Proc. of the 13th Conf. on Uncertainty in Artificial Intelligence (UAI-1997)*, pages 302–313. Morgan Kaufmann, 1997.

[18] D. Madigan, S. A. Andersson, M. Perlman, and C. T. Volinsky. Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. *Comm.Statist. Theory Methods*, 25:2493–2520, 1996.

[19] B. Milch, B. Marthi, and S. Russell. Blog: Relational modeling with unknown objects. In *Proc. 21th Int. Conf. on Machine Learning (ICML)*, pages 157–165, 2004.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, CA, 1988.

[21] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Discovery of local regulatory structure from microarray gene expression data using Bayesian networks

Emma Peeling and **Allan Tucker**
Centre for Intelligent Data Analysis
Brunel University
Uxbridge, Middx, UB8 3PH, UK
emma.peeling@brunel.ac.uk

Peter A.C. t'Hoen
Center for Human and Clinical Genetics
Leiden University Medical Center
Postzone S-04-P, PO Box 9600,
2300 RC Leiden, Netherlands

Abstract

Modelling gene regulatory networks is a key goal in bioinformatics research. In this paper we use Bayesian networks to identify regulatory genes and their targets. We focus on identifying the regulatory structure at a local level for individual target genes, taking into account multiple regulators acting in combination and other influencing factors such as disease type by incorporating additional parent nodes in the networks. We use a well-studied regulatory module in E. coli to validate our method prior to focusing on microarray expression data that has been generated in order to explore different types of muscular dystrophy. We use the prediction of gene expression measurements to evaluate learnt networks. Our results indicate that the most suitable regulatory model (single or multiple regulators; incorporation of disease type) varies with each individual gene - a conclusion that makes sense biologically. The results also show that the majority of genes are best modelled using a simple single-regulator network, indicating that it is only a smaller subset of genes that require more complex co-regulation models.

1 Introduction

Gene Regulatory Networks (GRNs) describe how the expression level of genes effect, or regulate, the expression of other genes. Modelling these networks is a topic of great interest in current bioinformatics research. At the most basic level, regulatory interactions occur where transcription factors (TFs - regulator genes) *activate* (turn on) or *repress* (turn off) the expression of certain genes. More complex interactions involve feedback loops: TFs can regulate themselves. TFs can control many genes (their *targets*) and in turn each gene may be regulated by a set of TFs acting in combination.

Gene expression can be measured using DNA microarrays - an experimental technique that allows the expression of thousands of genes to be measured simultaneously. In this paper, we use Bayesian networks to learn local regulatory interactions between genes from microarray expression data that has been generated in order to explore different types of muscular dystrophy, a muscle wasting disorder.

Bayesian Networks (BNs) [Pearl, 1991] have become a popular method for computational modelling of GRNs from expression data [Friedman *et al.*, 2000; Hartemink *et al.*, 2002; Pe'er *et al.*, 2006] since they are able to represent the network qualitatively (with a network graph) and quantitatively (probability distributions quantify the strength of influences and dependencies between nodes/variables in the network graph) and thus are relatively easy to interpret by non-statisticians (e.g. biologists). Prior to BNs, most analyses performed on gene expression data were clustering techniques used to extract groups of co-regulated genes. However, clustering only extracts groups of correlated genes and not the regulatory network structure. BNs are able to discover more complex, nonlinear relationships and transparently represent the nature of interactions (for example, *how* regulators act in combination) through their conditional probability distributions.

Our research focuses on identifying the nature and structure of regulatory interactions at a local level - in other words, which and how many regulators interact together to control the expression of a certain gene and how they do so. Many genes are regulated by more than one TF and modelling this is a key goal of our research. TFs may interact in a number of different ways to effect regulation - for example, a pairing of TFs may work where one is an activator and another is a repressor. Whilst some work on BNs for modelling regulatory networks has considered multiple regulators, this has often been within the scope of global regulatory networks [Segal *et al.*, 2003] where interactions between 'modules' of genes is the focus. Our method aims to to discover a more detailed regulatory structure at a lower level, focusing on individual interactions.

More recently, [Yeang and Jaakkola, 2006] have used a method based on the use of conditional probability functions to model regulatory control by multiple TFs. This is similar to the BN framework, but the regulatory programs are not explicitly represented using the network formalism. Whilst we use the BN conditional

probability distributions in a similar way, to model the combining nature of multiple regulators and to predict gene expression values, our method differs from this work as we also consider the effect of class information (such as disease types) by incorporating it as a node in the BN structure. Previously, [Tucker *et al.*, 2005] have used BN classifiers to identify disease types in muscular dystrophy, but class information such as this has not been incorporated into regulatory network models previously.

It should also be noted that the research presented in this paper constitutes the initial step of a longer-term project on modelling gene regulation in muscular dystrophy. Further work as part of the project will involve modelling temporal information through time nodes and dynamic Bayesian networks, the use of hidden nodes to model unobserved variables, and the incorporation of other data sources or expert knowledge (such as multiple microarray gene expression datasets, TF binding sites, the gene function ontology and textual information extracted from scientific literature).

This paper is organised as follows. Section 2.1 introduces the concept of conditional independence and BNs. In section 2.2 we outline how we can use BNs to model gene regulatory networks. In section 2.3 we describe the algorithm used to discover local regulatory structure. In sections 3 and 4 we discuss our results from the application of our method to an E.coli dataset and two different muscular dystrophy gene expression datasets. Finally in section 5 we present our conclusions.

2 Methods

2.1 Conditional independence and Bayesian networks

The concept of conditional independence between sets of variables or data is a key underlying principle of our algorithm. Suppose we have three random variables X,Y and Z. Then, X and Z are conditionally independent given Y if $p(X|Z,Y) = p(X|Y)$. A more intuitive description of conditional independence would be to say that once we know the value of Y, then X and Z become independent as any further information about Z will not change the outcome of X.

BNs are graph-based models of probability distributions that capture properties of conditional independence between variables. A BN consists of two components. The first is a Directed Acyclic Graph (DAG) consisting of links between nodes that represent variables in the domain. If there is a link from node A to another node B, then A is said to be a *parent* of B, and B is a *child* or *descendant* of A. A link between nodes indicates a direct influence between the parent and child variables. The second component is a set of conditional probability distributions associated with each node that quantify the strength of influence from its parent nodes. Prior probability distributions are specified for root nodes (those without parents). Probability distributions may be modelled by discrete (tabular) or continuous (e.g. Gaussian) distributions.

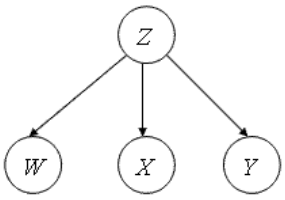
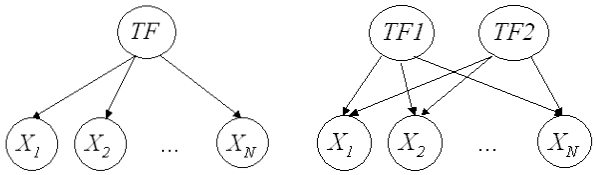


Figure 1: Conditional independence in Bayesian networks: the nodes W, X and Y are conditionally independent given the value of Z.



(a) Standard regulatory model: single regulator

(b) Multiple regulators

Figure 2: (a) shows a standard gene regulatory network, involving a TF parent node to target genes X_i. (b) is a modified version of the same network, but with an additional parent node representing a second regulating gene.

It can be shown that each node is conditionally independent of all its non-descendants given its parents [Pearl, 1991]. For example, for the network shown in Figure 1 the nodes W, X and Y are conditionally independent given the value of their parent node Z.

2.2 Bayesian networks to model gene regulation

The notion of conditional independence can be applied to gene regulation. It makes sense that genes which are regulatory in nature (TFs) will render the genes that they control independent. Therefore, we can represent a simple regulatory structure involving a TF and its target genes using a BN.

Suppose we have a set of target genes X_i and a regulatory gene TF. Then a network representing this can be formed with the TF variable as a parent node to the X_i nodes, as shown in Figure 2a. In this case we have that the X_i are conditionally independent on the TF. The network structure also makes sense in terms of links between nodes i.e. the TF directly influences the values of the target genes. For control by multiple regulators, additional parent nodes rep-

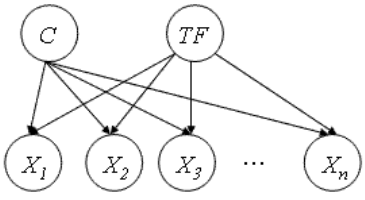


Figure 3: Incorporation of a class node to represent the sample type

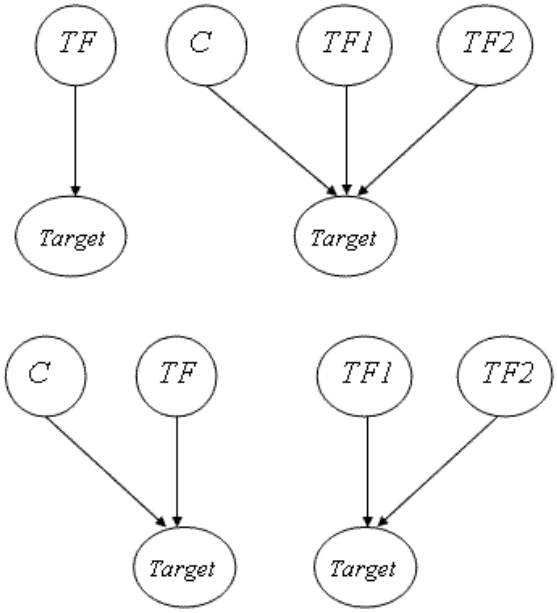


Figure 4: Local regulatory structure types (clockwise from top left): single regulator, two regulators with class, single regulator with class, two regulators

resenting TFs can be added, as shown in Figure 2b. Modelling class (e.g. disease type from different samples, including control/healthy samples) can be easily incorporated by using a class node to represent disease type, as shown in Figure 3.

2.3 Learning the local regulatory structure

For each target gene we perform a search for the type of regulatory structure (with/without class node, number of TFs/regulator genes - see Figure 4), over all combinations of candidate regulator genes. With a preselected set of candidate TFs, it is possible to perform an exhaustive search.

We use the *Bayesian Information Criterion* (BIC) to score candidate structures. The BIC function is a combination of the model log-likelihood and a penalty term that favours less complex models - as such it is similar to the minimum description length:

$$BIC = log\ P(\theta|D) - \ 0.5\ k\ log(n)$$

where θ represents the model, D is the data, n is the number of observations (sample size) and k is the number of parameters. $log\ P(\theta|D)$ is the log-likelihood while the term $0.5\ k\ log(n)$ is a penalty term, which specifically penalises more complex models with more parameters. The BIC is good for dealing with small samples of data as is common with microarray data, as the penalty term helps to prevent overfitting.

For each target gene, we can identify the highest-scoring combinations of TF(s) for each structure type (e.g. single and multiple regulators; with or without class information incorporated) and thus compare the different models of regulatory structure.

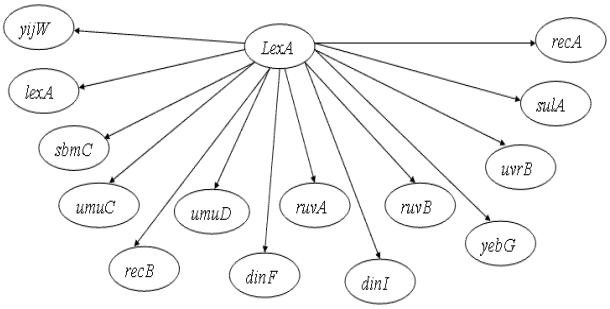


Figure 5: Ecoli SOS network module, regulated by the repressor LexA. Each of the 14 target genes are significantly differentially expressed between wild type and mutant samples.

3 Evaluation

We have evaluated our method on an E. coli dataset and two Muscular Dystrophy (MD) datasets (human and mouse models). The E. coli dataset is a smaller problem, restricted to one network module with a single known TF. We use the E. coli dataset to validate our algorithm before applying it to the more complicated MD data. This section contains details of the datasets, experiments performed, and validation methods used. The experiment results are discussed in the following section 4.

3.1 Datasets

For each dataset the gene expression values were discretised into three states using an equal frequency method. Discretisation allows complex interactions to be captured using relatively simple conditional probability distributions. We used three states as a balance to mitigate against a loss of information but also avoid overfitting.

E. coli data

We consider an example of a single network module - an SOS repair system that includes > 30 genes with one transcriptional regulator, LexA. Our aim is to evaluate whether the known TF, LexA, can be correctly identified for each target.

The dataset consists of two gene expression time series for UV exposure, one for wild-type (healthy) cells and one for lexA1 mutants, which are unable to induce genes under LexA control. [Khanin *et al.*, 2006] find that fourteen of the target genes in the SOS repair system are significantly differentially expressed between healthy and mutant cells so we focus on these, shown in Figure 5. A set of 101 candidate TFs was selected from the RegulonDB database [Salgado *et al.*, 2006].

Mouse MD data

Muscular dystrophies are a heterogeneous group of inherited disorders characterised by progressive muscle wasting and weakness. A particular subset of muscular dystrophies is caused by mutations in genes coding for constituents of the dystrophin-associated glycoprotein complex (DGC). Mutations in the dystrophin gene

cause Duchenne muscular dystrophy, whereas mutations in sarcoglycan genes are responsible for Limb-Girdle Muscular Dystrophies.

The MDX mouse is a mouse model for Duchenne muscular dystrophy, and beta-sarcoglycan-deficient (BSG) and gamma-sarcoglycan-deficient (GSG) mice are mouse models for Limb-Girdle Muscular Dystrophies 2E and 2C. Expression profiles were generated from two individual mice (biological replicates) at eight different points between 1 and 20 weeks. There were technical replicates in the experiment, providing data from 4 time-series (biological and technical repeats) for each mouse model. The dataset also includes expression profiles for a healthy (wild-type) mouse model, giving four different classes (three types of MD and one wild-type). More details can be found in [Turk *et al.*, 2005]. A set of 812 candidate TFs was selected.

Human MD data

The second MD dataset contains expression profiles of in vitro muscle differentiation for six human individuals: three healthy and three Duchenne MD patients. Each time series consists of 7 measurements taken between 1 and 14 days. Therefore the dataset contains three expression profiles in each of the two classes (healthy and Duchenne MD). More details can be found in [Sterrenburg *et al.*, 2006]. A set of 296 candidate TFs was selected.

3.2 Experiments

For the MD dataset, in order to keep initial experiments of a small size, for each dataset we selected the top 50 most significantly differentially expressed target genes using a temporal Hotelling T^2 test algorithm [Vinciotti *et al.*, 2006]. For the E. coli dataset the set of targets consisted of the 14 most significantly differentially expressed genes (as shown in Figure 5).

We performed an exhaustive search over four structure types: a single regulator structure, a single regulator with a class node, two regulators, and two regulators with a class node (all shown in Figure 4). All possible combinations of regulators were considered, selected from the set of candidate TFs.

The list of candidate TFs is supplied by biologists. It should be pointed out that the set of candidates may not be exhaustive. In particular some of the targets may be unknown TFs. This could potentially affect the directionality of arcs between genes. For example if a target is actually a TF, it may actually be the parent of another TF rather than its descendant.

3.3 Validation

Obtaining biological validation of the results is difficult. Most often, TF-target gene interactions are unknown, and expensive to establish through laboratory experiments - one reason why this research is important. This is especially true for our MD datasets. Sometimes limited supporting data, such as TF binding location data, is available to back up findings. For the E. coli dataset we have some confirmed TF-target interactions to compare our findings with. However this list of interactions is by no means exhaustive.

Therefore, as an indicator of performance we have used prediction of unseen gene expression measurements. Using the BN we can predict the value of target genes (i.e. its discrete expression state) based on observed values of the TF(s) and class nodes where appropriate.

To ensure reliable and consistent results we used k-fold cross validation (with $k = 3$ or 4 depending on the number of samples and class distribution of each dataset) where the BN conditional probability distributions were learnt based on the whole dataset minus the kth fold, and prediction was performed on the unseen kth fold. The target prediction accuracy is the percentage of correct predictions over the unseen data fold, averaged over all k folds.

For each target gene, we compare the different network structures using the average target predictive accuracy for the highest scoring (in BIC) network for each target gene.

Name	Best structure	Top TF	Top acc	lexA rank	lexA acc
umuC	1P class	soxS	0.81	2	0.94
umuD	1P class	metJ	1.00	5	0.50
ruvA	1P class	phoP	0.75	9	0.38
ruvB	2P class	lexA,gcvR	0.81	1	0.81
recA	1P class	mall	0.88	7	0.88
recN	1P class	lexA	0.94	1	0.94
dinF	2P	gutM,caiP	0.63	4	0.75
dinI	2P class	lexA,gcvR	0.81	1	0.81
uvrB	2P class	lexA,gcvR	0.81	1	0.81
yebG	1P class	mall	0.88	7	0.88
yijW	2P	malT,melR	1.00	15	0.75
sbmC	1P class	uidR	0.75	8	0.88
sulA	2P	uidR, rob1	0.88	5	0.38
lexA	1P class	lexA	0.94	1	0.94
Mean	-	-	0.85	4.8	0.76
St. dev	-	-	0.10	4.1	0.20

Table 1: E. coli results: for each target gene, the highest scoring TF(s), their prediction accuracy and the ranking and prediction accuracy of true TF lexA is provided. Structure descriptions refer to the number of parent TFs (1P, 2P) and whether a class node was used.

4 Results

4.1 E. coli data

Table 1 lists the highest scoring regulator(s), together with the network structure type and its prediction accuracy for each of the 14 target genes.

LexA is found to be the highest scoring TF for 5 out of 14 regulators (sometimes paired with another TF) with target prediction accuracy always over 0.8. We can also see that LexA scores highly for each target - within in the top 10% of candidate TFs 99% of

Name	Best structure	Highest-scoring TF	Target accuracy
MTUS1	1P	BACH1	0.81
STX3A	1P	SET	0.79
ACAA2	1P	BACH1	0.71
PIG8	1P class	HDAC7A	0.83
IMPA2	1P class	ELF4	0.67

Table 2: Human MD - target genes modelled best with 1 parent TF

Name	Best structure	Highest-scoring TF	Target accuracy
FUNDC2	2P	SET,FOXJ3	0.81
EIF3S10	2P	SET,ZNF406	0.76
FGG	2P	PHF10,NFIB	0.71
LIPA	2P class	PDCD2,RBBP6	0.64
LRRFIP2	2P	NACA, PHF16	0.67

Table 3: Human MD - target genes modelled best with 2 parent TFs

Name	Best structure	Highest-scoring TF	Target accuracy
Dlk1	1P class	Klf4	0.93
Plagl1	1P class	Phf1	0.83
Fpr-rs3	1P class	Scand1	0.80
Capn6	1P class	Lhx5	0.78
Abca3	1P class	Nfatc1	1.00
Catna1	1P class	Myf6	0.93

Table 4: Mouse MD results - most genes were best modelled with 1 parent TF and a class node

the time (the exception being gene yijW). Table 1 also shows the rank and accuracy of the highest scoring TF combination involving lexA for each target. For example, for the target umuC, lexA is the second highest scoring (BIC) TF after soxS, and obtains a higher prediction accuracy of 0.94. Similarly for the target dinF, the TF combination lexA, nhaR ranks fourth behind highest scoring combination gutM,caiP and has a higher target prediction accuracy of 0.75.

The high scoring TFs that have been found in addition to LexA may be unknown gene relationships of biological interest. For example, the pairing lexA and gcvR is selected as the highest scoring TF combination for three target genes. This could be an indication that gcvR plays an important role in this E. coli network module.

4.2 Human MD data

Tables 2 and 3 show the target prediction accuracy for the highest scoring TF(s) for a selection of targets. We find that in many cases a particular network structure performs much better than others, indicating that a particular gene may be regulated by one or by two TFs, and/or that its regulatory program depends on class. For example, target gene FGG performs best with 2 regulators, obtaining prediction accuracy of 71% in this case, whilst PIG8 is best described with 1 TF, obtaining 83% prediction accuracy.

Moreover, the results show that the most suitable network for a particular target with one TF can be improved by the addition of another TF or class node. For example, the highest scoring 2-parent TF combination for target FGG is NFIB and PHF10, obtaining 71% predictive accuracy. The highest scoring 1-parent network is the TF PHF10, obtaining only 48% accuracy. A similar effect occurs with other genes.

We find that the majority of genes - 68% - are best described with a simple 1-TF network. The remainder require a more complex 2-TF network.

The addition of a class node seems more successful with some genes than others. This is an interesting point as the target genes selected are the most differentially expressed between healthy and diseased data samples. Therefore we might expect a class node to be important. However some targets, for example FGG and EIF3S10, appear more suited without a class node - but the highest scoring TFs in these cases are significantly differentially expressed themselves, making a class node more redundant. For targets where a class node is important (e.g. LIPA and IMPA2) the highest scoring TFs are not as differentially expressed.

Some targets did not obtain good prediction accuracy under any network structure due to of a lack of quality data for these genes - 10% of target genes have no network structure obtaining greater than 50% prediction accuracy (where random is 33%).

4.3 Mouse MD data

The results from the mouse data do not show such a broad variation in network structure across the target genes as the human data. Table 4 shows that in all cases only one parent TF is necessary. Whilst in some cases, 2 parent TFs provides good accuracy, invariably one parent TF does even better. For example for gene Dlk1, 2 parent TFs and a class node obtains a good 74% accuracy, but just one parent TF and a class node obtains an excellent 93% prediction rate. Similarly to our findings from the human data results, when a class node is not necessary, the highest scoring TF is itself significantly differentially expressed between classes.

4.4 Results summary

We applied the algorithm to E. coli gene expression data, in order to validate our method by comparison to a known regulatory network motif. Our results were impressive - we were able to identify the known repressor LexA as the highest scoring gene just over a third of the time and LexA was in the top 10% of candidate TFs 99% of the time. We also applied the algorithm to more complex muscular dystrophy data, where there is less knowledge about known regulator-target interactions.

Over all datasets, our results showed that the most suitable network structure depends on the target gene. Biologically, the fact that some targets are better controlled with 2 regulators and some with a single reg-

ulator makes sense as every gene is controlled by a different regulation program, which may consist of 1, 2 or more TFs.

The results also show that the majority of genes are best described with a simple 1-TF network, indicating that it is only a smaller subset of genes that require more complex co-regulation models. In particular this is the case with the E. coli and human MD datasets, although all mouse genes were best described with only one TF.

For some genes there was not enough data to obtain good prediction accuracy under any network structure indicating that in some cases there is a lack of quality data necessary to make reliable predictions and assertions about the regulatory network structure.

5 Conclusions and further work

In this paper we have investigated different Bayesian network structures to model the regulatory program of a set of given target genes. We have applied the algorithm to an E. coli and two muscular dystrophy gene expression datasets. Our results indicate that the most suitable network model (single or multiple regulators; incorporation of disease type) varies with each individual gene - a conclusion that makes sense biologically. We also find that the majority of target genes are regulated by only one TF. It is only a small subset of genes that require a more complex model.

In this paper we have only considered regulation models of up to 2 TFs. In reality a gene maybe controlled by a complex program of many TFs. Though algorithm complexity is also a factor, the small size of many gene expression datasets prevents reliable results for > 2 TF networks. Indeed for some genes in our experiments there was not enough data to obtain good prediction accuracy under any network structure.

In future work we intend to integrate other types of data (such as multiple microarray gene expression datasets, location binding data, text mining analysis on biology literature and gene ontology information) into our algorithm to alleviate these data issues and improve reliability and robustness of results. We also plan to investigate incorporating temporal information, which is a key aspect of gene regulation - for example in feedback loops, through Temporal Bayesian networks and Dynamic Bayesian networks.

References

[Friedman *et al.*, 2000] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[Hartemink *et al.*, 2002] A.J. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems*, 17(2):37–43, 2002.

[Khanin *et al.*, 2006] R. Khanin, V. Vinciotti, and E. Wit. Reconstructing repressor protein levels from expression of gene targets in escherichia coli. *PNAS*, 103(49):18592–18596, 2006.

[Pearl, 1991] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, San Francisco, CA, USA, 1991.

[Pe'er *et al.*, 2006] D. Pe'er, A. Tanay, and A. Regev. Minreg: A scalable algorithm for learning parsimonious networks in yeast and mammals. *J. Machine Learning Res*, 7:167–189, 2006.

[Salgado *et al.*, 2006] H. Salgado, S. Gama-Castro, M. Peralta-Gil, E. Diaz-Peredo, F. Sanchez-Solano, A. Santos-Zavaleta, I. Martinez-Flores, V. Jimenez-Jacinto, C. Bonavides-Martinez, J. Segura-Salazar, A. Martinez-Antonio, and J. Collado-Vides. Regulondb (version 5.0): Escherichia coli k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, 34, 2006.

[Segal *et al.*, 2003] E. Segal, M. Shapira, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):168–176, 2003.

[Sterrenburg *et al.*, 2006] E. Sterrenburg, C. van der Wees, S.J. White, R. Turk, R. de Menezes, G.J.B. van Ommen, J.T. den Dunnen, and P.A.C. t Hoen. Gene expression profiling highlights defective myogenesis in dmd patients and a possible role for bone morphogenetic protein 4. *Neurobiology of Disease*, 23(1):228–236, 2006.

[Tucker *et al.*, 2005] A. Tucker, V. Vinciotti, P.A.C. 't Hoen, and X. Liu. Bayesian network classifiers for time-series microarray data. In *Advances in Intelligent Data Analysis VI (LNCS)*, volume 3646, pages 475–485. Springer-Berlin/Heidelberg, 2005.

[Turk *et al.*, 2005] R. Turk, E. Sterrenburg, E.J. de Meijer, G.J.B. van Ommen, J.T. den Dunnen, and P.A.C. 't Hoen. Muscle regeneration in dystrophin-deficient mdx mice studied by gene expression profiling. *BMC Genomics*, 6(98), 2005.

[Vinciotti *et al.*, 2006] V. Vinciotti, X. Liu, R. Turk, E.J. de Meijer, and P.A.C. 't Hoen. Exploiting the full power of temporal gene expression profiling through a new statistical test: Application to the analysis of muscular dystrophy data. *BMC Bioinformatics*, 7(183), 2006.

[Yeang and Jaakkola, 2006] C.H. Yeang and T. Jaakkola. Modeling the combinatorial functions of multiple transcription factors. *Journal of Computational Biology*, 13(2):463–480, 2006.

Tensor Decompositions for Probabilistic Classification

Marcel van Gerven
Department of Knowledge and Information Systems
Radboud University Nijmegen
Toernooiveld 1, 6525 ED, Nijmegen, The Netherlands
marcelge@cs.ru.nl

Abstract

Tensor decompositions are introduced as a novel approach to probabilistic classification. The approach is validated by means of a clinical database consisting of data about 1002 patients that suffer from hepatic disease. The approach performs comparably to results that have been obtained using a naive Bayes classifier.

1 Introduction

We present a novel probabilistic classification technique which is based on the decomposition of a multiway array, also known as a *tensor*. Components of the decomposition are given by a set of vectors that allow for a compact representation of the original tensor. We call classifiers that use this technique *decomposed tensor classifiers*, and test their performance by means of a database that contains data about 1002 patients that present with hepatic disease. Classification performance is analyzed and compared with that of the naive Bayes classifier [1].

2 Tensors and their Decompositions

A tensor is a concept taken from multilinear algebra which generalizes the concepts of vectors and matrices.

Definition 1. *Let $I_1, \ldots, I_N \in \mathbb{N}$ denote index upper bounds. A* tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *is an N-way array where elements $a_{i_1 \cdots i_n}$ are indexed by $i_j \in \{1, \ldots, I_j\}$ for $1 \leq j \leq N$.*

We call N the *order* of a tensor, such that a tensor of order one denotes a vector $\mathbf{a} \in \mathbb{R}^{I_1}$, and a tensor of order two denotes a matrix $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$. The *outer product* $\mathcal{A} \circ \mathcal{B}$ of two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_n}$ is defined as the tensor $\mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times J_1 \times \cdots \times J_n}$ such that $c_{i_1 \cdots i_n j_1 \cdots j_n} = a_{i_1 \cdots i_n} \cdot b_{j_1 \cdots j_n}$ for all elements of $\mathcal{C}$. The rank of a tensor is then defined as follows.

Definition 2. *A tensor of order N has* rank *one if it can be written as an outer product $\mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)}$ of vectors. The rank of a tensor $\mathcal{A}$ is defined as the minimal number of tensors $\mathcal{A}_1, \ldots, \mathcal{A}_K$ of rank one such that $\mathcal{A} = \sum_{k=1}^{K} \mathcal{A}_k$.*

One way of finding a rank-1 approximation of a tensor $\mathcal{A}$ is by means of the *higher-order power method* (HOPM) as described in Ref [2]. The method finds a tensor $\hat{\mathcal{A}} = \lambda \cdot \mathbf{b}^{(1)} \circ \cdots \circ \mathbf{b}^{(N)}$, with scalar λ and unit-norm vectors $\mathbf{b}^{(n)}$, $1 \leq n \leq N$, that minimizes a least-squares cost function. A greedy approach to finding a sum of rank-1 terms is to apply the higher-order power method to the residuals that remain after obtaining a rank-1 approximation. By defining $\mathcal{A}^1 \equiv \mathcal{A}$ and $\mathcal{A}^k \equiv \mathcal{A}^{k-1} - \text{HOPM}(\mathcal{A}^{k-1})$ the following rank-K approximation of a tensor $\mathcal{A}$ is obtained:

$$R_K(\mathcal{A}) \equiv \sum_{k=1}^{K} \text{HOPM}(\mathcal{A}^k)\,. \tag{1}$$

For our classification purposes we start HOPM from one random initialization since we have observed no significant differences when using more elaborate schemes. The algorithm has converged when the increase in fit between the tensor and its approximation that is gained after one iteration drops below a small error criterion ϵ.

3 Classification with Tensor Decompositions

We focus on a multiset $\mathbf{A} = \{\mathbf{a}^1, \ldots, \mathbf{a}^n\}$ that represents our data, where an instance $\mathbf{a}^i = (x_1^i, \ldots, x_N^i)$ consists of evidence $(x_1^i, \ldots, x_{N-1}^i)$ and a class label x_N^i. We assume that all variables are discrete and use I_j with $1 \leq j \leq N$ to denote the finite number of values x_j of a variable X_j. The basic idea is to obtain an approximation of a *incomplete* tensor $\mathcal{A}$ using a tensor decomposition. Let $\mathbf{x}$ denote the evidence and let $n(\mathbf{x}, x_N)$ stand for the number of times $(\mathbf{x}, x_N)$ occurs in $\mathbf{A}$. We transform $\mathbf{A}$ into an incompletely specified tensor $\mathcal{A} \in [0,1]^{I_1 \times \cdots \times I_N}$, such that

$$a_{x_1 \cdots x_N} = \frac{1}{n} n(\mathbf{x}, x_N) \tag{2}$$

for all $(\mathbf{x}, x_N)$ for which some $(\mathbf{x}, j)$ with $1 \leq j \leq I_N$ occurs in $\mathbf{A}$. Hence, $a_{x_1 \cdots x_N}$ is undefined for unseen evidence $\mathbf{x}$, which implies that the tensor is incomplete. The element $a_{x_1 \cdots x_N}$ is used to represent an estimate of the joint probability $p(\mathbf{x}, x_N)$. We may use a sparse representation of tensors $\mathcal{A} \in [0,1]^{I_1 \times \cdots \times I_N}$, where N may be large, provided that only some of the elements are defined. In case of probabilistic classification, our interest is in computing $p(x_N \mid \mathbf{x})$ based on our estimate of $p(\mathbf{x}, x_N)$. Although $p(\mathbf{x}, x_N)$ is approximated by $R_K(\mathcal{A})_{x_1 \cdots x_N}$, we have no guarantee that the tensor approximation represents a proper probability distribution for unseen evidence (which is the goal of probabilistic classification), since the approximation may be unnormalized or even lying outside the unit

interval. Therefore, we use the following transform when computing the conditional probability of X_N given $\mathbf{x}$:

$$p(x_N \mid \mathbf{x}) = \frac{R_K^+(\mathcal{A})_{x_1 \cdots x_N}}{\sum_{1 \leq j \leq I_N} R_K^+(\mathcal{A})_{x_1 \cdots x_{N-1} j}} \qquad (3)$$

where $R_K^+(\mathcal{A})_{x_1 \cdots x_N}$ is defined as

$$R_K(\mathcal{A})_{x_1 \cdots x_N} - \min \left\{ 0, \min_j (R_K(\mathcal{A})_{x_1 \cdots x_{N-1}, j}) \right\},$$

which ensures that we sum over positive terms by making (small) negative terms non-negative. We use the term *decomposed tensor classifier* (DTC) to denote a classifier that uses the approximation $R_K(\mathcal{A})_{x_1 \cdots x_N}$ for the purpose of classification, as shown in Algorithm 1.

Algorithm 1 Decomposed tensor classification.

input: $\mathbf{A}_{\text{train}}$, $\mathbf{A}_{\text{test}}$, K
transform the dataset $\mathbf{A}_{\text{train}}$ into the tensor $\mathcal{A}_{\text{train}}$ using (2)
learn the approximation $R_K(\mathcal{A}_{\text{train}})$ using (1)
for all rows $(\mathbf{x}) \in \mathbf{A}_{\text{test}}$ **do**
  for $j = 1$ **to** I_N **do**
    compute $p(j \mid \mathbf{x})$ using (3)
  end for
  assign class label $\mathcal{L}(\mathbf{x}) = \arg\max_j \{p(j \mid \mathbf{x})\}$
end for
return class labels $\mathcal{L}$

In order to examine the performance of decomposed tensor classifiers, we have made use of the COMIK dataset, which was collected by the Copenhagen Computer Icterus (COMIK) group and consists of data on 1002 jaundiced patients that may be classified into one of four diagnostic categories[3]. As a preprocessing step, we have computed the mutual information between evidence variables and the class variable, and selected the eighteen evidence variables that show highest mutual information (MI) with the class variable as the basis for classification. Classification performance of the decomposed tensor classifiers is compared with that of a naive Bayes classifier using a ten-fold cross-validation scheme. Since the COMIK dataset contains missing values, and the decomposed tensor classifiers require complete data, we have used multiple imputation to create three complete datasets from the incomplete dataset. Since we have no knowledge about the missing data mechanism, we make the (admittedly unrealistic) assumption that data is missing completely at random, and use the prior probabilities of the evidence variables to determine the imputed values. This allows a comparison in terms of classification performance between the naive Bayes classifier and the DTC, where performance is defined as the percentage of correctly classified cases and averaged over the ten folds and over the three complete datasets.

The comparison of the classification accuracy of the decomposed tensor classifier with that of the naive Bayes classifier is shown in Fig. 1. The highest average accuracy for the decomposed tensor classifier is reached at nineteen components with an accuracy of 76.75%, whereas for the naive Bayes classifier, the average classification accuracy is 77.25%. Although the naive Bayes classifier and the decomposed tensor classifier operate differently, they perform comparably with respect to classification accuracy. If

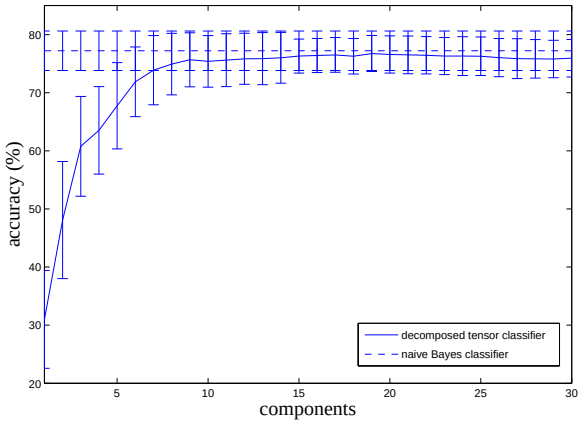


Figure 1: Average classification accuracy.

we inspect the classifications that were made by the classifiers then it is interesting to see that only 254 out of a total of 2955 cases (8.60%) have been classified differently by the two classifiers. Out of these 254 cases, the naive Bayes classifier assigned 107 cases to the correct class, whereas the decomposed tensor classifier assigned 93 cases to the correct class. Hence, the classifiers are able to classify different cases correctly, suggesting that there are certain problems for which the naive Bayes classifier is more suitable, and other problems for which the decomposed tensor classifier is more suitable.

4 Conclusion

We have shown that tensor decompositions can be used for the purpose of probabilistic classification. The classification performance of the DTC on a problem in medical diagnosis is comparable to that of the naive Bayes classifier. The different mode of operation, together with the results concerning correctly classified cases, suggest that there may be particular problems for which this new technique outperforms the naive Bayes classifier. Current limitations are the requirements that data is discrete and complete, and the fact that learning the classifiers requires more computational resources than the (easy to learn) naive Bayes classifier. This work has demonstrated the potential of this new classification technique. At present, practical usefulness of the DTC requires further validation and a more complete understanding of the conditions under which the technique may be appropriate.

References

[1] Maron, M.E.: Automatic indexing: An experimental inquiry. Journal of the ACM **8**(3) (1961) 404–417

[2] de Lathauwer, L., de Moor, B., Vandewalle, J.: On the best rank-1 and rank-($R_1, R_2, \ldots, R_N$) approximation of higher-order tensors. SIAM J Matrix Anal Appl **21**(4) (2000) 1324–1342

[3] Malchow-Møller, A., Thomson, C., Matzen, P., Mindeholm, L., Bjerregaard, B., Bryant, S., Hilden, J., Holst-Christensen, J., Johansen, T.S., Juhl, E.: Computer diagnosis in jaundice: Bayes' rule founded on 1002 consecutive cases. J Hepatol **3** (1986) 154–163

Paper session:

Feature Selection / Reduction and Visualisation

Improving the Performance of Consensus Clustering Through Seeding: An Application to Visual Field Data

Stephen Swift, Allan Tucker, Michael Hirsch
Brunel University, West London, UXB 3PH, UK
{stephen.swift, allan.tucker, michael.hirsch}@brunel.ac.uk

Abstract

The data clustering of Visual Field test points is an important pre-processing step in the modelling of a number of ophthalmic conditions, e.g. glaucoma and ocular hypertension. However, there has been a recent increase in the advent of new clustering methods that claim superior performance over classical methods. Thus, it has become increasingly difficult to choose the right kind of method for many applications. This has led to the development of ensemble clustering methods such as Robust and Consensus Clustering. In this paper we compare the scalability of Consensus Clustering to a modified version which is seeded with the results of Robust Clustering. We show that the search space can be reduced significantly using our technique, producing better results in a shorter execution time. We demonstrate our methods on an ophthalmic Visual Field dataset and synthetic data.

1 Introduction

The data clustering of Visual Field data test points is an important pre-processing step in the modelling of a number of ophthalmic conditions. Often the number of observations per patient is less than the number of Visual Field points. This can lead to many models encountering problems, e.g. the Vector Autoregessive process [Swift and Liu, 2002]. Clustering the points into small highly related subsets can often avoid this problem. However, recently it has become harder for a practitioner to choose the correct method due to the explosion in the number of clustering methods available. Ensemble clustering methods, which combine a number of clustering results into a single set of clusters, can be used if there is no obvious choice of clustering method.

There are many practical applications involving the partitioning of a set of objects into a number of mutually exclusive subsets, which is a known NP hard problem. Research associated with achieving this based on distance metrics or correlations is known as clustering. Any algorithm which applies a global search for the optimal clusters in a given data set will run in exponential time to the size of the problem space, and so a heuristic or approximate procedure is normally required to cope with most real world problems.

Many different heuristic algorithms are available for clustering, perhaps the most common being K-Means and hierarchical clustering [McQueen, 1967; Ward 1963]. Most algorithms make use of a starting allocation of variables, for example, based upon random points in the data space or upon the most correlated variables and therefore contain bias in their search. They are also prone to becoming stuck in local maxima during the search. There has also been research into the use of artificial intelligence techniques such as genetic algorithms, neural networks and simulated annealing to solve the grouping problem resulting in a more general partitioning method that can be applied to clustering [Falkenauer, 1998; Kohonen, 1989; Lukashin and Fuchs, 2001]. These methods aim to overcome the biases and local maxima involved with heuristic searches but require the fine-tuning of parameters.

Due to the high degree of variation between clustering methods, we have previously developed techniques for combining the results of these methods to produce more reliable clusters. In particular, we apply two algorithms for generating robust [Kellam *et al.*, 2001] and consensus clusters [Swift *et al.*, 2004] in the context of ophthalmic (Visual Field) analysis. Similar work can be seen in protein secondary structure prediction, where methods fail to completely agree, consensus algorithms are employed [Cuff *et al.*, 1998]. These can either report only full agreements, or the majority of agreements. Additionally, in [Monti *et al.*, 2003] a Consensus Clustering type technique was developed for testing the stability of clustering methods applied to gene expression data. This method differs from ours since the inputs to the consensus method are the results of running a single algorithm on datasets that are perturbations of the original. Strehl *et al.* investigated the use of ensemble methods to combine a set of partitions [Strehl and Ghosh, 2003]. Yeung *et al.* have compared a number of clustering methods based upon a figure of merit metric which rates the predictive power of a clustering arrangement [Yeung *et al.*, 2001]. In this paper we compare the similarity of two clustering arrangements using the Weighted-Kappa metric [Altman, 1997].

Within this paper we present an improved version of the Consensus Clustering algorithm, which is seeded with the results of Robust Clustering. We carry out extensive analysis on our improved Consensus Clustering method, along with other well-documented clustering methods, on both ophthalmic (Visual Field) and synthetic data.

This paper is organised as follows: Section 2 details the methods we use in this paper, along with notation and

comparison metrics used; Section 3 describes the datasets that are used in this paper, along with a description of the experiments carried out; Section 4 details the results of all of the experiments, and discusses their implications; finally, Section 5 draws some conclusions.

2 Methods

2.1 Notation

Let $G = \{g_1, \ldots, g_m\}$ be a partition of the variables $\{x_1, \ldots, x_{n.}\}$. The union of all the g_i is $\{x_1, \ldots, x_{n.}\}$ and $g_i \cap g_j = \phi$, $1 \le i \ne j \le m$ where g_{ij} is the ith element of the jth cluster/group and s_i is size of each cluster/group g_i of G.

2.2 Clustering Techniques

Both Robust Clustering and Consensus Clustering require a number of clustering results as input to them. The methods we use within this paper are described within Table 1.

Method	Abbreviation	Distance Metric
PAM	PAM	Correlation
K-Means	KME	Euclidean
Hierarchical (Average)	HAV	Correlation
Hierarchical (Complete)	HCO	Correlation
Hierarchical (Single)	HSI	Correlation
Hierarchical (Ward)	HWA	Correlation
Hierarchical (McQuitty)	HMC	Correlation
Hierarchical (Median)	HME	Correlation
Hierarchical (Centroid)	HCE	Correlation
Model Based Clustering	MBC	N/A

Table 1. Clustering Methods.

A description of PAM can be found in [Kaufman and Rousseeuw, 1987] and model based clustering in [Fraley and Raftery, 2002].

Input:	*List*: all pairs (i,j) from the agreement matrix such that all corresponding A_{ij} contain the maximum possible agreement
1)	The i and j from the first element of *List* form the first robust cluster
2)	For all of the remaining i,j pairs in *List*, i and j are searched for within the current robust clusters
3)	If i is found (in robust cluster x) and j is not found then j isadded to robust cluster x
4)	If j is found (in robust cluster y) and i is not found then add i to robust cluster y
5)	If i is found in robust cluster x and j is found in robust cluster y then merge x and y
6)	If i and j are not found then place i and j into a new robust cluster
7)	End For
Output:	Set of Robust Clusters

Table 2. The Robust Clustering Algorithm.

2.3 Robust Clustering

The Robust Clustering (RC) algorithm [Kellam *et al*., 2001] is designed to take numerous sets of clusters as input and uses these to generate a set of *robust clusters*. A robust cluster must only consist of objects that appear together in *all* the input clusters produced from different clustering algorithms. Firstly, an upper triangular $n \times n$ agreement matrix (called A) is generated with each cell containing the number of agreements amongst methods for clustering together the two variables, represented by the indexing row and column indices. This matrix is then used to cluster variables based upon their cluster agreement (as found in the matrix). The algorithm works by taking in the agreement matrix in order to generate a list, *List*, which contains all the pairs (i,j) where the appropriate cell in the agreement matrix contains a value equal to the number of methods being combined (i.e. full agreement). This method is described in Table 2. Note that RC does not always assign all of the variables to robust clusters.

2.4 Consensus Clustering

The RC algorithm is subject to discarding variables if only one clustering method fails to agree with the other methods. In fact, if no two variables have maximum agreement then no robust clusters will be found. Therefore, in order to generate clusters with high agreement across methods but not so restrictive as to discard otherwise consistent variables as happens with the RC algorithm, we have used an algorithm for generating consensus clusters. Consensus Clustering (CC) was introduced to analyse gene expression data in [Swift *et al*., 2004] and makes use of the agreement matrix, A. However, rather than grouping variables based only on full agreement, it attempts to maximise a metric which rewards variables in the same cluster if they have high agreement and penalises variables in the same cluster if they have low agreement. In particular, the algorithm tries to maximise agreement over all clusters (the best arrangement will maximise Equation 1) using the function in Equation 2 to score each cluster, which we will refer to as the *Fitness* of a candidate CC arrangement. Within these equations, β is a user-defined parameter (the agreement threshold), which determines whether the score for the cluster is incremented or decremented.

Definition 1. *Max*(A) is defined, where A is an agreement matrix, as the largest value in the upper triangle of A.

Definition 2. *Min*(A) is defined, where A is an agreement matrix, as the smallest value in the upper triangle of A.

$$f(G) = \sum_{i=1}^{m} H(g_i) \tag{1}$$

$$H(g_i) = \begin{cases} \sum_{j=1}^{s_i-1} \sum_{k=j+1}^{s_i} (A_{g_{ij}g_{ik}} - \beta) & , s_i > 1 \\ 0 & , \text{otherwise} \end{cases} \tag{2}$$

If β is less than or equal to *Min*(A), then Equation 1 is maximised when all variables are placed into the same cluster. Alternatively, when β is greater than or equal to *Max*(A), then Equation 1 is maximised when each variable is placed into its own cluster. The following two proofs have been adapted from [Tucker *et al*., 2001].

Proposition 1. If $\beta \ge Max(A)$, then a maximum for $f(G)$ will occur when each variable is placed into their own cluster.

Proof. From Equation 2: if $\beta \geq Max(A)$ then $A_{g_{ij}g_{ik}} - \beta \leq 0$. Therefore the maximum value for $H(g_i)$ will be zero, which will occur when each $s_i = 1$. If each $s_i = 1$ then all variables will be in their own cluster.■

Proposition 2. If $\beta \leq Min(A)$, then $f(G)$ is maximised when each variable is placed into a single cluster.
Proof. If $\beta \leq Min(A)$, then it holds that $A_{ij}-\beta \geq 0$ for all i, j (in the upper triangle). Therefore $f(G)$ is greatest, if the sum of all of the $H(g_i)$ involve all of the elements of the upper triangle of A, this will occur only if all variables belong to one cluster. For a more complete version of these proofs see [Tucker *et al*., 2001].■

A sensible value for β should be chosen that lies between the minimum and the maximum agreement, i.e. $Min(A) \leq \beta \leq Max(A)$. Essentially all clusters produced by CC are scored by $f(G)$, rewarding and preserving clusters with high agreement between members, whilst penalising and discarding clusters containing low agreement between members. A value for β should lie between the minimum and the maximum agreement so as not to skew the scoring function. For a uniformly and/or symmetrically distributed agreement matrix, $(Max + Min)/2$ is the mean value, therefore we penalise values below the mean agreement and reward above it. A search is needed to implement CC. There are many methods for performing a search. In [Swift *et al*., 2004] it was found that Simulated Annealing [Kirkpatrick *et al*., 1983] performed best because it is an efficient search/optimisation procedure that does not suffer from getting stuck in local minimums. Table 3 describes the consensus clustering algorithm; a way of determining suitable parameter values is suggested in [Swift *et al*., 2004].

Input:	Agreement Matrix ($n \times n$), A; Number of Iterations, *Iter*; Agreement Threshold, β, Initial Temperature, θ_0; Cooling Rate, c
1)	Generate a random number of empty clusters (≤n)
2)	Randomly distribute the variables 1..n between the clusters
3)	Score each cluster according to Equation 1
4)	For I = 1 to *Iter* do
5)	Either Split a cluster, Merge two clusters or Move a variable from one random cluster to another
6)	Set Δf to difference in score according to Equation 1
7)	If $\Delta f < 0$ Then
8)	Calculate probability, p, according to Equation 3
9)	If p > random(0,1) then undo operator
10)	End If
11)	$\theta_i = c\ \theta_i$-1
12)	End For
Output:	Set of Consensus Clusters

Table 3. The Consensus Clustering Algorithm.

Note that random(0,1) (line 9) returns a random uniformly distributed real number between 0 and 1. The probability, p (line 8), is calculated by:

$$p = \Pr(\text{accept new}) = e^{-\Delta f}, \Delta f = \frac{f(old) - f(new)}{\theta_t} \quad (3)$$

2.5 Robust Cluster Seeded Consensus Clustering

It was noted in [Swift *et al*., 2004] that there was a significant overlap between the robust clusters and the corresponding consensus clusters. In fact, by definition any two variables that have full agreement should be contained within the same consensus cluster. Therefore, it makes sense to exploit the deterministic and efficient Robust Cluster search in order to reduce the search space when finding the Consensus Clustering results. This is achieved in our algorithm Robust Cluster Seeded Consensus Clustering (RCCC) by mapping all variables which appear in the same robust cluster to a single new variable which represents that Robust Clustering result. (see Figure 1 for a graphical description). This effectively reduces the search space by $\left(\sum_{i=1}^{m}|RC_i|\right) - |RC|$ where RC_i is the set of variables in the ith robust cluster and RC is the set of robust clusters. Consensus Clustering can then be applied as normal to the dataset using the mapped variables.

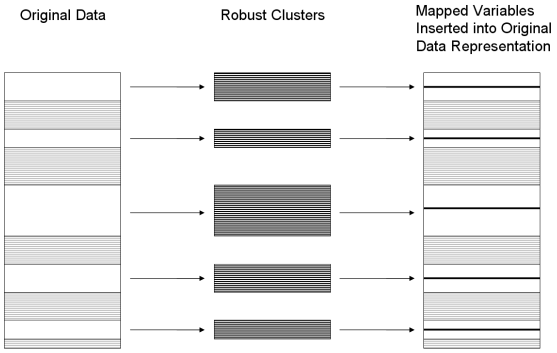


Figure 1. Mapping Robust Clusters to new variables to reduce dimensionality.

3 Datasets and Experiments

3.1 Datasets

Two datasets are used within this paper. Both datasets have an expected structure, i.e. there is a notion of the perfect clustering arrangement. This allows any clustering results to be evaluated for accuracy using a cluster comparison metric such as Weighted-Kappa [Altman, 1997].

The first dataset is the previously mentioned ophthalmic data which was the result of a study carried out in Australia called the Blue Mountain Eye Study [Healey and Mitchell, 2004], and will be referred to as the BMVF dataset. This study concerned the vision and prevalence of common eye diseases of an urban population, and was carried out between 1992 and 1994 on several thousand people. One of the sets of measurements taken during this study was based on Visual Field tests. To conduct visual field analysis the retina is divided into a set of 54 points, the level of sensitivity of the eyesight of a patient is tested at each point and is assigned a numerical value. A specialised machine is used to conduct these tests, which can take several hours for all the points of both eyes.

Research has been carried out which maps the distribution of nerve fibre bundles around the optic nerve head [Garway-Heath et al., 2000], and current theory states that any measurements in the same nerve fibre bundle sector should be highly related. Within this paper it is aimed to test this theory by applying a number of clustering methods to the visual field dataset and then obtaining a consensus set of clusters through the use of Consensus Clustering. The accuracy of the clustering methods and Consensus Clustering can be directly measured using the WK metric and the allocation of visual field points to nerve fibre bundle. The mapping of the distribution of the visual field points can be found in [Garway-Heath et al., 2000]. Previous results of clustering Visual Field data can be found in [Mandava *et al*., 1993] and [Spenceley *et al*., 1996].

The second dataset is a synthetic set generated from the multivariate normal distribution (MVN). The variables forming each cluster are generated randomly from an MVN distribution with the same mean vector and covariance matrix. This dataset was used in [Swift *et al*., 2004]. The dataset consists of predefined clusters ranging in size from 1 variable to 25 variables in size giving a total of 325 variables within the dataset, each of these predefined clusters will be referred to as *mvn*(1), *mvn*(2),…,*mvn*(25). In order to test the scalability of the methods presented in this paper this dataset is further divided to create a series of increasing dimensionality synthetic datasets. 16 datasets are created, the first consists of the concatenation of *mvn*(1) to *mvn*(10) creating a 55 variable MVN dataset, the second consists of *mvn*(1) to *mvn*(11) creating a 66 variable MVN dataset, etc..., and the sixteenth dataset consists of *mvn*(1) to *mvn*(25) creating a 325 variable dataset. It is expected that clustering methods should be able to separate the individual components of each of the MVN datasets, e.g. clustering the 55 variable MVN dataset back to *mvn*(1), *mvn*(2),..., *mvn*(10). Spurious relationships may exist between each MVN dataset, hence the results may not be perfect (i.e. achieving a Weighted-Kappa of 1.0, see section 3.2).

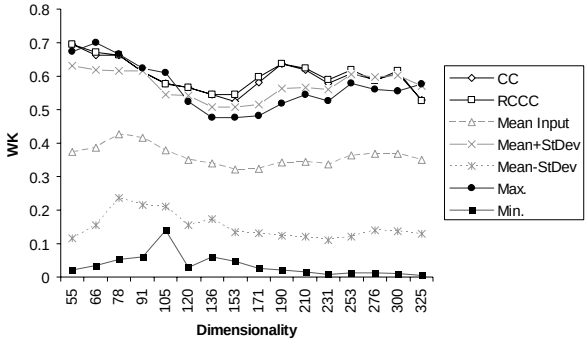


Figure 2. Weighted-Kappa of CC, RCCC, the input clustering methods and their mean, mean+standard deviation (Mean+StDev) and mean-standard deviation (Mean-StDev) for all the MVN datasets.

3.2 Experiments

The first set of experiments involved exploring the quality of existing clustering methods to CC and the novel RC Seeded CC, applying a number of existing clustering methods to the two datasets described above and comparing the resulting clusters with the "original" arrangements dictated by the MVN distributions and the BMVF array arrangements. This was performed using the Weighted-Kappa metric; the metric ranges from 1.0 (complete agreement) to -1.0 (total disagreement).

Having compared the quality of the clusters we then look at the efficiency of the RC seeded CC compared to the standard CC. This is done by exploring the convergence graphs for the two methods on the different datasets as well as the fitness and number of fitness function calls (FC) at the point of convergence.

The CC and RCCC methods are stochastic algorithms so they were run ten times on each dataset and the results averaged, to reduce the chance of getting "fluke" results.

4 Results

4.1 Weighted-Kappa Comparison

Figure 2 shows the WK results for the Consensus Clustering (CC) and the Robust Clustering Seeded Consensus Clustering (RCCC) along with the mean (+/- the standard deviation) of the input clustering methods for each MVN dataset. It can be seen that for smaller datasets (where *n* is less than 120) the best input clustering method does about the same as CC and RCCC. The consensus methods can be seen as a way to automatically select the best input method for smaller dimensionality datasets. However, for the larger datasets (where *n* is greater than 120) the consensus methods do considerably better than the best of the input methods. This is reflected in results from the BMVF dataset shown in Figure 3 where the consensus methods generally do better than the other input methods.

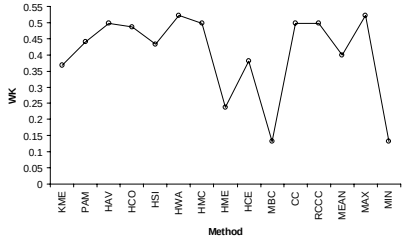


Figure 3. Weighted-Kappa of CC, RCCC, the input clustering methods and their mean, max and min for the BMVF data.

4.2 Convergence Analysis

We now turn to the convergence of CC compared to RCCC. Figure 4 shows the convergence graphs of the MVN data for n=55, 105, 171 and 325 and Figure 5 shows the graphs for the BMVF dataset. For nearly all of these experiments, and on the larger dimensionality ones in particular, there is a clear improvement in efficiency when seeding the Consensus Clustering algorithm with robust clusters. Throughout the learning curve, the fitness is higher for RCCC and, as Table 4 shows, the final fitness is higher and the number of function calls is lower at the point of convergence.

	Unseeded		Seeded	
n	**Fitness**	**FC**	**Fitness**	**FC**
55	464.0	4123918.4	464.0	4155970.4
66	628.0	3958294.3	628.0	4060154.2
78	866.0	4369621.7	866.0	4485488.0
91	998.0	4515860.1	998.0	4718021.4
105	1120.0	4643804.1	1120.0	4756861.0
120	1335.2	4747805.5	1336.0	5132994.3
136	1573.0	4601307.7	1573.0	4848421.0
153	1754.8	5262977.3	1792.7	5128405.2
171	2097.0	5052788.8	2119.0	5123576.6
190	2598.7	5089474.8	2599.0	5293983.5
210	3144.2	5100526.1	3153.0	5297003.0
231	3287.0	5969251.3	3344.6	5640023.2
253	4147.1	5400520.9	4211.9	5414867.9
276	4686.9	6048275.6	4700.9	5708973.8
300	5080.5	6002740.1	5090.3	5581987.5
325	5783.5	5768558.6	5805.9	5742477.3
54(BMVF)	1151.0	2932766.6	1151.0	2728614.5

Table 4. Number of Fitness calls (FC) and fitness at convergence.

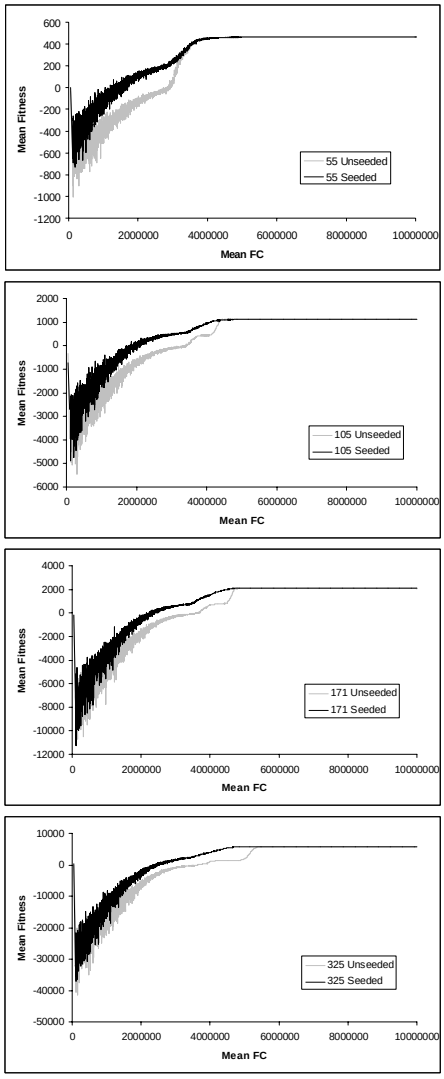


Figure 4. Convergence Graphs for CC and RC Seeded CC for MVN data with dimensionalities: 55, 105, 171 and 325.

Figure 6 shows plots of the scaling curves for the difference in fitness between standard CC and the proposed RCCC as dimensionality increases, as well as the change in percentage of variables which are assigned to robust clusters. The difference in fitness is the average between the seeded and unseeded fitness at each iteration. It appears that the maximum fitness will increase as dimensionality increases (see Equation 1). It seems as if the difference in fitness is growing with an exponential curve as *n* increases whilst the percentage of robust clusters decreases linearly with a very small gradient. The Pearson's correlation coefficients [Snedecor and Cochran, 1989] for both of these are 0.972. This suggests that for datasets with larger dimensionality RCCC should offer considerable savings in terms of efficiency resulting in a substantially improved final fitness.

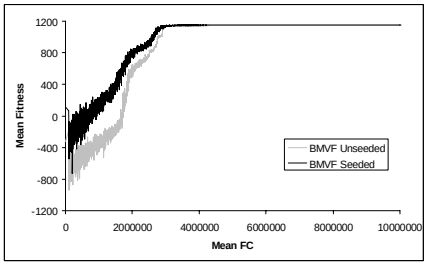


Figure 5. Convergence Graphs for CC and RC Seeded CC for the BMVF Data.

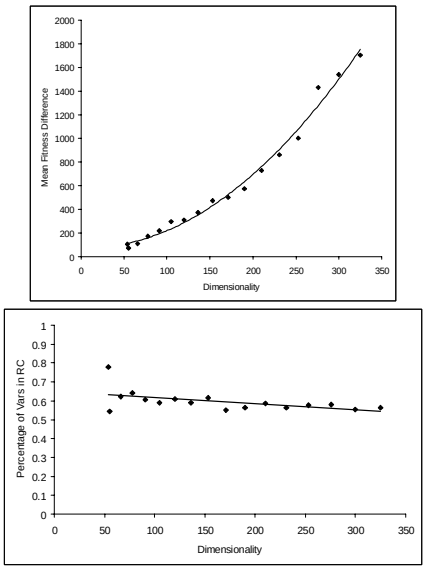


Figure 6. Scaling graphs of Difference in Fitness between CC and RCCC (top) and Percentage of Variables assigned to RCs (bottom) for increasing Dimensionality.

5 Conclusions

In this paper we have shown that we can use ensemble clustering methods to cluster Visual Field test points into highly related subsets, which agree with known anatomical knowledge regarding the physiology of the eye. We have shown that ensemble clustering methods can successfully combine the results of a diverse number of (potentially biased) clustering methods into a reliable consensus. The method presented in this paper is a useful preprocessing stage, prior to the mathematical modelling of Visual Field deterioration. Each consensus cluster can be modelled independently using techniques such as Bayes-

ian networks or the Vector Auto-regressive processes requiring considerably less parameters than modelling the entire Visual Field, thereby reducing the risk of overfitting data. We have shown that the final clustering arrangement found using our method is considerably better in terms of fitness calculated from the data and similarity to the original clusters found using the Weighted-Kappa metric. The results on the ophthalmologic BMVF dataset (and the MVN dataset) show that using Consensus Clustering can improve the accuracy of clustering, if it is unknown what the most appropriate method would be.

Future work will use the consensus clusters to model Visual Field deterioration, in order to predict the progression of conditions such as glaucoma. The seeded approach, introduced in this paper, improves the efficiency of the Consensus Clustering method, which will enable us to look at simultaneously clustering both patients and field points on considerably larger datasets.

6 Acknowledgements

We would like to thank Paul Healey and Paul Mitchell for making the Blue Mountain data-set available. We would also like to thank David Crabb and Haogang Zhu for preparing the Visual Field data.

7 References

[Altman, 1997] Altman D.G., *Practical Statistics for Medical Research,* Chapman and Hall, London 1997.

[Cuff *et al*., 1998] Cuff J.A., Clamp M.E., Siddiqui S.A., Finlay M. and Barton G.J., JPred: A consensus secondary structure prediction server, *Bioinformatics*, 14:892-893, 1998.

[Falkenauer, 1998] Falkenauer E., *Genetic Algorithms and Grouping Problems*, Wiley, 1998.

[Fraley and Raftery, 2002] Fraley C. and Raftery A.E. Model-based clustering, discriminant analysis, and density estimation, *Journal of the American Statistical Association*, 97:611-631, 2002.

[Garway-Heath *et al*., 2000] Garway-Heath D.F., Poinoosawmy D., Fitzke F. and Hitchings R.A., Mapping the Visual Field to the Optic Disc, *Opthalmology*, 107:1809-1815, 2000.

[Healey and Mitchell, 2004] Healey P.R. and Mitchell P., Visibility of lamina cribrosa pores and open-angle glaucoma, *American Journal of Ophthalmology*, 138(5):871-872, 2004.

[Kaufman and Rousseeuw, 1987] Kaufman L. and Rousseeuw P.J., Clustering by means of medoids, *Statistical Analysis Based Upon the L1 Norm* Edited by: Dodge Y. Amsterdam, North Holland, pages 405-416, 1987.

[Kellam *et al*., 2001] Kellam P., Liu X., Martin N., Orengo C., Swift S. and Tucker A., Comparing, contrasting and combining clusters in viral gene expression data, In: *Proceedings of the IDAMAP2001 Workshop*, pages 56-62, London, UK, 2001.

[Kirkpatrick *et al*., 1983] Kirkpatrick S., Gelatt Jr C.D. and Vecchi M.P., Optimization by simulated annealing, *Science*, 220:671-680, 1983.

[Kohonen, 1989] Kohonen T., *Self Organization and Associative Memory* 3rd edition, New York: Springer-Verlag, 1989.

[Lukashin and Fuchs, 2001] Lukashin A.V. and Fuchs R., Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters, *Bioinformatics*, 17:405-414, 2001.

[Mandava *et al*., 1993] Mandava S., Zulauf M., Zeyen T. and Caprioli J., An Evaluation of Clusters in the Glaucomatous Visual Field, *American Journal of Ophthalmology*, 116:684-691, 1993.

[McQueen, 1967] McQueen J., Some methods for classification and analysis of multivariate observations, In: *The 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281-297, Berkeley, 1967.

[Monti *et al*., 2003] Monti S., Tamayo P., Mesirov J. and Golub T., Consensus clustering: a resampling-based method for class discovery and visualization of gene expression, microarray data, *Machine Learning*, 52:91-118, 2003.

[Snedecor and Cochran, 1989] Snedecor G. and Cochran W., *Statistical Methods* 8th edition. Ames: Iowa State University Press, 1989.

[Spenceley *et al*., 1996] Spenceley S. E., Henson D. B. and Bull D. R., Spatial Classification of Glaucomotas Visual Field Loss, *British Journal of Ophthalmology*, 80:526-531, 1996.

[Strehl and Ghosh, 2002] Strehl A. and Ghosh J., Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions, *Journal of Machine Learning Research*, 3:583-617, 2002.

[Swift and Liu, 2002] Swift S., Liu X., Predicting Glaucomatous Visual Field Deterioration Through Short Multivariate Time Series Modelling, Artificial Intelligence in Medicine, 24:5-24, 2002.

[Swift *et al*., 2004] Swift S., Tucker A., Vinciotti V., Martin N., Orengo C., Liu X. and Kellam P., Consensus Clustering and Functional Interpretation of Gene Expression Data, *Genome Biology*, 5(11): R94.1-R94.16, 2004.

[Tucker *et al*., 2001] Tucker A., Swift S. and Liu X., Variable Grouping in Multivariate Time Series Via Correlation, *IEEE Transactions on Systems, Man and Cybernetics* (*Part B: Cybernetics*), 31:235-245, 2001.

[Ward 1963] Ward J.H., Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association*, 58:236-244, 1963.

[Yeung *et al*., 2001] Yeung K.Y., Haynor D.R. and Ruzzo W.L., Validating clustering for gene expression data, *Bioinformatics*, 17:309-318, 2001.

Visual Feature Selection in Biological Time-Series for Mass Spectrometry based Biomarker Discovery

Christian Fuchsberger [a], **Sye-Min Christina Chan** [b], **Stefano Ongarello** [c], **Mike Sips** [b], **Isabel Feuerstein** [c], **Alexandre Pelzer** [a], **Günther Bonn** [c], **Georg Bartsch** [a], **Helmut Klocker** [a]

[a] Department of Urology, Medical University Innsbruck, Austria.

[b] Stanford University, USA.

[c] Institute of Analytical Chemistry and Radiochemistry, University of Innsbruck, Austria.

Abstract

The basic idea in biomarker discovery is the identification of highly discriminatory patterns between pathological samples and controls. Data analysis is a challenging task due to the so-called "small n, large p" problem: the relatively small number of subjects must be discriminated by time series consisting of hundreds of thousands of measurements.

We developed a robust visual feature selection method that uses different temporal abstractions to identify high-level discriminatory patterns. Furthermore, our method allows the ranking of the resulting patterns based on analytical and visual information.

1 Introduction

Biomarkers are biochemical features or substances, like proteins, that are specifically associated to a disease. Mass spectrometry (MS) is a high-throughput technology that is recently being used to discover disease-related proteomic patterns in complex mixtures of proteins derived from body fluids. Since known biomarkers, like PSA (Prostate Specific Antigen) for prostate cancer, often suffer from low specificity (while retaining high sensitivity) these proteomic patterns represent a promising approach for the early diagnosis of such diseases.

The result of a single MS run is a sequence of value pairs composed of intensity, which is coupled to the quantity of the detected substance, and mass-to-charge ratio (m/z), which depends on the molecular mass of the detected molecules and provides the information needed to identify the specific substance. The raw signal is characterized by several imperfections as a result of noise, machine miscalibration and various contaminants. Therefore, proper pre-processing, like noise removal and normalization, is needed.

There have been questions about the reproducibility and the reliability of the detected biomarkers. These questions are strongly connected with the feature selection process: in early experiments, the raw signal, consisting of hundreds of thousands of values, was used for classification without taking into account the "small n, large p" problem; irreproducibility was the consequence [Coombes *et al.*, 2005].

In later works, the number of features was reduced by extracting peaks for the identification of discriminatory features. However, due to high variance of up to 3.5% on the detected m/z, peak alignment across the spectra became crucial. To overcome this problem, a second abstraction level was introduced, namely regions of interest (ROIs)[Fushiki *et al.*, 2006], which are essentially peaks grouped together based on their neighbourhood (see Figure 1).

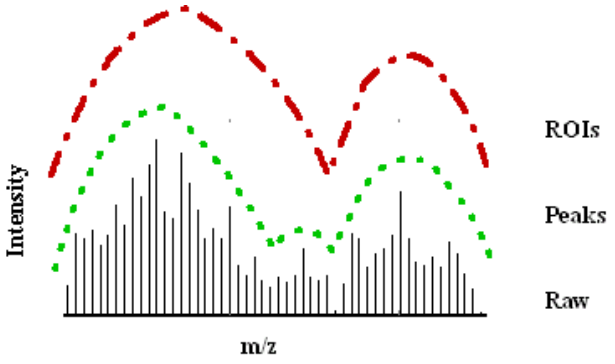


Figure 1: Abstraction levels

Since then, there has been a lot of work on the application of machine learning and statistical methods on feature selection [Liu and Motoda, 1998; Geurts, 2001; Bamgbade *et al.*, 2005], but most work considers only one abstraction level while it has been observed that domain experts normally inspect both the peak and the ROI abstraction levels simultaneously during analysis. Furthermore, such analytical methods lack the ability to consider visual information such as peak geometries, which can substantially reduce the false positive rate [Du *et al.*, 2006].

There exist visualization tools for the analysis of time series and mass spectrometry data, but to the best of our knowledge, none of them provides a framework in which analysts can perform analytical reasoning with the aid of visualizations that take into account MS specific features such as multiple abstraction levels. For example, Time Searcher 2 [Buono *et al.*, 2005] allows analysts to interactively specify a pattern, such as peaks, to search for. While it is a great tool for generic time series analysis, it is not suited for biomarker discovery, since it does not provide

support for the comparison and selection of retrieved patterns for classification. SpecAlign [Wong *et al.*, 2005] is designed specifically for the analysis of mass spectrometry data, and includes features such as spectral processing functions and spectra alignment capability, but it does not allow the interactive feature selection based on there discriminatory power.

In this paper we introduce a novel visual method for feature selection in biological time-series. Our interactive method is able to perform feature selection by (i) considering different levels of abstractions, (ii) taking into account the biological variance and (iii) visually and analytically ranking the identified features.

2 Method

The principle of "Visual Analytics" [Thomas and Cook, 2006] is to combine the outstanding visual capabilities of humans with the power of analytical methods to support the knowledge discovery process. Most importantly, the analyst is not only an interpreter of visual and analytical output, but also takes an active role in driving the whole process. According to [Thomas and Cook, 2006], the visualization must:

- Facilitate the understanding of large heterogeneous data sets
- Support the understanding of uncertain and incomplete data
- Provide adaptive representation for different user-tasks.
- Support different data types on various levels of abstraction into a single representation.

Our approach is to employ visual analytics to the feature selection process. With the aid of data mining techniques and statistical measures, the analysts can retrieve a list of biomarker candidates and have an objective means to compare different results. Using visualization, they can judge quickly and robustly whether a certain peak or ROI should be included for classification. And by allowing interaction, the analysts can easily incorporate their domain knowledge into the biomarker discovery process and quickly explore the parameter space.

2.1 Analysis

There are three steps in our approach: (i) peak finding, (ii) grouping peaks to ROIs, (iii) and computing the histograms of peak intensities for each group.

Finding peaks: To begin, we first search for interesting features or candidate biomarkers individually for each spectrum. We start with the identification of peaks by finding the local maxima over a neighborhood; in order to increase the robustness against noise, we place a threshold on the noise-to-signal ratio at the peak. Similar to [Morris *et al.*, 2005], noise is estimated to be the median absolute deviation (MAD) divided by a factor. In our system, both the noise-to-signal ratio threshold and the neighborhood size, expressed as a percentage of mass per charge ratio, are user specified and can be interactively adjusted by the analyst.

ROIs: Due to noise in measurements, peak locations can have a m/z variance of up to 3.5%. We therefore group the peaks across spectra according to their proximity in m/z ratio, with a restriction on the maximum distances among the peaks. We also restrict the minimum group size, since peaks that are displayed in only a few spectra are most likely resulted from noise or conditions that are not of interest in this setting. Again, both restrictions are user-defined and can be interactively adjusted to incorporate domain knowledge, like the value of peaks based on their shape.

Computing histograms: For each ROI created in step 2, we construct two histograms of peak intensities, one for the controls and the other for the cases. By normalizing the histograms by the size of their respective groups, we have an estimate of the underlying distribution of intensities. We then compute the distance between the two distributions within each ROI using the Jensen-Shannon (JS) Divergence:

Let $Hist_p$ and $Hist_c$ be the normalized histograms of the patient and the control groups respectively. M is the average of $Hist_p$ and $Hist_c$.

$D_{JS} = \frac{1}{2} D_{KL}(Hist_p||M) + \frac{1}{2} D_{KL}(Hist_c||M)$

where $D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$

Jensen-Shannon Divergence is an extension of Kullback-Leibler (KL) Divergence, which is an information theoretic measure of the differences between two probability distributions. Unlike KL divergence, JS divergence does not require absolute continuity in the distributions and is therefore more suited to our application. A comparison of the JS divergence gives us a sense of how much discriminatory power each ROI has.

2.2 Visualization

The feature selection process is supported by an interactive interface consisting of five panels (Figure 2). The timeline panel (panel 1) at the top is created to allow the analyst to have an overall idea of where the ROIs reside. It consists of a timeline that shows the range of mass per charge ratios under inspection. Each ROI is highlighted on the timeline, with the average peak location marked above or below the timeline. The gray box indicates the range that is currently being viewed.

The middle panel is the ROI panel (panel 2). It supports two views: the heat map view and the histogram view. The heat map view is essentially a table where each row represents a spectrum in the data, with detected peaks marked in green. The intensity of the green corresponds to the height of the peak, i.e. the darker the green is, the higher the peak is. The ROIs are shown as black bounding rectangles in the table. In the histogram view, the histogram set for each ROI is displayed. As discussed in the previous section, there are two histograms for each ROI; the left (red) histogram represents the controls, while the right (blue) histogram represents the cases. The Jensen-Shannon divergence between the two histograms is displayed below the plot. To avoid suppression due to high dynamic range in peak intensities (up to 40%) in the overall spectra, the range for a histogram set is determined according to the intensity range in the peak group.

The spectra panel (panel 3) at the bottom shows plots of the data spectra. This panel supports three different views of the data: the normal view, the ROI view, and the group

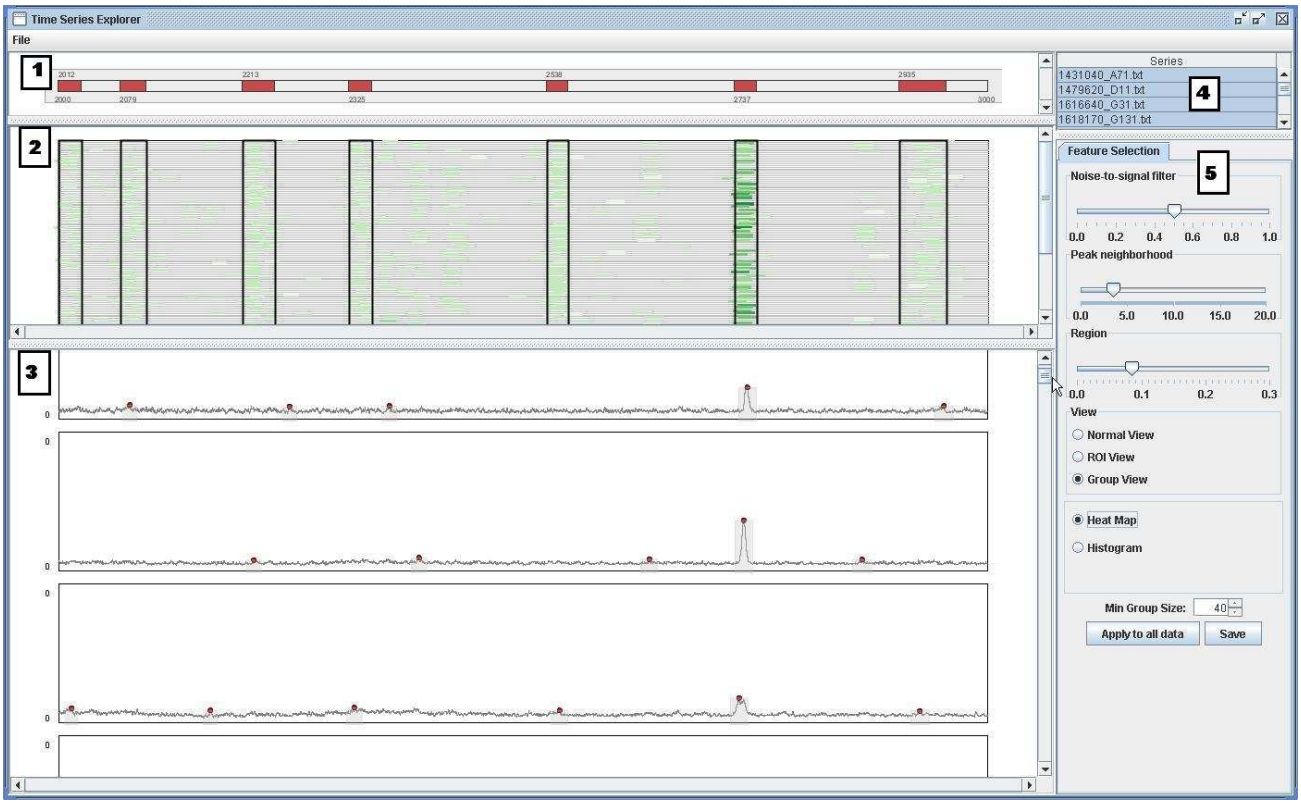


Figure 2: This shows the timeline panel (1), ROI panel (2), spectra panel (3), data panel (4), and the control panel (5) in grouped heat map view.

view. In the normal view, each spectrum is plotted separately to allow detailed inspection of individual spectrum. In this view, a peak is marked with a red dot, and the m/z range over which the peak location can vary is highlighted in gray. The ROI view is similar to the normal view except that only ROIs are displayed. In the grouped view, only ROIs are displayed, but instead of plotting each spectrum separately, the data spectra are sorted into their labeled groups (control or case) and plotted collectively. This facilitates the comparison across the two groups. To maximize the use of screen space, the ROIs are stacked together; in other words, each plot is composed of subplots of peak groups, created in the ROIs step. The peaks in each peak group have been aligned with the average spectrum shown in black. While information about the actual peak locations of the ROIs is lost in this view, the analyst can easily refer to the timeline panel for the m/z location.

On the right, there are the data panel (panel 4) and the control panel (panel 5). The data panel shows the list of spectra being analyzed. The control panel consists of sliders for the adjustment of parameters such as maximum noise-to-signal ratio and size of peak neighborhood, and radio buttons to toggle between different views.

2.3 Interactive Exploration

The choice of parameters is of key importance to the analysis process. Different sets of peak neighborhood size and maximum noise-to-signal ratio may lead to different sets of regions of interest. In order to allow the analyst to explore the parameter space and incorporate his or her domain knowledge into the selection process, fluid interaction and fast system response are necessary. This is achieved by dynamic query. The system first loads in a smaller subrange of the data spectra so that computations can be done in real-time. The analyst can then adjust the parameters using the sliders in the control panel. As the parameters change, the displays in the timeline, ROI, and spectra panels change accordingly. The fast feedback of the system provides a means of instant evaluation and allows the analyst to quickly find the desired set of parameters. After a choice is made, the analyst can either test the set on another sub-range for further refinement, or apply the set to the complete range.

When viewing the complete range, a problem that arises is the suppression of low peak intensities due to the high dynamic range in intensities. We solve this problem by following the principle of ”overview first, details on demand”. In the grouped view, each plot consists of a series of ROI subplots. To examine a ROI in more detail, the analyst can click on it to create a new plot of the ROI (Figure 3), scaled according to its own range. Moving the mouse over a plotline not only highlights the spectrum but also displays the label or name of the spectrum. This allows the analyst to have a detailed view of a ROI as well as compare individual spectrum against other spectra in the group.

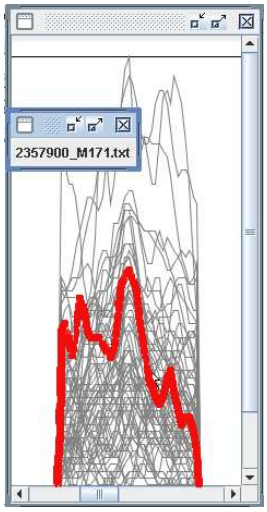


Figure 3: Detailed view of a ROI with a highlighted spectrum.

3 Case Study and Results

We tested the proposed visual feature selection method on a MS dataset composed of 60 healthy and 60 prostate cancer patients. We focused on the commonly used m/z range from 2,000 - 20,000 m/z, consisting of 95,743 measurements. All spectra were pre-processed according to the techniques described in [Morris *et al.*, 2005; Yu *et al.*, 2005]. Using the spectra panel two low quality spectra were identified and excluded by the two involved domain experts.

To start off, for peak detection, the analysts explored different noise-to-signal parameters (from 0.2 to 0.9 in 0.1 steps) and peak neighborhood settings (from 1% to 10% of the m/z value in 1% steps) to identify robust peaks and reduce the amount of noise (Figure 4). With the help of the

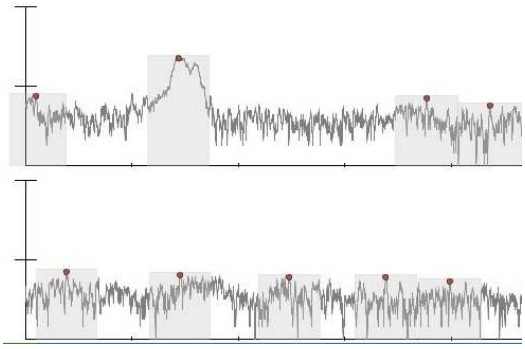


Figure 4: Peak identification. Dots mark identified peaks, gray boxes highlight the peak neighborhoods. Each plot represents one spectrum.

visual interface they could apply their background knowledge about peak shapes: starting with a high noise-to-signal ratio and narrow peak neighborhoods, they identified the maximum number of peaks including a lot of false positives caused by noise. Increasing step-wise the restrictions (lower noise-to-signal ratios and wider peak neighborhoods) they estimated the proper levels for robust peaks.

Then to group the peaks into ROIs, the analysts had to configure the region parameter properly to compensate for the detectable m/z variance across the spectra. The minimum group size was set to 40, which means that the same ROI must appear in 40 spectra to be included (Figure 5).

Afterwards, the analysts switched to the histogram view (see Figure 6) and ranked all 39 identified ROIs according to the Jensen-Shannon Divergence.

We compared our method with the feature extraction method proposed by Bamgbade et al.[Bamgbade *et al.*, 2005]. To investigate the discriminatory power of the identified feature we used a support vector machine classifier [Chang and Lin, 2001]. Additionally, we calculated the overlap between the selected features and the top 10 candidate biomarker molecules found in [Villanueva *et al.*, 2006]. The results, summarized in Table 1, show that our method did not generate a better classification rate; however, the higher overlap with [Villanueva *et al.*, 2006] suggests the identification of more reliable candidate biomarkers.

Method	Sensitivity	Specifitiy	Overlap with [Villanueva *et al.*, 2006]
Visual Method	78.5%	79.1%	80%
Evidence Accumulating	83.3%	80%	45%

Table 1: Results case study

4 Conclusion

We proposed a novel feature selection method based on an interactive visual framework for the identification of discriminatory features for biomarker discovery. We incorporated different levels of abstractions and integrated the analyst in the selection process to take advantage of their domain knowledge. The preliminary results show its validity for biological time series, where the visual aspect is of high importance due to high sample variances and different possible abstraction levels.

Acknowledgments

We would like to thank Silvia Miksch for the valuable comments. This research has been supported by GEN-AU (GENome Research in AUstria), and Max Planck Center for Visual Computing and Communication.

References

[Bamgbade *et al.*, 2005] A. Bamgbade, Ray L. Somorjai, B. Dolenko, Erinija Pranckeviciene, A. Nikulin, and Richard Baumgartner. Evidence accumulation to identify discriminatory signatures in biomedical spectra. In *Artificial Intelligence in Medicine, AIME 2005, Aberdeen, UK, July 23-27, 2005, Proceedings*, pages 463–467. Springer Berlin / Heidelberg, 2005.

[Buono *et al.*, 2005] Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. Interactive pattern search in time series. volume 5669, pages 175–186. SPIE, 2005.

[Chang and Lin, 2001] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[Coombes *et al.*, 2005] Kevin R. Coombes, Jeffrey S. Morris, Jianhua Hu, Sarah R. Edmonson, and Keith A. Baggerly. Serum proteomics profiling—a young technology begins to mature. *Nature Biotechnology*, 23:291–292, 2005.

[Du *et al.*, 2006] Pan Du, Warren A. Kibbe, and Simon M. Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, 2006.

[Fushiki *et al.*, 2006] Tadayoshi Fushiki, Hironori Fujisawa, and Shinto Eguchi. Identification of biomarkers from mass spectrometry data using a common peak approach. *BMC Bioinformatics*, 7:358, 2006.

[Geurts, 2001] Pierre Geurts. Pattern extraction for time-series classification. In *Proceedings of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery*, volume 2168, pages 115–127. Springer Berlin / Heidelberg, 2001.

[Liu and Motoda, 1998] Huan Liu and Hiroshi Motoda. *Feature Extraction, Construction and Selection. A Data Mining Perspective*. Springer, US, 1998.

[Morris *et al.*, 2005] Jeffrey S. Morris, Kevin R. Coombes, John Koomen, Keith A. Baggerly, and Ryuji Kobayashi. Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. *Bioinformatics*, 21(9):1764–1775, 2005.

[Thomas and Cook, 2006] J.J. Thomas and KA. Cook. A Visual Analytics Agenda. *IEEE Computer Graphics and Applications*, 26(1):10–13, 2006.

[Villanueva *et al.*, 2006] Josep Villanueva, David R. Shaffer, John Philip, Carlos A. Chaparro, Hediye Erdjument-Bromage, Adam B. Olshen, Martin Fleisher, Hans Lilja, Edi Brogi, Jeff Boyd, Marta Sanchez-Carbayo, Eric C. Holland, Carlos Cordon-Cardo, Howard I. Scher, and Paul Tempst. Differential exoprotease activities confer tumor-specific serum peptidome patterns. *The Journal of Clinical Investigation*, 116:271–284, 2006.

[Wong *et al.*, 2005] Jason W.H. Wong, Gerard Cagney, and Hugh M. Cartwright. SpecAlign–processing and alignment of mass spectra datasets. *Bioinformatics*, 21(9):2088–2090, 2005.

[Yu *et al.*, 2005] J. S. Yu, S. Ongarello, R. Fiedler, X. W. Chen, G. Toffolo, C. Cobelli, and Z. Trajanoski. Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data. *Bioinformatics*, 21(10):2200–2209, 2005.

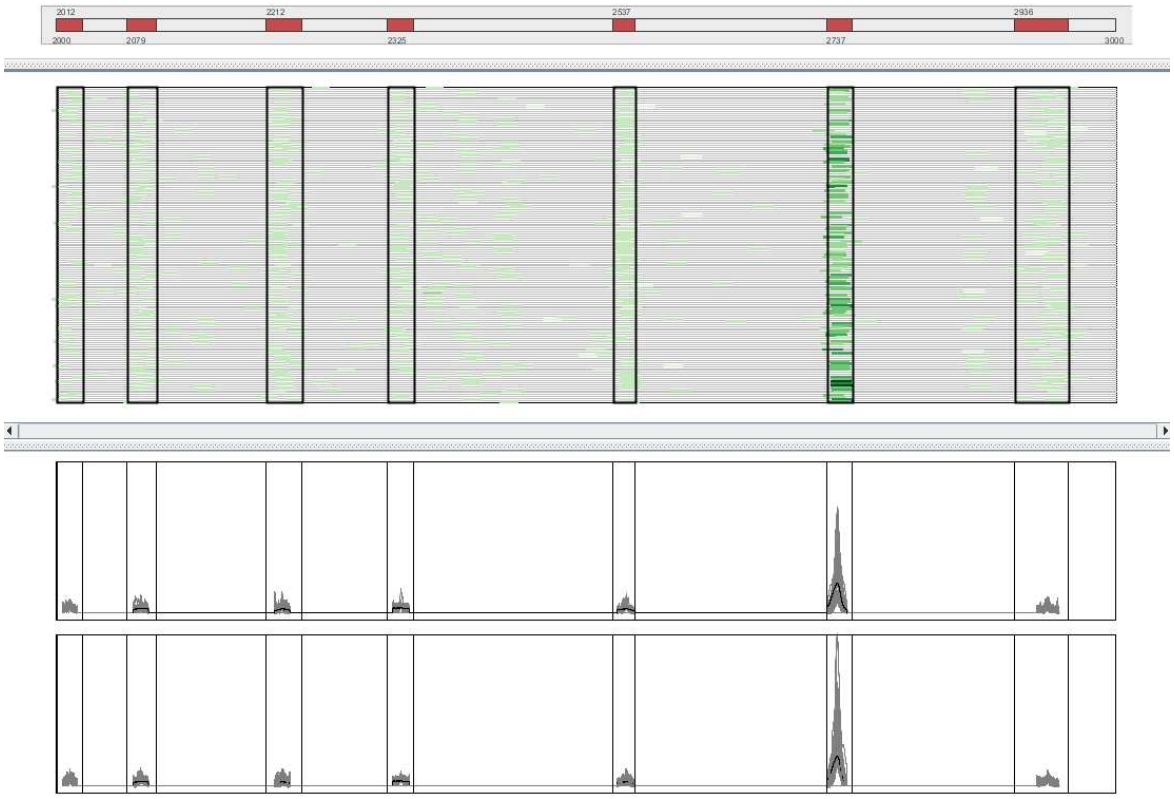


Figure 5: Regions of interest (ROIs)

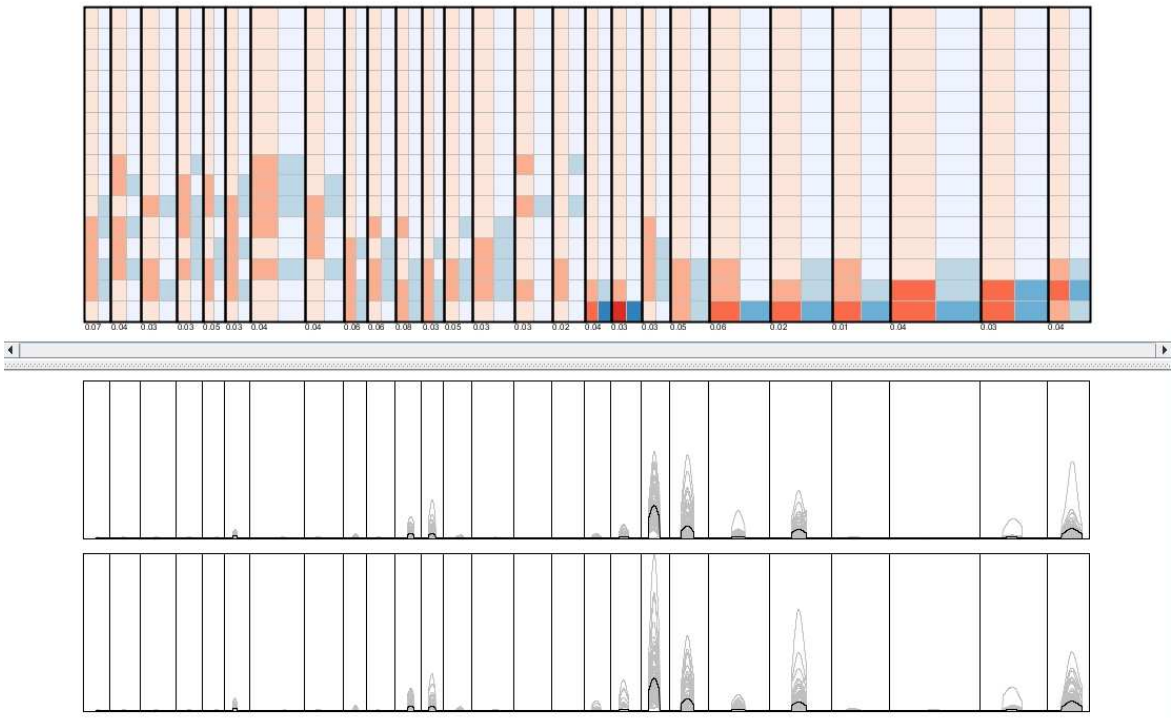


Figure 6: Identified discriminant ROIs in the histogram view

Intelligent Visualization of Temporal Associations for Multiple Time-Oriented Patients Records*

Denis Klimov and Yuval Shahar
Medical informatics Research Center
Department of Information System Engineering
Ben Gurion University of the Negev, Beer Sheva, Israel
{klimov, yshahar}@bgu.ac.il

Abstract

In this paper we present several novel knowledge-based visualization techniques for temporal and statistical associations for multiple patient records. These visualizations are part of an interactive system, called VISITORS, which enables intelligent visualization and exploration of patient longitudinal data. We finished the functional evaluation, and are in the process of an usability evaluation of the capability of clinicians to construct complex temporal queries.

1 Introduction: Knowledge-Based Temporal Analysis

In a medical world with a large volume of time-stamped information, the clinicians and medical researchers need useful, intuitive intelligent tools to process the multiple time-oriented patient data. Standard means, such as tables, statistical tools, and even more advanced temporal data mining techniques, are often insufficient, can help only in particular cases, or require special experience.

To solve the computational aspect of this problem, we have been using the *knowledge-based temporal abstraction* (KBTA) method [Shahar, 1997] for automated derivation of meaningful interpretations and conclusions, called *temporal abstractions*, from the raw time-oriented patient data, using a domain-specific knowledge-base (KB). The input of the KBTA method includes a set of time-stamped parameters (e.g., platelet counts) and events (e.g., bone-marrow transplantation (BMT)); and the output is a set of interval-based, context-specific parameters at the higher level of abstraction (e.g., a period of nearly 3 months of grade II bone-marrow toxicity). Furthermore, the output temporal abstractions can be efficiently visualized. The KNAVE-II system, which we had developed previously [Shahar *et al.*, 2006], supports the visualization and exploration of raw data and derived temporal abstractions for an *individual* patient or a small group of patients.

However, to analyze clinical trials, or for quality assessment purposes, an aggregated view of a group of patients is more effective than exploration of each individual record separately. In addition, certain patterns can only be discovered through the analysis of multiple patients.

Therefore, we have designed and developed a new system called VISualizatIon of Time-Oriented RecordS (VISITORS) [Klimov, 2005] which combines the intelligent temporal analysis and information visualization techniques:

- Time-oriented data are graphically displayed and explored for both individual and multiple patients.
- The time in our conceptual and graphical representations is of major importance. It can be explored in various granularities, such as hour, day, and month. We also support a calendar (absolute) timeline and a timeline relative to special events (e.g., the six months following a BMT).
- The computational reasoning supports not only a view of raw time-oriented data and their statistics but also a summarization of the raw data as clinically meaningful concepts, based on the temporal-abstraction domain ontology and the KBTA computational mechanisms.

2 The VISITORS system

VISITORS is an intelligent visualization system specific to the tasks of querying, knowledge-based visualization and interactive exploration of time-oriented patient records. It interacts with the time-oriented mediator (which manages the relevant knowledge and data bases) through temporal aggregation queries: *Select Patients Query*, *Select Time Intervals Query* and *Get Patients Data Query*, based on the aggregation query-language semantics [Klimov, 2007]. These queries retrieve the list of patients, list of relevant time intervals and time-oriented patients data. For example, the typical *Select Patients Query* is: ”Select all male patients, either younger than 20 or older than 70, whose hemoglobin (HGB) state was abstracted as “moderately low” or lower, during at least seven days, starting at least two weeks after BMT”.

2.2 Temporal Association Graph

The data set retrieved by the *Get Patients Data Query* can be visualized and explored using an appropriate visualization [Klimov, 2005]. However, the most interesting task is to discover new interrelations or patterns, especially temporal interrelations, within a set of patient data. For such purposes, we developed the *temporal association graph*.

* The IDAMAP workshop is organized in collaboration with the IMIA Intelligent Data Analysis and Data Mining WG and the AMIA Knowledge Discovery & Data Mining SIG.

Delegate Functions

In order to aggregate the patients data we defined a "*delegate value*" method: Given the patient's time-oriented data for the specific concept (raw or abstract), for each patient, over a specific time interval, we calculate the delegate (representative) value of the patient's data by using a representative function specific to each temporal granularity and defined in a KB or chosen by the user. For example, assume that the patient had on Jan 1 three laboratory tests of the hemoglobin (HGB): 8.80 *g/dl* at 5AM, 9.30 *g/dl* at 11AM and 11.90 *g/dl* at 8PM. If we select the *mean* as the delegate *daily* function, then the patient had a 10 *g/dl* daily average value for HGB. However, the user can choose another suitable representative function (such as *mode* or *maximum*).

Temporal Association Graph Interface

In Figure 1 we show the temporal and statistical associations among four concepts selected by the user for 58-patients group, retrieved by using a *Select Patients Query*. In this visualization we can see the distribution of the values of the HGB and Platelet state abstractions; and the WBC and RBC yearly average values for each patient over 1995. Values of all parameters for each patient are connected by lines. Only 45 patients in this group have data during 1995.

Edges between abstract concepts provide additional statistical information, and represent the relation of specific values of the first concept to specific values of the second concept. The edge width denotes the proportion of the patients population. The support, confidence and actual number of patients of relation are displayed on the edge. For example, the widest edge in Figure 1 represents a relationship between the "*low*" value of the Platelet state and the "*moderately_low*" value of the HGB state: 55.8% of the patients have this *combination of values* (i.e. *support*), while 96.6% of the patients who have the "*low*" value of the Platelet state, have the "*moderately_low*" value of the HGB state (i.e. *confidence*), and this association is valid for 25 patients.

In this visualization, by using direct manipulation, the user can dynamically apply additional constraints; e.g., we can answer the question "how constraining one parameter can affect the association between multiple concepts". The user is able to select another range of the values for the raw-concept data by using trackbars, or to select a subset of the relevant values for the abstract concept. For example (See Figure 2), increasing the WBC value from 1400 *cells/ml* to 5400 causes a refiltering of the patient data: in the new subgroup, there are no patients with a "*very_low*" value of the Platelet state, and only half of the patients have a "*moderately_low*" value of the HGB state.

The system also supports displaying associations of one or several concepts among *different* time intervals, changing the order of displayed concepts, and displaying associations in relative time (i.e. starting from some event).

We used the *parallel coordinates* visualization in our system, one of the most popular multidimensional visualization techniques. This technique was previously used in *The Cube* system [Falkman, 2001]. With *The Cube*, a clinician can interactively select a number of attributes of the patient record, inspect the resulting multiple 3D diagrams,

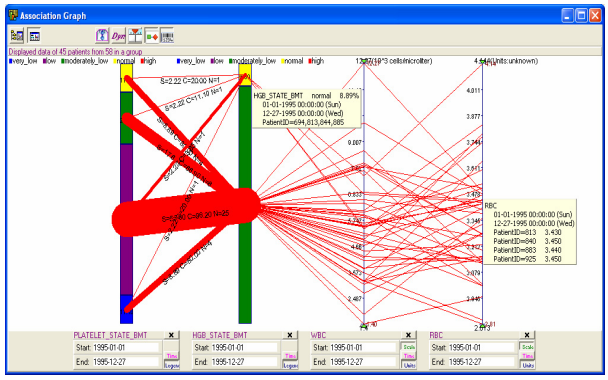


Figure 1. Temporal association graph visualization

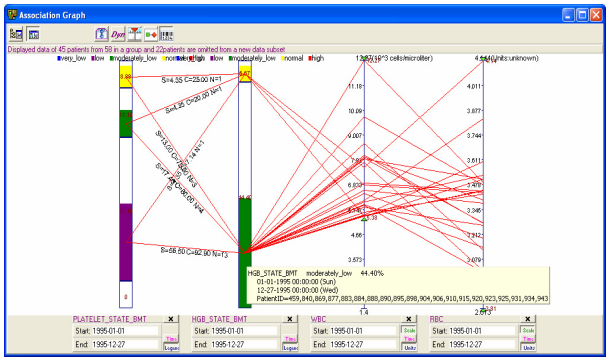


Figure 2. Dynamic applying of constraints to the patient data

and recognize the patients pattern (the similar patients have a parallel lines between attributes).

However, we emphasize in our work the temporal aspects of the visualized data, automatic aggregation of patients and on a clinically meaningful summarization of raw patient data, enabled by using medical knowledge.

Acknowledgments

This research was supported by Deutsche Telekom Company and Israel Ministry of Defense.

References

[Shahar, 1997] Yuval Shahar. A framework for knowledge-based temporal abstraction. *Artificial Intelligence*, 90(1-2), 1997

[Shahar *et al*., 2006] Yuval Shahar, Dina Goren-Bar, David Boaz and Gil Tahan. Distributed, intelligent, nteractive visualization and exploration of time-oriented clinical data and their abstractions. *Artificial Intelligence in Medicine*, 38 (2):115-135, 2006.

[Klimov, 2005] Denis Klimov and Yuval Shahar. A framework for intelligent visualization of multiple time-oriented medical records. In *Proceedings of AMIA Annual Symposium 2005*, pages 405-409.

[Klimov, 2007] Denis Klimov and Yuval Shahar. Intelligent Querying and Exploration of Multiple Time-Oriented Medical Records. In *Press of Proceedings of MEDINFO 2007*.

[Falkman, 2001] Goran Falkman. Information visualization in clinical Odontology: multidimensional analysis and interactive data exploration, *Artificial Intelligence In Medicine*, 22(2):133-158, 2001.

Reduction of Large Training Set by Guided Progressive Sampling: Application to Neonatal Intensive Care Data

François Portet[1], Feng Gao[1], Jim Hunter[1] and René Quiniou[2]
[1]Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, Scotland, UK
[2]Irisa, INRIA, Campus de Beaulieu, 35042, Rennes, France
{fportet,fgao,jhunter}@csd.abdn.ac.uk, quiniou@irisa.fr

Abstract

Although large training sets are supposed to improve the performance of learning algorithms, there are limits to the volume of data such an algorithm can handle. To overcome this problem, we describe an improvement to a progressive sampling method by guiding the construction of a reduced training set. The application of this method to neonatal intensive care data shows that it is possible to reduce a training set to a third of its original size without decreasing performance.

1. Introduction

Intensive Care Units generate large volumes of data - about 1 MB per patient per day. However, such large volumes are difficult to analyze, so data mining or machine learning techniques are often used to learn classifiers for prediction and decision support. Although the general approach is to learn classifiers from the largest possible dataset, learning a classifier from too large a dataset can be computationally impossible or time-consuming and thus the training set must be reduced.

'Data reduction' aims at aggregating the information contained in large datasets into manageable smaller information chunks, using simple tabulation, clustering, principal component analysis (PCA), etc. However, these methods need either data pre-processing or modification of the example datasets in such a way that it is more difficult to interpret the model which has been learned (e.g. PCA). Progressive Sampling (PS) [Provost et al. 1999] incrementally constructs a training set from a larger dataset without decreasing the classification performance and without altering the initial format of the examples. In this paper, we propose a variant of PS and show its application to the domain of Neonatal Intensive Care.

2. Progressive Sampling

Progressive Sampling (PS) starts with a small training subset (*TS*) of the full dataset (*FDS*) and incrementally extends *TS* until the learning accuracy satisfies some convergence criteria. The resulting dataset is expected to be smaller than *FDS* and to lead to (at least) the same performance. Figure 1 shows the general algorithm.

Let *FDS* be the Full Dataset
Let $S = \{n_0, .. n_k, \ldots n_K\}$ be the planned sizes of *TS*
$k = 0$;
While *not converged* **do**
  TS ← computeTS(*FDS*) // copy n_k examples from *FDS* to *TS*
  M ← learn(*TS*) // learn the model *M*
  Evaluate(*M, FDS*) // evaluate *M* on *FDS*
  inc(*k*)
End do
Return *M*

Fig. 1 Progressive sampling algorithm.

Before starting the learning process, the progressive sizes of *TS* are scheduled (planned). Then *TS* is used to learn the classifier model *M* (by a decision tree, neural network, etc.) which is tested until convergence is attained. The optimal training set is computed by mean of a learning curve which is used to retain the best balance between size and learning performance. Provost *et al.* [1999] have showed that when dealing with large volume of data, PS is more efficient than using the entire dataset. However, PS does not explicitly deal with unbalanced datasets. To face this problem, Ng and Dash [2006] introduced a method to improve the relative distribution of each class by over-sampling the minor class in *computeTS(FDS)*. But, as they emphasized, replicating examples from the smaller classes (over-sampling) leads to over-fitting.

These approaches select the examples to be added into *TS* at random. We believe that it is possible to speed up the convergence by using *a priori* information to select the most appropriate examples to add.

3. Guided Progressive Sampling

Guided PS (GPS) uses a distance measure *d* between the samples in *TS* and the samples in *FDS* to guide the selection of samples to add to *TS*. Once *M* is learned, each $e_i \in FDS$ is tested to form the triple $(e_i, m(e_i), d(e_i))$ where $m(e_i)$ is the

result of the classification of e_i using M ($m(e_i) \in$ {correct, incorrect}) and $d(e_i)$ gives the distance from e_i to the centroid of the class to which it actually belongs. This set of triples is used in *computeTS(FDS)* according to one of two strategies:

1. ***GPS*** adds to each class in *TS*, the worst *misclassified* examples i.e. those with the highest values of $d(e_i)$. This is intended to improve learning robustness by considering the difficult cases.
2. ***GPS+*** extends GPS by additionally adding the best *correctly classified* examples i.e. with the lowest values of $d(e_i)$. This is intended to reinforce learning stability which can be distorted by only including the worst misclassifications.

These choices rest on the assumption that learning is most influenced by the extreme examples of each class (correct classifications and misclassifications). The distance measure *d* does not need to be exact (otherwise it would be directly used to learn the model!) but is a heuristic estimate of how much the classification is wrong.

4. Case study: bradycardia detection

The method has been tested on the detection of bradycardias by decision tree learning (C4.5 with pruning). The dataset consists of thirteen heart rate (HR) time series each covering 24-hours recorded from premature babies receiving intensive care. The episodes of bradycardia were annotated by two clinical experts. Each example in FDS is described by 25 attributes (raw HR value and min, max, slope etc. over several centered windows). The size of the complete 13 record dataset is more than 80MB. Given such a large dataset, learning on the entire set is impossible. Moreover, the dataset is completely unbalanced. For example, in record #16234, the *bradycardia* class contains 533 examples whereas the *no-bradycardia* class contains 79875 examples, the *bradycardias* representing only 0.66% of the total dataset. However, this is to be expected, as bradycardia is defined as a short transient event. In addition, the records contain episodes of artifact that can perturb learning.

Random sampling (RS), GPS, and GPS+ have been used. To try to balance the large difference between *bradycardia* and *no-bradycardia*, the initial TS contained 100% of the *bradycardias* and 1% (selected at random) of the *no-bradycardias*. On each iteration, 3.33% more of the *no-bradycardias* were selected from *FDS* according to the particular strategy in use. Learning was stopped when the learning curve become sufficiently stable [Provost *et al.,* 1999].

Fig.2 shows the number of classification errors against the training size for record #16234. GPS converged faster than GPS+ and RS. GPS+ and RS converged at the same iteration however GPS+ led to higher accuracy. GPS and GPS+ produced more errors at the beginning of the process as they initially selected the most difficult examples to classify but this led rapidly to a more stable plateau (fewer oscillations) than RS. The figure shows that after reaching the beginning of the plateau, the examples added do not provide information that has not already been learned by the decision tree. Results found with GPS led to 111 errors for TS=9317 examples. Thus around 90% of the dataset is not useful for learning. The decision tree learned with GPS led to the same performance (111 errors) as the decision tree learned from FDS but with a slightly smaller tree. The proportion of *bradycardias* is still not equally distributed but increased from 0.66% to 5.72%.

Mean accuracy over the 13 datasets was 99.66% for RS (20 runs), 99.84% for GPS+ and 99.85% for GPS, with significant differences between GPS (or GPS+) and RS ($p<0.04$ in the worst case, $p<0.0001$ for #16234).

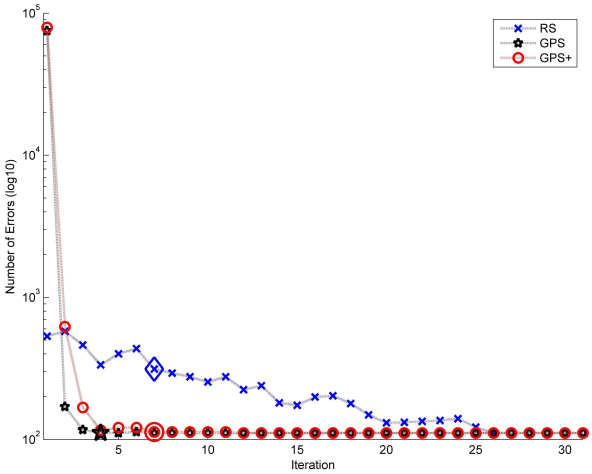


Fig. 2 Progressive learning for the record #16234. Large marks show convergence.

5. Discussion

Guided progressive sampling has shown to be more efficient than random progressive sampling for learning from a massive training set. Using *a priori* knowledge to guide the sampling leads to a faster convergence and a better selection of the "relevant" examples to use for learning. Further experiments will be undertaken to improve bradycardia detection with the reduction of larger datasets. This approach can also be useful in a situation with a small dataset to capture the "best" training examples.

References

[Provost *et al.,* 1999] F. Provost, D. Jensen and T. Oates. Efficient progressive sampling. In *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, USA. 1999.

[Ng and Dash, 2006] W. Ng and M. Dash. An Evaluation of Progressive Sampling for Imbalanced Datasets. In *Sixth IEEE International Conference on Data Mining Workshops*. Hong Kong, China. 2006.

Paper session:

Classification and Filtering

The Weight of Variable Groups for the Prediction of Probability of Survival in ICU Patients

Oscar Luaces[1], **Francisco Taboada**[2], **Guillermo M. Albaiceta**[2],
Antonio Bahamonde[1], **GRECIA Group**[3]
[1]Centro de Inteligencia Artificial. Universidad de Oviedo at Gijón. Gijón, Asturias, Spain
[2]Hospital Univ. Central de Asturias (HUCA). Universidad de Oviedo. Oviedo, Asturias, Spain
[3]Grupo de Estudios y Análisis en Cuidados Intensivos

Abstract

In this paper we discuss the weight of groups of variables involved in the prediction of probability of survival in patients admitted to an Intensive Care Unit (ICU). The whole set of variables describe the state of critically ill patients, and it is divided in three groups: clinical, monitoring and laboratory data. The weight is assessed according to the performance of the prediction models that can be built with a group of variables using a method based on Support Vector Machines (SVM). In this way, we measured the differences between the relevance of monitoring and laboratory data in some contexts with acknowledged medical differences. Additionally, we identified that most of the prediction capabilities of SVM models are captured by a group of basic clinical data that are routinely recorded in ICU admissions. The conclusion is that it is possible to tailor reliable and cheap prediction models for specific kinds of patients and ICUs.

1 Introduction

Predictions of probability of survival in critically ill patients are mainly used to measure the efficacy of Intensive Care Unit (ICU) treatments. The risk stratification of patients allows comparison of the observed outcomes versus accepted standards provided by score functions. Notice that ICU assessment is very important since it is estimated that end-of-life care consumes 10% to 12% of all healthcare costs. Moreover, in 2001 the average daily cost per patient in ICUs was about $\$3000$ in the USA [Provonost and Angus, 2001]. On the other hand, the literature also shows that prognoses have constituted an important dimension of critical care, as patients and their families seek predictions about the duration and outcome of illness [Lemeshow *et al.*, 1993].

The available models for predicting outcomes in ICUs are usually scoring systems that estimate the probability of hospital mortality of critically ill adults. This is the case of APACHE (*Acute Physiology And Chronic Health Evaluation*)[Knaus *et al.*, 1991], SAPS (*Simplified Acute Physiology Score*) [Le Gall *et al.*, 1984], and MPM (*Mortality Probability Models*) [Lemeshow *et al.*, 1993]. The score functions of these predictors were induced from data on thousands of patients using logistic regression. The data required by these systems is gathered, for each patient, in a set of variables that can be split into 3 groups according to the source of information: monitoring devices, laboratory analysis, and demographic and diagnostic features.

In this paper we seek to measure the weight or relevancy of these 3 groups of variables. The idea is to assess each group of variables according to their performance when they are used to learn the probabilities of survival. For this purpose we shall use a method by Luaces *et al.* [2007] based on *Support Vector Machines SVM* [Vapnik, 1998] that optimizes the *Area Under the ROC Curve (AUC)* prior to fit a sigmoid using the scaling algorithm of Platt [2000]. The next section details this Machine Learning method.

The aim of the paper is to gain insight into all the factors that contribute to the actual prediction capabilities in different medical meaningful contexts. Thus, we shall discuss the role of groups of variables in Units with and without coronary patients, and in patients aggregated according to the treatment location immediately prior to their ICU admission.

In all cases we found medical explanations to back our achievements, but the contribution of the paper is that we can provide accurate measurements. Moreover, our results suggest that it is possible to build customized prediction systems according to the peculiarities of ICUs and patients. These predictors could be reliable, and their construction and use could be cheap since they would require a small number of variables. Let us recall that some of the variables used for the prognostic scores mentioned above are not eventually part of the clinical routine, and may not be registered in some patients. Therefore, simplifying the data sets required could make the calculation of a score easier, even in a retrospective manner.

The study presented here was done using data collected in general ICUs at 10 hospitals in Spain, 6 of which include coronary patients, while the other 4 do not treat coronary diseases. The total number of patients considered in our study was 2501, 19.83% of whom did not survive.

2 Predicting Probabilities with SVM

We shall face the prediction of probabilities task from the point of view of Machine Learning. Thus, in order to induce such predictions, we collect training sets of pairs of descriptions of patient states and their outputs codified by '$+1$' when the patient has survived, and '-1' otherwise.

The first temptation is to tackle this learning task as a binary classification since there are two classes. Unfortunately, we must acknowledge that the performance of classification learners is not satisfactory in the ICU problem; otherwise, nobody would turn to probabilities. However, some useful knowledge, represented by probabilities, can be drawn from data, despite accurate crisp predictions are difficult to make.

An important issue when we are learning is to fix the way in which we are going to measure the quality of the result using the so-called loss functions. Formally, given a training set $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ for a learning task, the aim is to find a hypothesis h (from a given hypothesis space) able to return outputs y_i from entries $\boldsymbol{x}_i$ of an input space $\mathcal{X}$ that minimizes the average *loss* extended over the set of independently identically distributed (i.i.d.) test sets S', usually represented by $\Delta(h, S')$.

When predictions are discrete probability distributions, usually the standard loss function is the average quadratic deviation. If there are two possible outputs, the probability loss is given by

$$\Delta_{\mathrm{Pr}}(h, S') = \frac{1}{|S'|} \sum_{\boldsymbol{x}'_i \in S'} (h(\boldsymbol{x}'_i) - p_i)^2 \qquad (1)$$

where the hypothesis h returns the estimation of the probability $h(\boldsymbol{x}) = Pr(y = +1|\boldsymbol{x})$, and p_i stands for the observed probability of the i-th case, $p_i = Pr^{true}(y = +1|\boldsymbol{x}_i)$.

The measurement in Equation (1) is frequently used in medicine and meteorology, and is known as the Brier [1950] index or score.

In the next section we shall spell out a method to learn probabilities in this context using Support Vector Machines (SVM), a state-of-the-art family of algorithms for learning tasks [Vapnik, 1998].

2.1 Optimizing a loss function plus a sigmoidal transformation

When used to learn a binary classification, SVMs compute a function that returns continuous numbers: positive values for cases of one of the classes, and negative for the other class. This function is always a linear map in a so-called feature space $\mathcal{H}$ where we represent the elements $\boldsymbol{x}$ of the input space $\mathcal{X}$. If $\phi : \mathcal{X} \to \mathcal{H}$ is the representation map, the classification learned by a SVM is accomplished by a hypothesis

$$\mathrm{sign}(f(\boldsymbol{x})) = \mathrm{sign}(\langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b) \qquad (2)$$

where $\langle \cdot, \cdot \rangle$ is the inner product that necessarily must exist in $\mathcal{H}$, $\boldsymbol{w} \in \mathcal{H}$, and $b \in \mathbf{R}$ is a real number. Both $\boldsymbol{w}$ and b are the unique solution to a quadratic convex program that optimizes the margin between the classes, and the errors measured with a given loss function. These two aims are weighted by means of a regularization parameter C. For instance, in classification SVM the loss function counts the proportion of misclassified cases; therefore, the SVM is asked to improve the accuracy of the classifier. In symbols, this loss function is given by

$$\Delta_{\mathrm{Ac}}(h, S') = \frac{1}{|S'|} \sum_{\boldsymbol{x}'_i \in S'} \mathbf{1}_{\{(h(\boldsymbol{x}'_i) \neq y'_i)\}} \qquad (3)$$

Formally, the convex optimization program to be solved by the SVM is the following:

$$\begin{aligned} \min_{\boldsymbol{w}, \xi} \quad & \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{n} \xi_i, \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i) \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, n \end{aligned} \qquad (4)$$

It can be seen that $\boldsymbol{w}$ is a linear combination of the representations in $\mathcal{H}$ of inputs of the training set S. Therefore, the discrimination function (Eq. (2)) can be described by

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b = \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b, \qquad (5)$$

where

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle \qquad (6)$$

is called the *kernel* function of the transformation ϕ. We shall use the rbf kernel that is defined by

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|^2}{2\sigma^2}}. \qquad (7)$$

To learn a probability using SVM, it is crucial to transform their scores or continuous outputs into probabilities. This is what a method presented by Platt [2000] does. The core idea is to fit a sigmoid, using a maximum likelihood procedure. Thus we obtain, from the function of Eq. (5) (that was learned to improve the accuracy), a hypothesis to estimate probabilities

$$h_{ac}(\boldsymbol{x}) = Pr(y = +1|\boldsymbol{x}) = \frac{1}{1 + e^{A_{ac} \cdot f(\boldsymbol{x}) + B_{ac}}} \qquad (8)$$

The method described so far produces good results (see the scores reported in section 4, Table 2); however, it could be argued that even better results could be achieved if we were able to include somehow the loss function of Eq. (1) into the optimization problem of Eq. (4). But this is not possible. However, following Luaces *et al.* [2007], we can observe that the sigmoid of Eq. (8) is a strictly increasing function that preserves the ordering induced by the function f (Eq. (5)) that gives rise to the classification hypothesis. Thus, it is reasonably to postulate that to compute Platt's sigmoid it is better to look first for a function like f whose objective was to keep the ordering as coherent as possible with the ordering of classes.

The advantage of this approach is that it is possible to measure the degree of coherence of the orderings in a way that can be then explicitly stated in an optimization convex problem. In fact, the Area Under the ROC (receiver operating characteristic) Curve (*AUC* for short) was interpreted by Hanley and McNeil [1982] as the probability of a correct ranking induced by a function f; in other words, it is the probability that a randomly chosen subject of class '$+1$' is (correctly) ranked by f with greater output than a randomly chosen subject of class '-1'. Therefore, AUC coincides with the value of the Wilcoxon-Mann-Whitney statistic. The relationship between the AUC and the Brier score has already been dealt in [Brier, 1950]. Therefore, according to the probabilistic interpretation of AUC, the complementary of this amount ($1-$AUC) can be used as a

loss function. Thus, if g is a discrimination function like f of Eq. (5), its loss evaluated on a test set S' is

$$\Delta_{AUC}(g, S') = Pr(g(\boldsymbol{x}'_i) \leq g(\boldsymbol{x}'_j) | y'_i > y'_j) = \\ = \frac{\sum_{i,j:y'_i>y'_j} \mathbf{1}_{\{g(\boldsymbol{x}'_i)\leq g(\boldsymbol{x}'_j)\}}}{\sum_{i,j} \mathbf{1}_{\{y'_i>y'_j\}}} \quad (9)$$

The convex optimization problem devised to improve this loss function is the following [Joachims, 2005; 2006]

$$\min_{\boldsymbol{w},\xi} \quad \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle + C \sum_{i,j:y_i>y_j} \xi_{i,j} \quad (10)$$
$$\text{s.t.} \quad \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle - \langle \boldsymbol{w}, \phi(\boldsymbol{x}_j)\rangle \geq 1 - \xi_{i,j},$$
$$\xi_{i,j} \geq 0, \quad \forall i, j : y_i > y_j$$

The solution of this problem gives rise to a function

$$g(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x})\rangle = \sum_{y_i>y_j} \alpha_{i,j}(K(\boldsymbol{x}_i, \boldsymbol{x}) - K(\boldsymbol{x}_j, \boldsymbol{x})) \quad (11)$$

Then, using again Platt's algorithm, we obtain

$$h_{AUC}(\boldsymbol{x}) = Pr(y = +1|\boldsymbol{x}) = \\ \frac{1}{1 + e^{A_{AUC}\cdot g(\boldsymbol{x}) + B_{AUC}}} \quad (12)$$

3 Groups of Variables for Predicting Probabilities

In the experiments reported in the next section, we shall describe the state of patients using the set of variables employed by APACHE III, the golden standard of the field. These variables, in addition to demographic and a brief clinical history, include 16 acute physiologic records that use the worst values from the first 24 hours in the ICU. In order to study the prediction capabilities of these variables, we have divided the whole set into 3 groups according to the source of information of these variables, see Table 1.

We labelled the first group of variables with the tag *clinical*. In this group we collect demographic and diagnostic data adding simple tests or observations. Let us emphasize that the recording of these data can be done without costs.

On the other hand, the second group of variables, monitoring, is formed by those data supplied by *monitoring* devices. Finally, the third group of variables comes from *laboratory* analysis.

4 Experimental Results

In this section we report a number of experiments carried out with a real data set described in the next subsection. The aim was first to show the performance of the SVM methods described in section 2, and then to gain insight into the weight of groups of variables involved when the predictions are sought in different contexts. We shall discuss contexts defined by different kinds of ICUs, and contexts characterized by the treatment location of patients immediately prior to ICU admission.

Group	Variables
Clinical	age, sex, mechanical ventilation, pre-existing comorbilities, major diagnostic category, type of patient (scheduled or urgent surgery, trauma, medical) location prior to ICU (other hospital, ward, scheduled or urgent surgery), itemized Glasgow Coma Score
Monitoring	temperature, blood pressure, heart and respiratory rate, urinary output
Laboratory	gas exchange (PO_2, PCO_2), white cell count, hematocrit, serum: sodium, blood urea nitrogen, creatinine, albumin, bilirrubin, glucose

Table 1: The division of variables used to record the state of UCI patients into 3 groups according to their source of information

In all cases, performance estimations were made using a 10-fold stratified cross-validation on each of the data sets, for all prediction methods except for APACHE III; since it was already trained with a different data set, we used the available data to test its predictions. Additionally, the data was standardized according to the mean and deviation observed on each training fold.

4.1 The Data Used in the Experiments

We used data collected from ICUs at 10 different Spanish hospitals, 6 of which include coronary patients. The total number of patients was 2501, and they were described using the set of variables of APACHE III detailed in section 3. In order to be handled by SVM, we codified each discrete variable using as many new binary variables (with values 0 and 1) as the number of possible values of the original variable, only setting to '1' the variable corresponding to the discrete value actually taken by the original variable.

First of all, we report [Luaces *et al.*, 2007] a comparison of the SVM methods discussed in section 2 and APACHE III. We used the customization of APACHE III developed to improve its performance in Spanish ICUs [Rivera-Fernández *et al.*, 1998]. Notice that this is an unfair comparison since APACHE III was trained with a cohort of 17440 patients from 40 different hospitals in the USA [Knaus *et al.*, 1991]; the Spanish version used records of 10929 patients from 86 ICUs; while the available data sets in our experiments only included 2501 patients. Nevertheless, this comparison is useful to test whether or not the scores achieved by SVM methods are good enough to draw some knowledge about the weight of different groups of variables in different clinical contexts.

It is important to recall that the AUC achieved by the Spanish version of APACHE III in our experiments, 82.27% (in percentage) is similar to the amount reported by Rivera-Fernández *et al.* [1998]: 81.82%. This fact supports the representativeness of the sample of critically ill patients considered in the experiments described here.

Table 2 shows the scores obtained. The data were organized in 13 different training sets, one for each Unit, two

# patients	Unit	SVM(AUC) Bs·10^2	SVM(AUC) AUC ·10^2	SVM(Ac) Bs·10^2	SVM(Ac) AUC ·10^2	APACHE III Bs·10^2	APACHE III AUC ·10^2
108	1	17.12	75.82	18.60	70.60	14.73	81.76
189	2	18.87	73.51	19.98	69.23	17.10	77.80
194	3	17.35	75.32	18.97	65.88	15.92	78.20
194	4	10.89	77.20	11.42	74.93	9.61	86.17
195	5	11.02	84.44	10.94	82.41	10.79	88.78
239	6	15.69	74.87	16.37	69.12	14.59	77.62
269	7	9.93	81.09	10.96	75.75	8.52	88.02
297	8	12.05	84.86	12.77	81.44	11.27	87.37
337	9	10.96	81.35	11.28	77.91	10.71	81.30
479	10	10.71	79.32	11.20	71.74	12.18	78.22
919	{2,3,6,8}	14.94	79.75	15.00	78.46	14.32	80.86
1582	{1,4,5,7,9,10}	10.86	81.79	11.08	80.37	10.94	82.63
2501	all	12.34	81.51	12.29	81.22	12.18	82.27

Table 2: Performance of survival predictions by Units ordered by the number of patients (*# patients*). The learning methods compared are the commercial *APACHE III*, and SVM followed by Platt's algorithm to transform the output into a probability. We have tested two SVM versions, *SVM(Ac)*, which tries to optimize the classification accuracy, and *SVM(AUC)* that optimizes first the Area Under the ROC Curve. In the SVM cases we used a 10-fold cross validation to estimate both the Brier score (*Bs*) and *AUC*

collecting the data from non-coronary/coronary ICUs respectively, and the last one containing all the data. We observe that the version that optimizes the AUC first (SVM(AUC)) outperforms (lower Brier score and higher AUC) the version that optimizes first the classification accuracy (SVM(Ac)). However, both SVM methods do not reach the results of the commercial APACHE III, although the differences decrease as the number of training cases increases.

These results allow us to conclude that the method SVM(AUC) is good enough so as to estimate the weight of groups of variables in different medical contexts.

4.2 Groups of Intensive Care Units

From a medical point of view, probably the most obvious division between ICUs can be stated into those that include or not coronary patients. It is acknowledged that coronary diseases generally have a lower mortality risk than other critical illnesses. Moreover, there are important differences in the treatments applied to patients in both kinds of ICUs. Therefore, we decided to consider if there are also differences in the hypothesis that predict the probabilities of survival.

Table 3 shows the scores achieved by SVM(AUC) in these kinds of Units considering different groups of variables defined in section 3. To contrast the results, we included three datasets: coronary, non-coronary, and all units. On the other hand, the groups of variables used were: all variables, clinical, and clinical plus the other two groups defined in section 3, laboratory and monitoring. We included clinical in addition to these two specialized groups since clinical variables constitute somehow the basic information about a patient that it is routinely recorded.

First of all, we observe that the basic clinical variables provide surprisingly good results. So, in AUC the differences with the whole set of variables are just around 3 points down, while in Brier score the gap is 0.7 when the units of these scores are multiplied by 10^2.

When we add monitoring or laboratory variables to the clinical ones, we almost reach the maximum predictive capacity. In the dataset of patients from all Units, the differences are inappreciable. But in ICUs with coronary patients, monitoring is more useful for a prediction task than laboratory variables. On the other hand, we have the opposite situation in Units without coronary patients. These results are consistent when we measure the performance with AUCs or Brier scores.

4.3 Groups of Patients

The second context where we studied the differences of weight of variable groups arose from considering the treatment location of patients immediately prior to ICU admission. Table 4 reports the number and percentages of patients for each location in the whole dataset of 2501 patients and the percentage of dead. Notice that survival percentages are dramatically different, thus we used the situations with higher dead in order to deepen our knowledge about the weight of variables.

Therefore, we now consider 3 contexts to study the performance of the groups of variables as we did in the experiments reported in the preceding subsection. Table 5 gathers the results so obtained. We observe, that in the case of patients that come from a different hospital, no matter if they come from ICU, ward, or any other location of the other hospital, laboratory data are more predictive than monitoring. On the other hand, for patients who come from a ward of the same hospital, to predict their survival probabilities, monitoring devices are more useful than laboratory data; in fact, it is preferable to get rid of laboratory data that in this case act as a noise source.

The third situation considered is that of patients whose admission in an ICU is after an urgent surgery. Probably this is a too broad circumstance that includes too many different cases; in fact, we observe that in this case monitoring and laboratory variables weight the same.

	All ICU Units		Coronary Units		Non-Coronary Units	
	Bs·10^2	AUC ·10^2	Bs·10^2	AUC ·10^2	Bs·10^2	AUC ·10^2
All var.	12.34	81.51	10.86	81.79	14.94	79.75
Clini.+Moni.	12.57	80.50	10.91	81.33	15.46	77.62
(Dif. all var.)	(-0.23)	(1.01)	(**-0.05**)	(**0.46**)	(-0.52)	(2.13)
Clini.+Lab.	12.57	80.54	11.33	79.79	15.05	79.49
(Dif. all var.)	(-0.24)	(0.97)	(-0.47)	(2.00)	(**-0.11**)	(**0.26**)
Clinical	12.94	78.80	11.44	78.33	15.78	76.54
(Dif. all var.)	(-0.61)	(2.71)	(-0.58)	(3.46)	(-0.84)	(3.21)

Table 3: Performance of survival predictions of SVM(AUC) by groups of ICU units and groups of variables using 10-fold cross validation. Small differences (in **bold**) both in *AUC* and Brier score (*Bs*) let us realize that monitoring has more relevancy than laboratory data in Units with coronary patients; in units without these patients, we have the opposite situation

	Other hospital	Ward	Scheduled surgery	Urgent surgery	Urgencies	Totals
% pat.	7.16%	20.03%	15.11%	9.08%	48.58%	100%
# pat.	179	501	378	227	1216	2501
% dead	**27.93**%	**36.53**%	7.94%	**25.11**%	14.49%	19.83%

Table 4: Distribution of patients according to the treatment location immediately prior to their ICU admission. In some cases (in **bold**) the percentage of dead is significantly high

4.4 Discussion

We have described two prediction contexts where laboratory data is more useful than monitoring in order to predict survival: Units without coronary patients, and patients coming from other hospitals. In both cases the dead risk of patients is usually related to multi-organic (respiratory, renal or hepatic) failures. The medical way to control the evolution of these diseases is by means of laboratory data, what explains the results obtained.

On the other hand, monitoring is more useful than laboratory data for patients coming from a ward of the same hospital of the ICU considered, or for Units with coronary patients. In these cases, survival is mostly threatened by cardiovascular complications, and they are controlled by means of monitoring devices.

5 Conclusions

We have presented a reliable method to estimate hospital survival probability of critically ill patients. The method is based on SVM aimed at optimizing the classification AUC followed by Platt's scaling algorithm [2000].

Using this tool, we have identified some medical contexts where the weights of monitoring and laboratory variables have meaningful differences. These results have clear medical explanations, but the novelty is that now weight differences can be measured in a precise sense.

Additionally, we have established that most of the prediction capability of SVM models can be reached by a group of basic clinical variables. This group is formed by demographic and diagnostic data adding simple tests or observations that are routinely recorded for ICU admissions.

From a practical point of view, the implication of the research reported here is that it is possible to tailor cheap and reliable prediction systems according to the peculiarities of ICUs and kinds of patients.

Acknowledgments

The research reported here is supported in part under grant TIN2005-08288 from the MEC (Ministerio de Educación y Ciencia of Spain).

References

[Brier, 1950] G.W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Rev*, 78:1–3, 1950.

[Hanley and McNeil, 1982] J.A. Hanley and B.J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[Joachims, 2005] T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the ICML '05*, 2005.

[Joachims, 2006] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2006.

[Knaus *et al.*, 1991] W.A. Knaus, E.A. Draper, D.P. Wagner, J.E. Zimmerman, M. Bergner, P.G. Bastos, C.A. Sirio, D.J. Murphy, T. Lotring, and A. Damiano. The APACHE III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. *Chest*, 100:1619–1636, 1991.

[Le Gall *et al.*, 1984] J.R. Le Gall, P. Loirat, A. Alperovitch, P. Glaser, C. Granthil, D. Mathieu, P. Mercier,

	Other hospital		Wards		Urgent surgery	
	Bs·10^2	AUC ·10^2	Bs·10^2	AUC ·10^2	Bs·10^2	AUC ·10^2
All var.	17.93	73.72	18.37	77.66	15.81	77.30
Clini.+Moni.	18.17	70.74	18.13	78.26	16.52	74.74
(Dif. all var.)	(-0.24)	(2.98)	(**0.24**)	(**-0.60**)	(-0.71)	(2.56)
Clini.+Lab.	18.09	74.60	18.89	76.29	16.16	74.91
(Dif. all var.)	(**-0.16**)	(**-0.88**)	(-0.52)	(1.37)	(-0.35)	(2.38)
Clinical	18.27	70.39	18.98	75.93	17.16	71.32
(Dif. all var.)	(-0.34)	(3.33)	(-0.61)	(1.73)	(-1.35)	(5.98)

Table 5: Performance of survival predictions by groups of patients and groups of variables using 10-fold cross validation. Small differences in the prediction error (in **bold**) indicate that laboratory data are more relevant than monitoring in patients from other hospitals, but in patients from wards monitoring scores are more relevant than laboratory data

R. Thomas, and D. Villers. A simplified acute physiology score for ICU patients. *Crit Care Med.*, 12:975–977, 1984.

[Lemeshow *et al.*, 1993] S. Lemeshow, D. Teres, J. Klar, J. S. Avrunin, S. H. Gehlbach, and J. Rapoport. Mortality probability models (MPM II) based on an international cohort of intensive care unit patients. *Journal of the American Medical Association*, 270(20):2478–2486, November 1993.

[Luaces *et al.*, 2007] O. Luaces, J.R. Quevedo, F. Taboada, G.M. Albaiceta, and A. Bahamonde. Prediction of probability of survival in critically ill patients optimizing the Area Under the ROC Curve. In *Proceedings of the IJCAI '07*, pages 956–961, 2007.

[Platt, 2000] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A.J. Smola, P.L. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[Provonost and Angus, 2001] P. Provonost and D.C. Angus. Economics of end-life-care in the intensive care unit. *Critical Care Med*, 29(Suppl):46–51, 2001.

[Rivera-Fernández *et al.*, 1998] R. Rivera-Fernández, G. Vázquez-Mata, M. Bravo, E. Aguayo-Hoyos, J. Zimmerman, D. Wagner, and W. Knaus. The APACHE III prognostic system: customized mortality predictions for Spanish ICU patients. *Intensive Care Medicine*, 24(6):574–581, June 1998.

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

Using Kernel Based Classifiers for Reliable Predictions in Medical Diagnostics

Matjaž Kukar[1], **Dimitris Tzikas**[2], **Aristidis Likas**[2], **Luka Šajn**[1]
[1]Faculty of Computer and Information Science, University of Ljubljana, Slovenia
[2]Department of Computer Science, University of Ioannina, Greece
matjaz.kukar@fri.uni-lj.si, tzikas@cs.uoi.gr, arly@cs.uoi.gr, luka.sajn@fri.uni-lj.si

Abstract

Estimating the reliability of individual predictions or classifications is very important in several applications such as medical diagnosis. Recently, the transductive approach to reliability estimation has been proved to be very efficient when used with several machine learning classifiers, such as Naive Bayes and decision trees. However, the efficiency of the transductive approach for state-of-the art kernel-based classifiers was not considered. In this work we deal with this problem and apply the transductive reliability methodology with sparse kernel classifiers, specifically the Support Vector Machine and Relevance Vector Machine. Experiments with medical and bioinformatics datasets demonstrate better performance of the transductive approach for reliability estimation compared to reliability measures obtained directly from the output of the classifiers. Furthermore, we apply the methodology in the problem of reliable diagnostics of the coronary artery disease, outperforming the expert physicians' standard approach.

1 Introduction

Decision-making is a complicated process that carries a certain amount of imperfectness and therefore cannot be considered completely reliable. However, it is often crucial to know the magnitude of diagnosis' (un)reliability in order to minimize risks, for example in the medical domain risks related to the patient's health or even life. One of the reasons why machine learning methods are infrequently used in practice is that they fail to provide an unbiased reliability measure of predictions.

Although there are several methods for estimating the overall performance of a classifier, e.g cross-validation, there is very little work on estimating the reliability of individual classifications. The transductive reliability methodology as introduced in [Kukar and Kononenko, 2002] computes the reliability of an individual classification, by studying the stability of the trained model when the training set is perturbed (the newly classified example is added to the training set and the classifier is retrained). For reliable classifications, this process should not lead to significant model changes. The transductive reliability methodology has been applied on traditional classifiers like Naive Bayes and decision trees with interesting results. Here, we examine the effectiveness of this methodology when applied on sparse kernel-based classifiers, such as the Support Vector Machine (SVM) and the Relevance Vector Machine (RVM), and compare transductive reliability estimations with reliability measures based on the outputs that SVM and RVM provide. SVMs and RVMs produce state-of-the-art classifiers that are used in a wide variety of problems. The aim of this paper is to evaluate the advantages of using the transduction principle to assess the reliability of individual classifications made with SVM/RVM. We also apply the methodology for diagnosis of the coronary artery disease (CAD) using kernel-based classifiers and compare our results to the performance of expert physicians using an established standard methodology.

2 Transductive Reliability Estimations

There have been numerous attempts to assign probabilities to machine learning classifiers' (decision trees and rules, Bayesian classifiers, neural networks, nearest neighbour classifiers, ...) in order to interpret their decision as a probability distribution over all possible classes. In fact, we can trivially convert every machine learning classifier's output to a probability distribution by assigning the predicted class the probability 1, and 0 to all other possible classes. The posterior probability of the predicted class can be viewed as a classifier's confidence (reliability) of its prediction. However, such estimations may in general not be trustworthy due to inherent applied algorithm's biases.[1] Reliability estimation of a classification ($\widetilde{y}$) of a single example (x), given its true class (y) should have the following property:

$$\mathrm{Rel}(\widetilde{y} \mid x) = t \;\Rightarrow\; P(\widetilde{y} \neq y) \leq 1 - t \qquad (1)$$

If Eq. 1 holds, or even better, if it approaches equality, a reliability measure can be treated as a confidence value [Melluish *et al.*, 2001].

Transduction is an inference principle that takes a training sample and aims at estimating the values of a discrete or continuous function only at given unlabelled points of interest from input space, as opposed to the whole input space

[1]An extreme case of inherent bias can be found in a trivial constant classifier that blindly labels any example with a predetermined class with self-proclaimed confidence 1.

for induction. In the learning process the unlabelled points are suitably labelled and included into the training sample. The usefullness of unlabelled data has also been advocated in the context of co-training. It has been shown [Blum and Mitchell, 1998] that for every better-than-random classifier its performance can be significantly improved by utilizing only additional unlabelled data.

The transductive reliability estimation process and its theoretical foundations originating from Kolmogorov complexity are described in more detail in [Kukar and Kononenko, 2002].

In practice, transductive reliability estimation is performed in a two-step process, featuring an *inductive step* followed by a *transductive step*.

- An *inductive step* is just like an ordinary inductive learning process in Machine Learning. A Machine Learning algorithm is run on the training set, *inducing* a classifier. A selected example is taken from an independent dataset and classified using the induced classifier. An example, labelled with the assigned class is temporarily included into the training set (Figure 1a).
- A *transductive step* is almost a repetition of an inductive step. A Machine Learning algorithm is run on the changed training set, *transducing* a classifier. The same example as before is taken from the independent dataset and is classified using the transduced classifier (Figure 1b). Both classifications of the same example are compared and their difference (distance) is calculated, thus approximating the randomness deficiency.
- After the reliability is calculated, the example in question is removed from the training set.

The machine learning algorithm, whose reliability is being assessed, is assumed to provide a probability distribution p that describes the probability that its input belongs at each possible class. In order to measure how much the model changes, we calculate the distance between the probability distribution p of the initial classifier and the probability distribution q of the augmented classifier, using the Symmetric Kullback-Leibler divergence, or J-divergence, which is defined as

$$J(p,q) = \sum_{i=1}^{n} (p_i - q_i) \log_2 \frac{p_i}{q_i}. \qquad (2)$$

$J(p,q)$ is limited to the interval $[0,\infty]$, with $J(p,p) = 0$. For the ease of interpretation, it is desirable for reliability values to be bounded to the $[0,1]$ interval, $J(p,q)$ is normalized in the spirit of Martin-Löf's test for randomness [Kukar and Kononenko, 2002; Li and Vitányi, 1997, pp. 129], to obtain the transductive reliability measure (TRE) used in our approach:

$$TRE = 1 - 2^{-J(p,q)}. \qquad (3)$$

Due to non-optimal classifiers resulting from learning in noisy and incomplete datasets, it is inappropriate to select *a priori* fixed boundary (say, 0.90) as a threshold above which a classification is considered reliable. To deal with this problem, we split the range $[0,1]$ of reliability estimation values into two intervals by selecting a threshold T. The lower interval $[0,T)$ contains unreliable classifications, while the higher interval $[T,1]$ contains reliable classifications. As a splitting point selection criterion we use maximization of the information gain[Dougherty *et al.*, 1995]:

$$Gain = H(S) - \frac{|S_1|}{|S|} H(S_1) - \frac{|S_2|}{|S|} H(S_2), \qquad (4)$$

where $H(S)$ denotes the entropy of set S, $S_1 = \{x : TRE(x) < T\}$ is the set of unreliable examples and $S_2 = \{x : TRE(x) > T\}$ is the set of reliable results.

Note that our approach is considerably different from that described in [Gammerman *et al.*, 1998; Saunders *et al.*, 1999]. Their approach is tailor-made for SVM (it works by manipulating support vectors) while ours requires only that the applied classifier provide a probability distribution.

3 Kernel-Based Methods

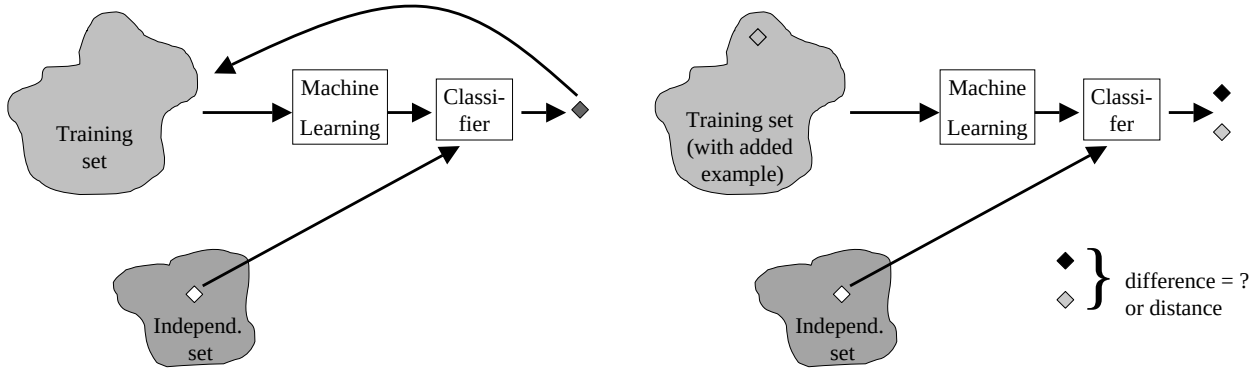
Kernel methods have been extensively used to solve classification problems, where a training set $\{x_n, t_n\}_{n=1}^{N}$is given, so that t_n is the target for training example x_n. The targets t_n are discrete, e.g. $t \in \{0,1\}$ for binary classification, and they describe the class where each training example belongs.

Kernel methods, are based on a mapping function $\phi(x)$ that maps each training vector to a higher dimensional feature space. Then, inner products between training examples are computed in this new feature space, by evaluating the corresponding kernel function $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. This kernel function, provides a measure of similarity between training examples. This similarity measure, is much more flexible than the inner product $x_i x_j$, however it introduces the additional task of selecting an appropriate mapping function ϕ. In practice, learning algorithms do not require computation of the mapping function ϕ, but only inner products which are evaluated by the corresponding kernel function $K(x_i, x_j)$.

Recently, there is much interest in sparse kernel methods, such as the Support Vector Machine (SVM) and the Relevance Vecor Machine (RVM). These methods are called sparse because, after training with the full dataset, they make predictions using only a small subset of the available training vectors. These training vectors, which are used for predictions are called support vectors (SV) in the case of SVM and relevance vectors (RV) in the case of RVM. The main reason why sparse kernel methods are so interesting and effective, is that during training, they automatically estimate the complexity of the dataset, and thus they have good generalization performance on both simple and complex datasets. In simple datasets only few support/relevance vectors will be used, while in more difficult datasets the number of support/relevance vectors will increase. Furthermore, making predictions using only a small subset of the initial training examples is typically much more computationally efficient.

3.1 Support Vector Machine

The support vector machine (SVM) classifier, is a kernel classifier that aims at finding an optimal hyperplane which separates data points of two classes. This hyperplane is optimal in the sense that it maximizes the margin between the



(a) Inductive step.

(b) Transductive step.

Figure 1: Transductive reliability estimation.

hyperplane and the training examples. The SVM classifier [Cortes and Vapnik, 1995] makes decisions for an unknown input vector, based on the sign of the decision function:

$$y_{SVM}(x) = \sum_{n=1}^{N} w_n K(x, x_n) + b \tag{5}$$

After training, most of the weights w are set to exactly zero, thus predictions are made using only few of the training vectors, which are the support vectors.

Assuming that the two classes are labeled with '-1' and '1', so that $t_n \in \{-1, 1\}$, the weights $w = (w_1, \ldots, w_N)$ are set by solving the following quadratic programming problem:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w \\ \text{subject to} \quad & t_n(w^T \phi(x_n) + b) \geq 1 \end{aligned} \tag{6}$$

The above formulation can only be applied in cases where the data points are separable. However, it can be extended in order to treat non-separable cases, by introducing the auxiliary variables $\xi = (\xi_1, \ldots, \xi_N)$:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \\ \text{subject to} \quad & t_n(w^T \phi(x_n) + b) \geq 1 - \xi_n \\ & \xi_n \geq 0 \end{aligned} \tag{7}$$

SVM makes predictions based on the decision function of eq. (5). Positive values of the decision function ($y_{SVM}(x) > 0$) correspond to class '1', while negative values ($y_{SVM}(x) < 0$) correspond to class '-1'. Furthermore, the absolute value of the decision function provides a measure of the certainty of the decision. Values near zero, correspond to points near the decision boundary and therefore may be unreliable, while large values of the decision function should correspond to reliable classifications. In practice, we can obtain a reliability measure, scaled in $[0, 1]$, by using the sigmoid function $\sigma(x) = 1/(1+\exp(x))$:

$$RE_{SVM} = |2\sigma(y_{SVM}(x)) - 1| \tag{8}$$

3.2 Relevance Vector Machine

The relevance vector machine (RVM) classifier [Tipping, 2001], is a probabilistic extension of the linear regression model, which provides sparse solutions. It is analogous to the SVM, since it computes the decision boundary using only few of the training examples, which are now called relevance vectors. However training is based on different objectives.

The RVM model $y(x; w)$ is the output of a linear model with parameters $w = (w_1, \ldots, w_N)^T$, with application of a sigmoid function for the case of classification:

$$y_{RVM}(x) = \sigma(\sum_{n=1}^{N} w_n K(x, x_n)), \tag{9}$$

where $\sigma(x) = 1/(1 + \exp(-x))$. In the RVM, sparseness is achieved by assuming a suitable prior distribution on the weights, specifically a zero-mean, Gaussian distribution with distinct inverse variance α_n for each weight w_n:

$$p(w|\alpha) = \prod_{n=1}^{N} N(w_n|0, \alpha_n^{-1}). \tag{10}$$

The variance hyperparameters $\alpha = (\alpha_1, \ldots, \alpha_N)$ are assumed to be Gamma distributed random variables:

$$p(\alpha) = \prod_{n=1}^{N} Gamma(\alpha_n|a, b). \tag{11}$$

The parameters a and b are assumed fixed and usually they are set to zero ($a = b = 0$), which provides sparse solutions.

Given a training set $\{x_n, t_n\}_{n=1}^{N}$ with $t_n \in \{0, 1\}$ training in RVM is equivalent to compute the posterior distribution $p(w, \alpha|t)$. However, since this computation is intractable, a quadratic approximation $\log p(w|t, \alpha) \approx (w - \mu)^T \Sigma^{-1} (w - \mu)$ is assumed and we compute matrix Σ and vector μ as:

$$\Sigma = (\Phi^T B \Phi + A)^{-1} \tag{12}$$

$$\mu = \Sigma \Phi^T B \hat{t} \tag{13}$$

with the $N \times N$ matrix Φ defined as $[\Phi]_{ij} = K(x_i, x_j)$, $A = \text{diag}(\alpha_1, \ldots, \alpha_N)$, $B = \text{diag}(\beta_1, \ldots, \beta_N)$, $\beta_n = y_{RVM}(x_n)[1 - y_{RVM}(x_n)]$ and $\hat{t} = \Phi\mu + B^{-1}(t - y)$.

The parameters α are then set to the values α_{MP} that maximize the logarithm of the following marginal likelihood

$$L(\alpha) = \log p(\alpha|t) = -\frac{1}{2}\left[N \log 2\pi + log|C| + t^T C^{-1} t\right], \quad (14)$$

with $C = B^{-1} + \Phi A^{-1} \Phi^T$. This, gives the following update formula:

$$\alpha_n = \frac{1 - \alpha_n \Sigma_{nn}}{\mu_n^2} \quad (15)$$

The RVM learning algorithm iteratively evaluates formulas (12),(13) and (15).

After training, the value of $y_{RVM}(x) = y(x; \mu)$ can be used to estimate the reliability of the classification decision for input x. Values close to 0.5 are near the decision boundary and therefore are unreliable classifications, while values near 0 and near 1 should correspond to reliable classifications. In our experiments, we used the reliability measure

$$RE_{RVM} = |2y_{RVM}(x) - 1|, \quad (16)$$

which takes values near 0 for unreliable classifications and near 1 for reliable classifications.

3.3 Incremental Relevance Vector Machine

An interesting property of the RVM model that can be exploited in the transductive approach, is that it can be trained incrementally, as proposed in [Tipping and Faul, 2003]. The proposed incremental algorithm, initially assumes an empty model, that does not use any basis functions. Then, it incrementally adds, deletes and re-estimates basis functions, until convergence. It is based on the observation that the marginal likelihood, see eq. (14), can be decomposed as:

$$L(\alpha) = L(\alpha_{-n}) + l(\alpha_n), \quad (17)$$

where $L(\alpha_{-n})$ does not depend on α_n and

$$l(\alpha_n) = \log \alpha_n - \log(\alpha_n + s_n) + \frac{q_n^2}{\alpha_n + s_n}, \quad (18)$$

with $s_n = \phi_n^T C_{-n}^{-1} \phi_n$ and $q_n = \phi_n^T C_{-n}^{-1} \hat{t}$. Here, $C_{-n} = B^{-1} + \sum_{i \neq n} \alpha_n^{-1} \phi_n \phi_n^T$ denotes the matrix C without the contribution of the n-th basis function, so that $C = C_{-n} + \alpha_n^{-1} \phi_n \phi_n^T$, s_n is the "sparseness" factor that measures how sparse the model is and q_n is the "quality" factor that measures how well the model fits the observations. Based on this decomposition, analysis of $l(\alpha_n)$ shows that it is maximized when

$$\alpha_n = \frac{s_n^2}{q_n^2 - s_n} \quad \text{if } q_n^2 > s_n \quad (19)$$

$$\alpha_n = \infty \quad \text{if } q_n^2 \leq s_n \quad (20)$$

Based on this result, the following algorithm is proposed in [Tipping and Faul, 2003]:

1. Initially assume an empty model, set $a_n = \infty$, for all n
2. Select a training point x_n and compute the corresponding basis function ϕ_n as well as s_n and q_n.
   (a) if $q_n^2 > s_n$ and $\alpha_n = \infty$ add the basis function to the model, using eq. (19) to set α_n
   (b) if $q_n^2 > s_n$ and $\alpha_n < \infty$ re-estimate α_n
   (c) if $q_n^2 <= s_n$ remove the basis function from the model, set $\alpha_n = \infty$
3. Compute Σ and μ, using eq. (12) and (13)
4. Repeat from step 2, until convergence.

4 Experimental Evaluation of Transductive Reliability Estimations in Medical and Biomedical Problems

In this section, we apply the transductive reliability methodology in a series of classification problems and compare the performance of transductive reliability estimations, with respect to the reliability measures that are directly computed based on SVM and RVM outputs.

Tranductive reliability estimations, are obtained following the procedure described in Section 2. After training the model and computing its output for a new test point x_*, we add this test point to the training set with the predicted label and retrain the model. Transductive reliability estimations are obtained by measuring the distance between the output distributions of the two models.

In the case of RVM we also considered a modification, where we used the incremental training algorithm to obtain fast transductive reliability estimations. Specifically, after adding the new training point x_*, instead of retraining from scratch, we can use the incremental algorithm to continue training the previous model. This is much more computationally efficient, and in the experiments it appears to provide better performance than the standard approach of training from scratch.

In order to evaluate the performance of the reliability estimation methods, we apply the following procedure. We perform leave-one-out cross-validation on the available training dataset and compute a prediction for the class of each training point and a reliability estimation (RE) of this prediction. Afterwards, we can discriminate reliable and unreliable classifications by selecting a threshold (T) for the reliability measure. Using an ideal reliability measure all correct classifications should be labeled reliable ($RE > T$), while all incorrect classifications should be labeled unreliable. Thus, an evaluation of the reliability measure is obtained by computing the percentage of correct and reliable classifications, and the percentage of incorrect reliable classifications. Plotting these percentages, for many values of the threshold, produces an ROC curve, which illustrates the performance of the reliability estimation method.

Allthough the ROC describes the overall efectiveness of a reliability measure, in practice, a single threshold value has to be used. This is selected by maximizing the information gain, as explained in Section 2. The information gain may also be used to compare the performance of several reliability measures. Table 1 shows the information gain that is achieved by: i) using directly the SVM/RVM reliability estimates RE_{SVM} and RE_{RVM}, ii) using the transduction reliability principle (TRE). Results are shown for two medical datasets from the UCI machine learning repository and the leukemia bioinformatics dataset. It is clear

Method	hepatitis	new-thyroid	leukemia
RE_{SVM}	0.106	0.083	0.054
TRE_{SVM}	0.120	0.092	0.073
RE_{RVM}	0.109	0.068	0.089
TRE_{RVM}	0.178	0.062	0.062
$TRE_{RVM(inc)}$	0.133	0.072	0.107

Table 1: Information gain of SVM/RVM reliability estimations and transductive reliability estimations.

that when SVM is used, transduction provides better information gain for all datasets. The same happens with incremental RVM, while when typical RVM is used, transduction is better in two of the three cases.

4.1 Diagnosis of Coronary Artery Disease

Coronary artery disease (CAD) is the most important cause of mortality in all developed countries. It is caused by diminished blood flow through coronary arteries due to stenosis or occlusion. CAD produces impaired function of the heart and finally the necrosis of the myocardium – myocardial infarction.

In our study we used a dataset of 327 patients (250 males, 77 females) with performed clinical and laboratory examinations, exercise ECG, myocardial scintigraphy and coronary angiography because of suspected CAD. The features from the ECG an scintigraphy data were extracted manually by the clinicians. In 228 cases the disease was angiographically confirmed and in 99 cases it was excluded. 162 patients had suffered from recent myocardial infarction. The patients were selected from a population of approximately 4000 patients who were examined at the Nuclear Medicine Department, University Clinical Centre, Ljubljana, Slovenia, between 1991 and 1994. We selected only the patients with complete diagnostic procedures (all four levels) [Kukar *et al.*, 1999].

Physicians apply a stepwise diagnostic process and use Bayes law to compute a posterior probability of disease, based on some diagnostic tests and a prior probability according to the age, gender and type of chest pain for each patient. Reliable diagnoses are assumed to be those whose posterior probability is over 0.90 (positive) or under 0.10 (negative). We considered treating the problem by training an SVM or an RVM classifier and using the transductive reliability principle to estimate the reliability of each classification. For evaluation purposes, we performed leave-one-out cross-validation, and for each example we predicted a class and a reliability of the classification. We then splitted classifications to reliable and unreliable by computing the threshold that maximizes the information gain and measured the percentage of reliable diagnoses (with the reliability measure above some threshold), and errors made in this process (percentage of incorrectly diagnosed patients with seemingly reliable diagnoses). The results are shown in Table 2, where it can be observed that when the transduction principle is used along with SVM and incremental RVM, the achieved reliability estimations performance is better compared to physicians. Furthermore, in Figure 2 ROC curves are plotted separately for the cases of positive and negative examples.

5 Conclusions

We applied the transduction methodology for reliability estimation on sparse kernel-based classification methods. Experiments on medical datasets from the UCI repository and a bioinformatics gene expression dataset, indicate that, when used with kernel-based classifiers, transductive reliability estimations are more accurate than simple reliability measures based on the outputs of kernel classifiers. While no experimental comparison with the specific approach of Gammerman *et al.* [1998] was made due to different implementation (our experiments were performed within Matlab), experiments with other classifiers [Kukar, 2004; 2006] show that our methodology, while completely generic, performs similarly to tailor-made transductive methods.

We also applied the transductive methodology in the problem of CAD diagnosis, achieving better reliability estimation performance compared to the standard physicians procedure.

Acknowledgements

This work was supported in the framework of the "Bilateral S+T cooperation between the Hellenic Republic and the Republic of Slovenia (2004-2006)".

References

[Blum and Mitchell, 1998] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[Dougherty *et al.*, 1995] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proc. ICML'95*, pages 194–202. Morgan Kaufmann, 1995.

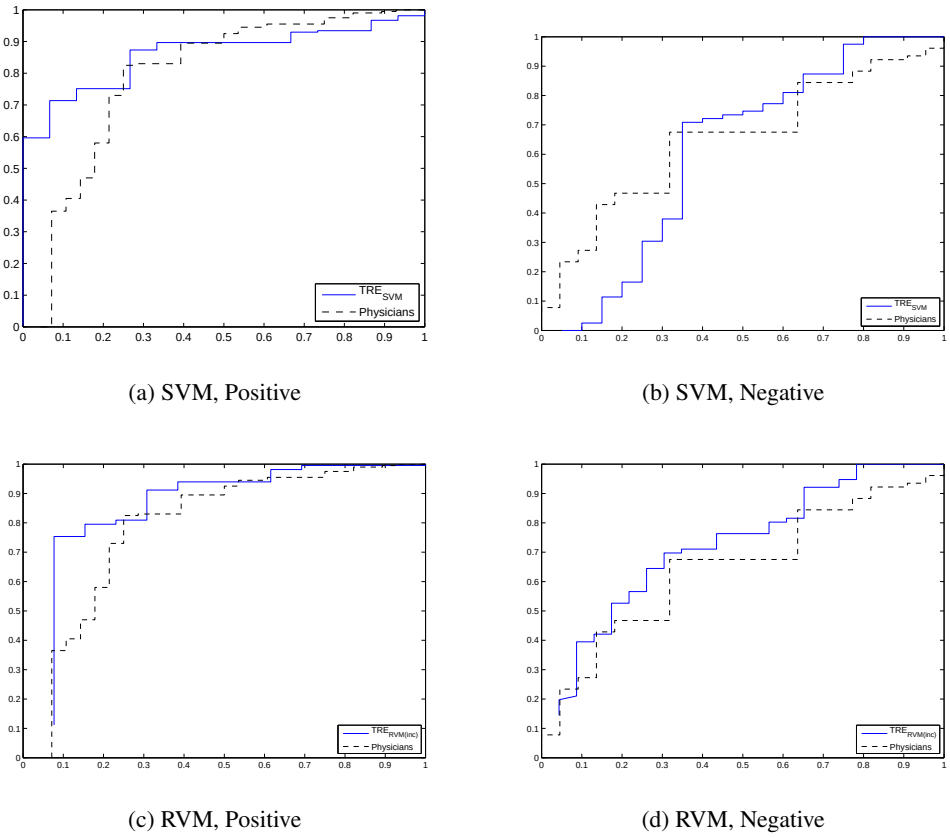
[Gammerman *et al.*, 1998] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the* 14^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 148–155, Madison, Wisconsin, 1998.

[Kukar and Kononenko, 2002] M. Kukar and I. Kononenko. Reliable classifications with Machine Learning. In *Proceedings of* 13^{th} *European Conference on Machine Learning, ECML 2002*, pages 219–231, 2002.

[Kukar *et al.*, 1999] M. Kukar, I. Kononenko, C. Grošelj, K. Kralj, and J. Fettich. Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine: Special Issue on Data Mining Techniques and Applications in Medicine*, 1999. In press.

Method	Positive			Negative		
	Reliable	Errors	AUC	Reliable	Errors	AUC
Physicians	164	9	0.790	45	7	0.650
RE_{SVM}	148	0	0.903	30	4	0.566
TRE_{SVM}	173	4	0.861	56	8	0.672
RE_{RVM}	144	1	0.842	53	6	0.729
TRE_{RVM}	152	3	0.767	49	5	0.702
$TRE_{RVM(inc)}$	158	1	0.850	53	7	0.720

Table 2: Comparison of the performance of expert physicians and machine learning classification methods for the CAD dataset.



(a) SVM, Positive

(b) SVM, Negative

(c) RVM, Positive

(d) RVM, Negative

Figure 2: ROC curves for the transduction reliability measures for SVM and incremental RVM, using the CAD dataset and considering separately the positive and negative examples. For positive patients, each ROC curve depicts a ratio between reliable true positives on y-axis and reliable false positives (errors) on x-axis. For negative patients, each ROC curve depicts a ratio between reliable true negatives on y-axis and reliable false negatives (errors) on x-axis.

[Kukar, 2004] M. Kukar. Transduction and typicalness for quality assessment of individual classifications in machine learning and data mining. In R. Rastogi, editor, *Proc. International Conference in Data Mining 2004*, pages 146–153, Los Alamitos (California), 2004. IEEE Computer Society.

[Kukar, 2006] M. Kukar. Quality assessment of individual classifications in machine learning and data mining. *Knowledge and Information Systems*, 9(3):364–384, 2006.

[Li and Vitányi, 1997] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, New York, 2^{nd} edition, 1997.

[Melluish *et al.*, 2001] T. Melluish, C. Saunders, I. Nouretdinov, and V. Vovk. Comparing the Bayes and typicalness frameworks. In *Proc. ECML 2001*, volume 2167, pages 350–357, 2001.

[Saunders *et al.*, 1999] C. Saunders, A. Gammerman, and V. Vovk. Transduction with confidence and credibility. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

[Tipping and Faul, 2003] M. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In *Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[Tipping, 2001] Michael E. Tipping. Sparse Bayesian learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

Modeling Genetic Susceptibility: a case study in periodontitis

V.S. Moustakis[1,2], M.L. Laine[3], L. Koumakis[1], G. Potamias[1], L. Zampetakis[2], and B.G. Loos[3]
[1]Foundation for Research and Technology – Hellas (FORTH), Institute of Computer Science, Bioinformatics Laboratory, Science and Technology Park of Crete, Heraklion 71110, Greece
[2]Technical University of Crete, Department of Production Engineering and Management, Chania, Greece
[3]Academic Centre for Dentistry Amsterdam (ACTA), Department of Oral Microbiology and Department of Periodontology, The Netherlands
{moustaki,koumakis,potamias}@ics.forth.gr, ML.Laine@vumc.nl, lzabetak@dpem.tuc.gr, B.Loos@acta.nl

Abstract

We present a model to assess genetic susceptibility toward disease manifestation. The model is based on post-processing analysis of association rule mining results. The model is validated using periodontal disease records from the data warehouse of ACTA in the Netherlands combining phenotype information such as disease severity with microbial measurements, environmental factors or patient demographics. Validation is performed by incorporating or integrating genetic susceptibility index results with decision tree learning, structural equation modeling, and metrics on microbial values. Initial analysis demonstrates that the proposed index represents a reliable disease predictor. We present results and discuss the implications in bioinformatics research and practice.

1 Introduction

Assessment of genetic susceptibility toward disease manifestation represents a key endeavor in the integration of genotype and phenotype information. The search for genetic markers and candidate disease-modifying genes is receiving considerable attention. Single nucleotide polymorphisms (SNP's) in genes encoding molecules of the host defense system are assessed in addition to other records of the patients such as, environmental, clinical, and microbial measurements.

A patient record may incorporate heterogeneous information such as: SNP's, age, gender, patient history (i.e., smoking, ethnic origin), clinical and microbial measurements. Integration is not trivial since heterogeneity across data does not make them amenable to uniform treatment. SNP data are symbolic while the other aforementioned components may be symbolic or numeric. Therefore, the classical model of patient records used to learn disease (or therapeutic) patterns may fail to generate meaningful results since different types of data may require different algorithmic treatment during learning. Furthermore, genetic markers denote susceptibility toward disease manifestation and it would be useful to exploit the information hidden into them and to derive a genetic susceptibility index (GSI).

This article presents a GSI assessment model. The model is motivated by the work of Ackoff (1958) on behavioral communication and the specific variation elaborated by Moustakis (2006). In the work of Ackoff central concepts on which the theory is based are purposefulness and knowledge gain. Purposefulness implies that choice is available and the entity involved is capable of choice. Knowledge gain relates to the added value as result of choice, or, of learning as result of choice. The work of Moustakis focuses on a knowledge gain computation model, which is based on learning outcome.

In the present article we present a methodology, which enables the derivation of a GSI from SNP's. We use association rule mining (ARM) to form disease and healthy status patterns using genetic markers. In the sequel we derive weights for the genetic markers and combine the weights to assess: (a) genetic susceptibility indices for each marker, $GSI(SNP)$; and, (b) the susceptibility of each individual toward disease manifestation $GSI(\mathrm{Re}cord)$. In the sequel we demonstrate two indicative ways that exploit susceptibility indices in learning. We demonstrate methodology using the periodontitis data warehouse (PDW) from ACTA (2005). We conclude the article by discussing results and placing work reported herein in context with biomedical research. We stress that results drawn from our work are still in progress.

In the following sections we overview the characteristics of the periodontitis data (section 2), the methods that support $GSI(SNP)$ and $GSI(\mathrm{Re}cord)$ assessment (section 3), the results we obtained using the periodontitis data (section 4), and conclude the article, and discuss areas for further work on the subject (section 5).

2 Periodontitis case study

Periodontitis is a chronic inflammatory disease of the supporting tissues of the teeth. If left untreated, teeth may show exposed root surfaces, in conjunction with red, swollen gums that easily bleed. Periodontitis is clinically defined by deepened pockets (>4 mm) and loss of attachment.

Etiology of periodontitis is multifactorial and involves infectious components, environmental factors and genetic susceptibility.

The PDW developed by ACTA (2005) incorporates over 850 records of periodontitis and control patients. Records incorporate SNP's, which are represented via gene, locus and genotype triplets; sixty two triplets are recorded across all records. However, the number of SNP's recorded is not the same for each individual and they range from three up to thirteen. In addition, there are records of seven bacterial species (*Actinobacillus actinomycetemcomitans, Porphyromonas gingivalis, Prevotella indermedia, Tannerella forsythensis, Peptostreptococcus micros, Fusobacterium nucleatum* and *Campylobacter rectus*), ethnic origin (based on the origin of parents and of the grandparents) as well as age, gender, smoking status, periodontal status (pocket depth and attachment loss), and severity assessment (valued over a nominal scale: healthy=0, mild periodontitis=1, and severe periodontitis=2).

Results presented herein are limited to the individuals with Caucasian origin; the group is composed of 675 individuals; 314 healthy, 94 mild periodontitis and 268 severe periodontitis individuals. A detailed presentation of records and record statistics is beyond the scope of this article and thus we skip it; however, as need arises we will present details of the data in the sequel.

3. GSI assessment process

We launched an association rule inquiry using SNP and disease status. Disease status was aggregated into two groups: Healthy and Diseased. ARM was performed using the HealthObs software (Potamias et al, 2005) while the basic algorithm that was used is also presented by Agrawal and Srikant (1994), and Mannila et al (1994).

Formally ARM is defined as follows: let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items and let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \varnothing$. The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if, *c%* of transactions in D that contain X also contain Y. Confidence establishes significance of association rule. Rule $X \Rightarrow Y$ has support s in the transaction set D if, *s%* of transactions in D contains $X \cup Y$. Support measures usefulness of the association. Given a set of transactions D, ARM proceeds to discover associations, which exhibit support and confidence values higher than specific thresholds, specified by the user: minimum support and minimum confidence.

We limited rule generation to rules with confidence equal to 100%. An example of a rule that was derived is: *{CARD15_Locus_3020insC_Genotype_11 & TGF-beta_Locus_cod25_Genotype_11} THEN Diseased*. This specific rule has support equal to s_j *%*, which means that it covers s_j *%* of the diseased records.

Thus the rule incorporates a *knowledge gain*, which, however, must be split equally between the SNP's which participate in rule formation. The s_j *%* metric represents knowledge gain, which in the specific rule is split equally to the two SNP's that participate in rule formation. Thus each of the participating SNP's receives a *credit*, which is equal to: $GSI(SNP; a_i) = s_i / 2$, where a_i is a rule counter.

Each SNP receives two types of credit: the first type associates with rules that link with diseased status and the second type associates with rules that are linked with healthy status. The overall disease-linked credit of a SNP is assessed by summing across all $GSI(SNP; a_i)$ values, namely: $GSI(SNP; D) = \sum GSI(SNP; a_i)$. Similarly, for the same SNP a GSI that relates to healthy status $GSI(SNP; H)$ is calculated. Then the overall GSI for each SNP is assessed by combining the two polar GSI values. We form a ratio in which the numerator corresponds to $GSI(D; SNP) - GSI(H; SNP)$ and the denominator to the addition of the two values: $GSI(D; SNP) + GSI(H; SNP)$. The result is denoted by $GSI(SNP)$ and based on the aforementioned definition $-1 \le GSI(SNP) \le +1$.

Following the definition of the GSI for each SNP we proceed to the assessment of the GSI with respect to each record by summing across the individual $GSI(SNP)$ for the SNP, which are expressed in the record.

4. Results

The process presented in section 3 was applied on the PDW data. ARM generated 55 rules for the Healthy group and 91 rules for the diseased group (both mild and severe periodontitis together). Assessment of $GSI(SNP)$ decomposed the markers across the -1 to +1 range. Out of 62 SNP's 7 markers received -1 value, which is associated with periodontal health and 9 markers received +1 $GSI(SNP)$ value, which is associated with periodontal disease. Many other SNP's ranged in-between -1 and +1 while 12 markers achieved a zero score, which indicates a complete neutral genetic susceptibility to periodontitis. The remaining SNP's ranged in-between.

We carried on with the assessment of $GSI(\mathrm{Re}cord)$ score values. Assessment was conducted over 675 Caucasian records and was also linked with severity of periodontitis (Table 1).

Table 1. $GSI(\mathrm{Re}cord)$ values and disease status.

Record-GSI	No of records	Healthy	Diseased	
			Mild	Severe
GSI ≤ 1	206	85%	5%	10%
1 <GSI ≤ 2	102	50%	5%	45%
2 <GSI ≤ 3	169	29%	19%	52%
3 <GSI ≤ 4	130	23%	24%	53%
GSI > 4	68	12%	24%	64%
Total	**675**			
%		46,52%	13,93%	39,56%

$GSI(\mathrm{Re}cord)$ correlates well with disease presence. When the overall score is less than 1 the predisposition toward healthy status is *85%* and when it is higher that 4 the predisposition toward disease is *88%*. In addition, when score value ranges between 1 and 2 there is a 50/50 chance toward either disease or healthy status.

The next step was to incorporate $GSI(\text{Re}cord)$ and $GSI(SNP)$ in further analysis. Decision tree learning using C4.5 (Quinlan, 1993) failed to produce good results judged by the average error rate during randomized *V*-cross folding validation by setting *V=10*. The dataset was split into five groups according to the $GSI(\text{Re}cord)$ scores reported in Table 1 and in each set decision tree learning was applied using age, smoking status, and the percentage values of the seven bacterial species. Average classification error ranged between 18% (for the $GSI(\text{Re}cord) \leq 1$ group) and 54% (for the $2 \leq GSI(\text{Re}cord) \prec 3$ group. Low classification accuracy convinced us that although $GSI(\text{Re}cord)$ assessment might have been in the right direction there should be another way of using the results.

We tested two different roads: (1) to replace $GSI(\text{Re}cord)$ numeric values with qualitative equivalents using an ordered scale with values: low, neutral, and high and then to proceed to decision tree learning; and, (2) to work by exploiting the detailed microbial percentage values in conjunction with $GSI(\text{Re}cord)$ scores.

4.1 Decision tree results

Decision tree induction was pioneered by Quinlan (1986) and it represents one of the most popular classification methods. The method proceeds by assessing information gain of features and then selects the most informative feature to create a decision branch and then to proceed. It is a non-backtracking process, which means that the the algorithm never looks back; however, it is fast and computationally efficient.

In the conducted experiment we used as features: age, smoking status, $GSI(\text{Re}cord)$ - valued as low, neutral or high, the seven microbial percentage values, and periodontal status as class valued either as healthy or diseased. In $GSI(\text{Re}cord)$ valuation high implies that the individual has high predisposition or genetic susceptible toward periodontitis while neutral or low valuations imply either no predisposition at all or good defense against disease. The qualitative assessment of *GSI(Record)* values was motivated by individual *GSI(SNP)* values; the records, which included at least one SNP with *GSI(SNP)* value of -1 were classified as being low in terms of *GSI(Record)* and records that included at least one *GSI(SNP)* with +1 value were valued with high *GSI(Record)*. All other records were marked as neutral. In cases where a record included at least one SNP with a -1 value and at least one SNP with a +1 value it was marked as neutral.

Classification accuracy improved significantly. During *V=10* cross-fold validation average classification error was 4.95% and when *GSI(Record),* high, neutral and low values, were removed error climbed up to 27.97%.

4.2 Exploitation of microbial measurements

The second experimental round involved linking of *GSI(Record)* score values with the microbial percentage values. Two different experiments were conducted. In the first experiment all microbial percentage values were added and formed an *m* metric. In the second experiment only three microbial percentage values were considered and formed an m_3 metric; the consideration of only three microbial values was motivated by Socransky et al (1998) and van Winkelhoff et al (2002). The *m* metric aggregates the seven bacterial species, namely: (1) *A.actinomycetemcomitans*, (2) *P. gingivalis*, (3) *P. intermedia*, (4) *T. forsythensis*, (5) *P. micros*, (6) *F. nucleatum*, and (7) *C. rectus*. The m_3 metric considers only bacterial species #1, #2 and #4 from the list, which is included in the m metric.

Table 2. Probability of an individual being periodontal healthy when *GSI(Record)* is linked with microbial values (*m* metric).

Probability of Healthy Status			
GSI(Record) value range	*m(%)* value range		
	$0, \leq 3$	$>3, \leq 35$	>35
$GSI \leq 1$	97%		44%
$1 < GSI \leq 2$	63%		32%
$2 < GSI \leq 3$	36%		4%
$3 < GSI \leq 4$	67%	22%	3%
$GSI > 4$	12%		0%

The results using the *m* metric are summarized in Table 2. The *GSI(Record)* correlates well with the sum percentage of the periodontitis associated bacteria. For example, when the susceptibility index is less or equal than one the individual can harbor high percentage (≤35%) of the seven microbial species and still be periodontal healthy. Conversely, when susceptibility increases (>4) even at low percentage (≤35%) of bacteria the probability of healthy status is low, only 12%.

However, *GSI(Record)* performs less well when it comes to discriminating between mild and severe periodontal disease.

Table 3. Conditional probability of severe periodontitis given that the person is diseased. Entries in the table correspond to probabilities.

GSI(Record) value range	*m* value range		
	$0, \leq 3$	$> 3, \leq 35$	>35
$GSI \leq 1$	100%		78%
$1 < GSI \leq 2$	90%		82%
$2 < GSI \leq 3$	83%		84%
$3 < GSI \leq 4$	74%		
$GSI > 4$	82%		

Conditional probability results summarized in Table 3 indicate that *GSI(Record)* values and microbial aggregates do not correlate well.

When investigation was limited to three microbial species alone (m_3 metric) results were analogous to those obtained with the metric (and reported in Table2 and 3) – see for example, Table 4.

The conditional periodontal disease probability results when m_3 is used are similar with results presented in Ta-

ble 3 and demonstrate the poor correlation between the aggregate of the three microbial values and *GSI(Record)*.
A third attempt to exploit microbial and *GSI(Record)* values was via structured equation modeling (SEM). SEM is a multivariate statistical technique, which is used to represent, estimate, and test hypotheses about relations between observed and latent variables. SEM is a method widely used in the behavioral sciences and it is an a priori technique, meaning that the researcher must specify a model in order to conduct the analysis (Kline, 2005). In SEM parameters are estimated by minimizing the difference between the observed co-variances and those implied by the model.

Table 4. Probability of an individual being periodontal healthy when *GSI(Record)* is linked with the three microbial values (m_3 metric).

Probability of Healthy Status				
GSI(Record) value range	m_3 value range			
	= 0	> 0, ≤ 2	>2, ≤ 10	> 10
≤ 1	100%	87%		22%
1 <GSI ≤ 2	100%	82%	57%	12%
2 <GSI ≤ 3	67%	22%		5%
3 <GSI ≤ 4	38%	26%		5%
>4	67%	0%		

We generated a structural model that shows significant association between GSI and bacterial species #2, #3, and #4, and then between these species and periodontal disease severity code (=0 for healthy, =1 for mild periodontitis and =2 for severe periodontitis). In addition to that, an independent link was generated between species #1 and disease severity code. We present the SEM model in Figure 1.
The model in Figure 1 corresponds to the regression equation: *Periodontal severity code* = 0,43 * *GSI(Record)* + 0,26 * *T. forsythensis* + 0,24 * *P. gingivalis* + *0,09* * *P. intermedia* + 0,10 * *A.actinomycetemcomitans.* The model was assessed using Analysis of Moment Structures (AMOS version 4.01 software) and explains 47% of the variation in disease severity code.

4.3. Discussion of results

In section 3 we presented the basics of *GSI(SNP)* and *GSI(Record)* calculation. In the sequel section 4 we presented the application of the susceptibility metrics in the periodontal disease data of ACTA.
Decision tree analysis was based on a qualitative representation of *GSI(Record)* values, which, in turn, were based on individual *GSI(SNP)* values. Classification accuracy improved significantly with respect to accuracy if genetic marker data were not used. This finding confirms that genetic markers add value to disease prediction.
The results presented in Tables 2 – 4 confirm to a large extent the susceptibility index value ranges reported in Table 1. Moreover, on the side of the *m* (or m_3) metrics the results confirm what is already known, which is that high microbial values are associated with periodontal disease manifestation – see for instance (van Winkelhoff et al. 2002). However, microbial value ranges intermingle between healthy and diseased individuals causing difficulties in discriminating between the two groups: see for instance Socransky et al (1998).

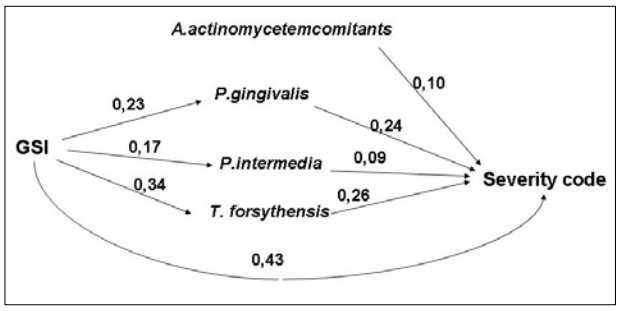


Figure 1. Linking *GSI(Record)* with microbials and disease severity code. The values on the arrows correspond to standardized regression weights. The model fits well the data; TLI = 1.000, CFI = 1,000, and RMSEA = 0.000. TLI is the Tucker Lewis Index, CFI is the Comparative Fitness Index and RMSEAis the Root Mean Square Error Approximation. No strict thresholds for these statistics currently exist, but the following general guidelines have been suggested in the literature: TLI and CFI values above 0.9 and RMSEA values less than 0.05 are generally interpreted as indicating good model fit [AMOS, 1999; Baumgartner & Homburg, 1995; Shook et al., 2004].

An additional validation of the *GSI(SNP)* and *GSI(Record)* calculation model comes from the SEM experiment. Despite the rather low percentage of variance in disease severity code explanation (equal to 47%) the model shows interesting direct and indirect associations (see Figure 1) as well as a significant link between the susceptibility index and disease status – the latter confirms once more the results presented in Table 1.
The implications of susceptibility index assessment are critical to bioinformatics research and practice. Essentially, *GSI(SNP)* and *GSI(Record)* point toward the integration of genotype and phenotype information and the improvement of clinical practice and decision-making. Genetic susceptibility to disease manifestation is already confirmed and in particularly for periodontal disease earlier studies – see previous studies for instance Michalowicz et al (2000), Loos et al (2005) and Laine et al (2001, 2005), among others, provide sufficient evidence. However, what is missing is an operational tool, which will take research results a step forward. Once, genetic susceptibility profile reaches the clinical practice level then it will become part of the patient's records. The clinician will be able to use the genetic profile of the patient, and via concrete and valid models and procedures incorporate genetics into medical decision-making and reasoning. The integration of genetic information into routine medical practice represents a challenging area of endeavor and is cast towards enabling the implementation of genomic medicine (Martin-Sanchez et al, 2004).
Blois (1988) argues that medical reasoning is vertical and demands the effective integration of multiple disciplines

as one moves from the molecular level to the organ, system levels and finally to the individual level. Addition of genetic information confirms the argument of Blois yet it adds to complexity since an additional and rather *large* set of data need to incorporated in the scene.

5. Concluding remarks

We started this study as a data mining project. Our aim was to quarry into the data warehouse of ACTA, use the periodontal records and to derive useful disease and healthy status patterns. We soon realized that the data that were available were *not balanced*. By that we mean that records did not have the same size and also incorporated heterogeneous data. In the genetic part information was encoded in terms of gene, locus and genotype and the combination of the three yielded 63 possible values. (In reality only 62 values were used). Thus in an early attempt we valued SNP's as binary features and then proceeded to *equalize* record size by incorporating all 63 possibilities in each record. Results were poor and to a large extent incomprehensible from a medical point of view. Equalizing record size gave us the opportunity to use decision tree learning, which at the outset seems to be a right tool to use [Kodratoff et al, 1994].

To overcome difficulties with record size we attempted to discard genetic data and to focus on microbial values. Results were again poor and that did not surprise us given earlier research findings reported in the literature – for example by [Socransky et al, 1998] not to mention that decision tree learning *may prove* naïve when it comes to numerically value features and association rule mining does not work at all with numerically valued attributes. During decision tree learning a numerically valued feature is split into two intervals and then treated as binary (on the basis of the less or equal and greater than threshold).

The two failures led us to the conclusion that we should take a different road. A road that would give a chance to the dataset to provide us with the optimum information that it would be able to provide. Symbolic data (such as SNP expressions) could speak well for themselves if they were used with the right learning algorithm and such was association rule mining (ARM). But ARM was not enough. We needed a procedure to take ARM results a step further and create an index that would summarize the output. At that point the rather old model of Ackoff joined the process and furnished us with the concept of knowledge gain – the operational structure was already available – see [Moustakis, 2006]. The combination yielded the genetic susceptibility indices we were looking for. Before continuing we should clarify that attribute assessment has been a long standing issue in inductive learning – see for instance the work of Baim [Baim, 1988]. The researcher interested in attribute weight value assessment may look into the literature and find other (and possibly better) ways of doing the work.

The next step was to validate susceptibility index values. At the SNP level we identified SNP's that are defenders, SNP's, which are betrayers, SNP's, which have not decided whether they are defenders or betrayers (these are the SNP's with zero susceptibility index value), and finally SNP's, which lie in-between and are either inclined toward defense or are inclined toward treachery. We presented a simple calculation formula to assess an overall susceptibility index score given the individual SNP susceptibility values. One may argue that the model is too simple. The argument may be correct yet there is not guarantee that a more complex model would yield better results; in addition, a more complex model would necessitate the formulation (and subsequent testing) of hypotheses and to this end one should keep in mind Occam's razor.

The fact that simplicity works is proven by the decision tree results that we achieved when genetic susceptibility was interpreted as low, neutral or high. We could have stopped at that point given the good classification accuracy (more than 95% during randomized testing) that we achieved. We did not do so because the trees were *poor* in content. Poor means that the tree provided a result, which was about the same with the result presented in Table 1.

Thus we decided to go along and to take susceptibility for an extra ride. We correlated susceptibility with microbial values and concluded our investigation in Tables 2 – 4 and in Figure 1. We found that the susceptibility index correlates with microbial values and on top of that explained why microbial values alone can be confusing with respect to disease manifestation.

Of special interest is the result that structural equation modeling provided – see Figure 1. The model that we learned shows direct and indirect association between the susceptibility index, selected microbial values and disease presence; it also shows that *actinobacillus* is an independent agent of disease status and is not affected directly by the genetic markers.

The results we present herein correspond to work in progress. Further confirmation is necessary at the SNP level; is for instance *CARD15_Locus_2104_Genotype_11* a loyal defender against periodontitis? Or, should on the basis of a +1 *GSI(SNP)* value for *FcyRIIIa* with *Locus_158VoverF_plus559* and *Genotype_11* be considered as a disease flag? To inquiry further, extensive biomedical literature search - possibly with the aid of text mining, is necessary.

The article tells a story. It shows how a *hybrid* data mining technique may be used to numerically assess a genetic susceptibility index. It also shows that effective mining in data that contain heterogeneous information requires hybrid algorithmic structures. We hope that other researchers will find our story interesting and pursue it in different domains.

Acknowledgment

Work reported herein is partially supported via the INFOBIOMED, *Structuring European Biomedical Informatics to Support Individualized Healthcare* Network of Excellence, IST-507585, http://www.infobiomed.org. Results, views and opinions expressed herein do not necessarily correspond to official INFOBIOMED or European Commission position and the responsibility lies entirely with the authors.

References

[Ackoff, 1958] R.L Ackoff , Towards a behavioral model of communication, *Management Science*, 4:218—234, 1958.

[ACTA, 2005] State of the Art on Data Biomedical Informatics in Chronic Infectious and Inflammatory Disease Research: Periodontitis as a Case Study (Public Deliverable). INFOBIOMED, IST IST-507585. Available from http://www.infobiomed.org

[Agrawal and Srikant, 1994] R. Agrawal, and S. Srikant, "Fast Algorithms for Mining Association Rules", 20th *Int'l Conf. on Very Large Data Bases*, Santiago, Chile, p. 487-499, 1994.

[AMOS, 1999] J. L. Arbuckle,. AMOS (version 4.01) [computer software]. Chicago: SmallWaters, 1999.

[Baim, 1988] P.W. Baim, A Method for Attribute Selection in Inductive Learning Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):888-896, 1988.

[Baumgartner & Homburg, 1995] H. Baumgartner, and C. Homburg, Applications of structural equation modeling in marketing research: A review. International *Journal of Research in Marketing*, 13, 139–161, 1995.

[Blois, 1988] M.S. Blois, Medicine and the nature of vertical reasoning, *The New England Journal of Medicine*, 318(13):847—851, 1988.

[Kline, 2005] R.B. Kline, *Principles and practice of structural equation modeling*. New York: Guilford, 2005.

[Kodratoff *et al*, 1994] Y. Kodratoff, V. Moustakis and N. Graner, Can Machine Learning Solve my Problem? *Applied Artificial Intelligence: An International Journal*, 8(1): 1-36, 1994.

[Laine *et al*, 2001] M.L. Laine, M.A Farre, G. Gonzalez, L.J. van Dijk, A.J. Ham, E.G. Winkel, J.B. Crusius, J.P. Vandenbroucke, A,J, van Winkelhoff, and A.S. Pena, Polymorphisms of the interleukin-1 gene family, oral microbial pathogens, and smoking in adult periodontitis. *J Dent Res* 80, 1695-1699, 2001.

[Loos *et al*, 2005] B.G. Loos, R.P. John, and M.L. Laine, Identification of genetic risk factors for periodontitis and possible mechanisms of action. *J Clin Periodontol* 32 Suppl 6, 159-179, 2005.

[Mannila *et al*, 1994] H. Mannila, H. Toivonen, A.I. Verkamo, Efficient algorithms for discovering association rules, *KDD-94: AAAI Workshop on Knowledge Discovery in Databases*, Seattle, Washington, pp. 181-192, 1994.

[Martin-Sanchez *et al*, 2004] F. Martin-Sanchez, I. Iakovidis, S. Norager, V. Maojo, P. de Groen, J. Van der Lei, T. Jones, K. Abraham-Fuchs, R. Apweiler, A. Babic, R. Baud, V. Breton, P. Cinquin, P. Doupi, M. Dugas, R. Eils, R. Engelbrecht, P. Ghazal, P. Jehenson, C. Kulikowski, K. Lampe, G. De Moor, S. Orphanoudakis, N. Rossing, B. Sarachan, A. Sousa, G. Spekowius, G. Thireos, G. Zahlmann, J. Zvarova, I. Hermosilla, and F.J. Vicente, Synergy between medical informatics and bioinformatics: facilitating genomic medicine for future health care. *J Biomed Inform* 37, 30-42, 2004.

[Michalowicz *et al*, 2000] B.S. Michalowicz, S.R. Diehl, J.C. Gunsolley, B.S. Sparks, C.N. Brooks, T.E. Koertge, J.V. Califano, J.A.., Burmeister, and H.A. Schenkein, Evidence of a substantial genetic basis for risk of adult periodontitis. *J Periodontol* 71, 1699-1707, 2000.

[Moustakis, 2006] V. Moustakis, Post-supervised based learning of feature weight values. In In: G. Antoniou, G. Potamias, C. Spyropoulos & D. Plexousakis (editors), *Advances in Artificial Intelligence: Proceedings of the 4th Hellenic Conference in AI – LNAI 3955*, Springer, pp. 279 – 289, 2006.

[Potamias *et al*, 2005] 18. G. Potamias, L. Koumakis and V. Moustakis, Mining XML Clinical Data: The HealthObs *System. Ingéniere des Systèmes d'Information*, vol. 10 (1): 59-80, 2005.

[Quinlan, 1986] J. R. Quinlan, Induction of decision trees, *Machine Learning*, 1:81--106. 1986.

[Quinlan, 1993] J.R. Quinlan, *Programs in machine learning*. SanMateo, CA: Morgan Kaufmann, 1993.

[Socransky *et al*, 1998] S.S. Socransky, A,D Haffajee, M.A Cugini, C. Smith, and R.L Kent Jr., Microbial-complexes in subgingival plaque. *J Clin Periodontol* 25, 134-144, 1998.

[Shook *et al*, 2004] C.L. Shook, D.J. Ketchen Jr., G. Hult, and K.M. Kacmar, An assessment of the use of structural equation models in strategic management research. *Strategic Management Journal*, 25, 397–404, 2004.

[Van Winkelhoff *et al*, 2001] A.J. Van Winkelhoff, C.J. Bosch-Tijhof, E.G. Winkel, and W.A. van der Reijden, Smoking affects the subgingival microflora in periodontitis. *J Periodontol* 72, 666-671, 2001.

An empirical comparison of four procedures for filtering monitoring data

Niels Peek[1], **Marion Verduijn**[1,4], **Evert de Jonge**[2], **Bas de Mol**[3,4]
[1]Dept. of Medical Informatics, [2]Dept. of Intensive Care Medicine, [3]Dept. of Cardio-thoracic Surgery,
Academic Medical Center, University of Amsterdam, P.O. Box 22700, 1100 DE Amsterdam, The Netherlands
[4]Dept. of Biomedical Engineering, University of Technology, Eindhoven, The Netherlands
n.b.peek@amc.uva.nl

Abstract

A well-known problem in critical care is the occurrence of erroneous measurements ("artifacts") in monitoring data. Experienced clinicians ignore these measurements when they interpret the data. For inexperienced clinicians, as well as computerized medical assistants, however, artifacts must be removed. This paper compares the performance of four artifact filtering procedures on monitoring data from a Dutch adult ICU. Three procedures (moving median filtering, ArtiDetect, and tree-based filtering) were earlier described in the literature; the fourth procedure is a new combination of existing approaches. The evaluation was carried out on blood pressure and heart rate measurements from cardiac surgery patients during their postoperative recovery. None of the four procedures was superior on all types of variables. It is advised to employ a well-chosen inductive bias when choosing an artifact filtering procedure for a given variable.

1 Introduction

Clinical treatment in anaesthesia and critical care requires a close and continuous watch on the patient's vital functions. For this reason, operating theatres and intensive care units (ICUs) are equipped with monitoring systems for automatically measuring and recording many clinical variables with high frequency. Monitoring data, however, often contain inaccurate and erroneous measurements ("artifacts"), caused by interferences on transducer signals and misplacement of probes [Cunningham *et al.*, 1994]. Such measurements hamper interpretation and analysis of the data, as they do not reflect the true state of the patient. In practice, experienced clinicians ignore artifacts when they inspect monitoring data. For inexperienced clinicians and residents, however, artifacts may pose serious problems and induce incorrect beliefs on the patient's condition. Similarly, computerized medical assistants that operate on monitoring data may be led astray by artifacts, resulting in incorrect warnings and recommendations [Miksch *et al.*, 1996; Michel *et al.*, 2003; Charbonnier, 2005].

During the last decade, several procedures for automatic detection of artifacts in monitoring data have been described in the literature. These procedures can be used to filter out artifacts from the data, thus facilitating interpretation of the data by clinicians and computerized assistants. A most basic, and frequently applied, procedure is *moving median filtering* [Mäkivirta *et al.*, 1991; Jakob *et al.*, 2000; Hoare and Beatty, 2000]. It removes data points with a relatively high or low value as compared to a moving median smoother. More sophisticated is the procedure described by C. Cao *et al.* [Cao *et al.*, 1999], called *ArtiDetect*, which considers both absolute and relative peaks in the data. C.L. Tsien *et al.* [Tsien *et al.*, 2000] compute various moving statistics (e.g. mean, median, slope, standard deviation) and select those that predict artifacts well by supervised learning. The procedures of Cao and Tsien have been evaluated by their developers, but not by others.

This paper compares the performance of these three artifact detection procedures on a set of monitoring data from a Dutch adult ICU. In addition, a fourth procedure, which was designed by the authors and combines the three procedures described above, is evaluated.

The evaluation is carried out on 30 series of blood pressure and heart rate measurements from cardiac surgery patients during their postoperative recovery. The same data were used in a preliminary experiment that was presented at last year's IDAMAP workshop [Verduijn *et al.*, 2006] which compared three different smoothing techniques (kernel smoothing, local regression, and smoothing splines) in a filtering procedure that resembled moving median filtering. We were able to filter out roughly 50% of all artifacts in that study; the differences between the three smoothers were small.

2 Data and methods

2.1 Monitoring data

Monitoring data were used of the department of Intensive Care Medicine of the Academic Medical Center in Amsterdam, The Netherlands. At this department, patients are monitored by Philips IntelliVue Monitor MP90 systems[1]. The monitoring data are recorded with a frequency of one measurement per minute in the Metavision ICU information system developed by iMDsoft[2].

Our study is restricted to three physiological variables that concern the cardiovascular system: mean arterial blood

[1] www.medical.philips.com
[2] www.imdsoft.com

pressure (ABPm), central venous pressure (CVP), and heart rate (HR). These variables are recorded in the ICU information system with equal frequency, but they differ greatly in their variability. For instance, arterial pressure and heart frequency are much more amenable to sudden changes than venous pressure.

The study population consisted of 367 patients who underwent cardiac surgery at the AMC in the period of April 2002 to June 2003. All available values for the three cardiovascular variables were retrieved from the ICU information system, yielding time series of several thousands of measurements for each patient. Using visual inspection of these data, 30 subseries with a relatively rough course were selected for our experiment. Each of these subseries included several hundreds of measurements (durations of two to five hours); they originated from 18 different patients. Overall, 10 ABPm, 13 CVP, and 7 HR subseries were selected, with a total length of 2701, 3193, and 2005 minutes, respectively.

The 30 time series were inspected by four senior ICU physicians from the Academic Medical Center (where the data were recorded). Their individual judgments were subsequently harmonized in a consensus meeting. Thirty measurements (1.1%) in the ABPm time series were judged as artifacts, 70 measurements (2.2%) in the CVP time series, and 46 measurements (2.3%) in the HR time series. We used the consensus judgments as reference standard for tuning and evaluating the automated filtering procedures.

2.2 Automated filtering procedures

Methods for automated artifact detection assume that each measurement $x(t)$ in a series is composed of the actual physiological state $f(t)$ of the patient at time at time t, and a random term $\varepsilon(t)$ representing the measurement error at time t. So, we have that

$$x(t) = f(t) + \varepsilon(t) \quad (1)$$

for all time points t where measurements are made. The error term $\varepsilon(t)$ is itself probably composed of multiple terms or factors with varying distributions. When $|\varepsilon(t)|$ is large, we say that $x(t)$ is an *artifact*. It is then better to replace $x(t)$ by a reconstruction of $f(t)$, or to remove $x(t)$ from the series. In this study, we confine ourselves to removing $x(t)$, which is called *filtering*.

The main problem for artifact detection methods is that we neither know $f(t)$ nor $\varepsilon(t)$. Roughly speaking, there are three directions to solve this problem:

A. One can focus on $f(t) + \varepsilon(t)$, and decide that when this quantity is large (in the absolute sense), then $\varepsilon(t)$ must have been large, and therefore $x(t)$ is an artifact.

B. One can try to reconstruct $f(t)$, and then estimate $\varepsilon(t)$ as the difference of $x(t)$ and the reconstruction $\hat{f}(x)$.

C. One can try to reconstruct $\varepsilon(t)$ directly by considering the variance of x.

Below, we describe the four automated filtering procedures that were applied and evaluated in this study. Each procedure employs one direction, or a combination of the above directions, and they jointly cover the spectrum of possibilities.

Moving median filtering A well-known approach to artifact filtering is based on direction B, and uses a statistical measure of central tendency to estimate $f(t)$. A popular choice is the median, which is very flexible due to its lack of distributional assumptions. The approach classifies measurement $x(t)$ as artifact when the absolute residual $|x(t) - \hat{f}(x)|$ is larger than a given threshold δ_x.

Because f may vary over time, $\hat{f}(t)$ is obtained by computing the so-called *moving* median on a small set $x(t-k)$, $x(t-k+1), \ldots, x(t+k)$ of measurements in the vicinity of $x(t)$. Here, $ws = 2k+1$ is called the *window size*.

In our study, we obtained moving medians of the time series for varying window sizes (i.e., 5, 11, 21, 31, 41, 51, 61, 71, 81, 91, and 101 minutes), and calculated the corresponding absolute residuals. For each of the three variables, window size ws and classification threshold δ_x were subsequently optimized by cross-validation on the data, using the artifact reference standard that was defined by the four clinicians.

ArtiDetect ArtiDetect [Cao *et al.*, 1999] is a procedure that combines two detectors, based on directions A and C, respectively. The *limit-based detector* classifies measurement $x(t)$ as artifact when it is outside an interval $I_x = [lb, ub]$ of admissible values. For each remaining data point $x(t)$, the *deviation-based detector* subsequently estimates $\varepsilon(t)$ as $x(t)$'s contribution to the moving standard deviation of x, and classifies $x(t)$ as artifact when $\hat{\varepsilon}(x)$ is larger than a given threshold ν_x.

For each of the three variables, we determined interval the parameters lb and uv, the window size of the moving standard deviation, and the classification threshold ν_x with cross-valdiation on the data, again using the consensus-based reference standard. For the moving standard deviation, the same eleven possible window sizes were considered as in the moving median.

Tree induction procedure Both moving median filtering and ArtiDetect employ moving statistics for artifact detection, and use the data to optimize the associated parameters (thresholds, window sizes). However, both procedures are biased by the choice of statistic and the term that they attempt to reconstruct.

C.L. Tsien *et al.* [Tsien *et al.*, 2000] have proposed an approach where the data is used to select both the appropriate statistic(s) and the associated parameters. To this end, a large number of moving statistics are computed for varying window sizes, and a multivariate tree model is induced from them. The available artifact reference standard is employed as class variable during tree induction. The procedure also takes *context information* into account, by computing the moving statistics not just for variable x but also for variables that were simultaneously measured.

In our study, we induced a tree model for each of the three variables as follows. First, we obtained eight moving summary statistics (i.e., mean, median, slope coefficient of a linear model, absolute value of that slope coefficient, standard deviation, maximum value, minimal value, and range) of the time series for three window sizes: 3, 5,

and 10 minutes.[3] These moving summary statistics were also obtained for the simultaneously measured time series of the two other variables in our study. The resulting 72 features (8 summary statistics $\times$ 3 window sizes $\times$ 3 variables) were subsequently used as predictive features for inducing a tree model.

Combined procedure As the three procedures described above may complement each other we integrated these procedures into a combined procedure, which operates as follows. First, interval and window size parameters for ArtiDetect's limit-based detector are derived from the data. After exclusion of all measurements that are classified as artifacts by this detector, for each $x(t)$ the absolute residual $|x(t) - \hat{f}(x)|$ with respect to the moving median $\hat{f}(x)$ is determined, and $\varepsilon(t)$ is estimated as $x(t)$'s contribution to the moving standard deviation of x (as in ArtiDetect's deviation-based detector). This is performed for the eleven window sizes that we used in these procedures. A multivariate tree model is subsequently built from the resulting 22 features. Note that we do not consider simultaneous measurements in the combined procedure.

2.3 Evaluation

We tuned the procedures for automated filtering to the 10 ABPm, 13 CVP, and 7 HR time series with the aim to compare their performance for the different variables. To make optimal use of the available data, we evaluated the performance of the procedures using 10-fold cross-validation. We used the consensus judgement of the measurements as reference standard, and we quantified the performance in terms of the sensitivity (i.e., the proportion of artifacts that have been classified as such by the automated filtering procedure) and the positive predictive value (i.e., the proportion of measurements that have been classified as artifacts by the automated procedure that are artifacts according to clinical judgement). As the non-artifacts were overrepresented in the time series ($> 97\%$), we do not report the specificity and negative predictive value.

3 Results

Table 1 lists the number of data points that were excluded, and the performance of each of the four filtering procedures. For ABPm, ArtiDetect has the best sensitivity (23 out of 30 artifacts detected) while moving median filtering has superior PPV (only 2 false positives). Overall, the performance of both procedures is reasonable on this variable, whereas the other two procedures perform poorly. For CVP, all procedure obtain satisfactory results. ArtiDetect and the combined procedure are notable for very good results, in terms of both sensitivity and PPV. For HR, the combined procedure is better than the others, with a reasonable to good performance (35 out of 46 artifacts detected, 7 false positives). ArtiDetect performs remarkably poor on this variable (15 artifacts detected, 9 false positives).

Figure 1 (next page) visualizes the results of the four filtering procedures on a series of ABPm measurements.

[3] These are the same window sizes as were employed by Tsien *et al.* [Tsien *et al.*, 2000] in their study.

Table 1: Number of data points classified as artifacts, sensitivity, and positive predictive value (PPV) of each of the four filtering procedures, listed per variable type (ABPm, CVP, and HR). All results obtained with 10-fold cross-validation.

Variable	Procedure	Excl	Sens	PPV
ABPm	Median filtering	22	0.667	0.909
	ArtiDetect	36	0.767	0.639
	Tree induction	26	0.600	0.692
	Combined procedure	32	0.667	0.625
CVP	Median filtering	86	0.871	0.710
	ArtiDetect	61	0.843	0.967
	Tree induction	61	0.729	0.836
	Combined procedure	65	0.857	0.923
HR	Median filtering	29	0.543	0.862
	ArtiDetect	24	0.326	0.625
	Tree induction	40	0.565	0.650
	Combined procedure	42	0.761	0.833

The left-hand graph shows that moving median filtering (crosses) only classified large outliers in the ABPm time series as artifact, while neglecting smaller artifact peaks. ArtiDetect (circles) also correctly identified a number of such less extreme artifacts, at the expensive of two false positives. These two data points were not considered as artifacts in the consensus judgment as they were part of an increasing trend; ArtiDetect turned out to be not able to discern these data points. The right-hand graph shows that the combined procedure (circles) behaved almost similarly as ArtiDetect on this series with two exceptions: it correctly classified one of ArtiDetect's false positives as a non-artifact, but it did not identify the artifact that is halfway the sudden increase to 160 mmHg. The tree induction procedure of Tsien *et al.* (crosses) failed to classify a large outlier in the series as artifact that has another outlier as neighbor measurement; one of the small outliers in the series was correctly identified as artifact by this procedure.

Table 2 and 3 list, for each of the monitoring variables, the parameters that were estimated from the data in moving median filtering and in ArtiDetect's limit-based detector and deviation-based detector. The parameters of the moving median filter reflect that the variable CVP is least amenable to sudden changes: the filter uses a relatively large window size and small classification threshold to detect artifacts. The variable also has a relatively small range of admissible values, as appears from Table 3. For the HR variable, no upper bound on valid measurements could be established.

Table 2: Estimated window sizes (ws) and classification thresholds (δ_x) for the moving median filter.

Variable	ws	δ_x
ABPm	11	51
CVP	91	16
HR	101	39

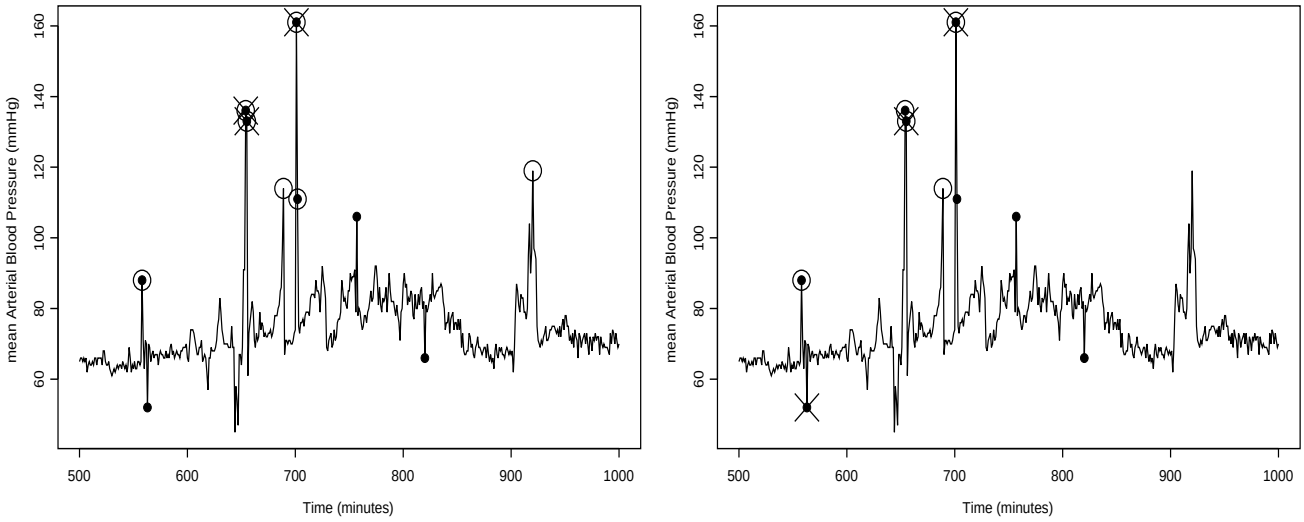


Figure 1: Results of automated filtering on a series of 500 ABPm measurements. Shaded circles represent data points that were judged to be artifacts by the physicians (reference standard, 8 data points). Left-hand graph (a): results of moving median filtering (crosses) and ArtiDetect (circles). Right-hand graph (b): tree induction procedure by Tsien *et al.* (crosses) and combined procedure (circles). All results were obtained by training and testing on separate sets (10-fold cross-validation).

In both the procedure of Tsien *et al.* and the combined procedure, a class probability tree is induced from the data. Due to space restrictions, we only display two of the resulting trees, and summarize the others. Figure 2 shows the two tree models that were induced for filtering CVP data. The left-hand tree, resulting from Tsien *et al.*'s procedure, uses a variety of different moving statistics to detect artifacts, including range, median, absolute value of the slope coefficient, and minimum value. The tree almost exclusively refers to CVP values, and uses only one of the other variables, ABPm, for a small set of cases. Closer scrutiny reveals that the tree imitates the limit-based detector of ArtiDetect at various places, using the moving median statistic with window size 3. For instance, the right-hand subgroup of the upper left branch judges data points with a moving median smaller than 0 to be artifacts with 82% certainty. The right-hand side of the tree similarly contains a branch where data points with a moving median greater than 41 are classified as artifacts with 100% certainty. Note that these boundaries exactly correspond to those of ArtiDetect's limit-based detector (Table 3). Another interesting phenomenon occurs at the rightmost leaf of the tree. This leaf represents data points in unstable parts of a CVP time series (range ≥ 16) without a clear trend (absolute slope coefficient < 5). They are estimated to have a high probability (88%) of being an artifact. A similar probability is found for relatively high CVP values that have been measured in the context of low mean arterial blood pressure measurements (rightmost of the two lowest leafs).

Table 3: Estimated parameters for ArtiDetect: ranges of admissible values (I_x) for the limit-based detector), and window sizes (ws) and classification thresholds (ν_x) for the deviation-based detector.

variable	I_x	ws	ν_x
ABPm	[1,154]	11	2.96
CVP	[0,41]	31	0.72
HR	[39,∞)	91	0.35

The right-hand tree, resulting from the combined procedure after filtering extreme values using the limit-based detector of ArtiDetect, uses statistics that quantify the measurements' contribution to the time-dependent standard deviation for a variety of window sizes. Statistics related to the deviations from the reconstructed time series (direction B of Sec. 2.2) were not included. Note that the primary split of the tree exactly corresponds to ArtiDetect's deviation-based detector for this variable (Table 3). When compared to ArtiDetect's deviation based detector, the combined procedure employs four extra features describing a measurement's contribution to the standard deviation.

Table 4 summarizes the moving statistics and number of leaf nodes of the tree models induced from the ABPm, CVP, and HR data in the tree induction procedure of Tsien *et al.* It appears from this table that moving statistics of the simultaneously measured blood pressure(s) was used as context information for filtering the CVP and HR data. No context information was used for filtering of ABPm time series. The included statistics and number of leaf nodes in the tree models that are induced in the combined procedure after filtering extreme values using the limit-based detector are summarized in Table 5. The primary split in the tree model for HR time series, an absolute error statistic, turned out to be an important filtering feature; this finding explains the poor performance of ArtiDetect for these data.

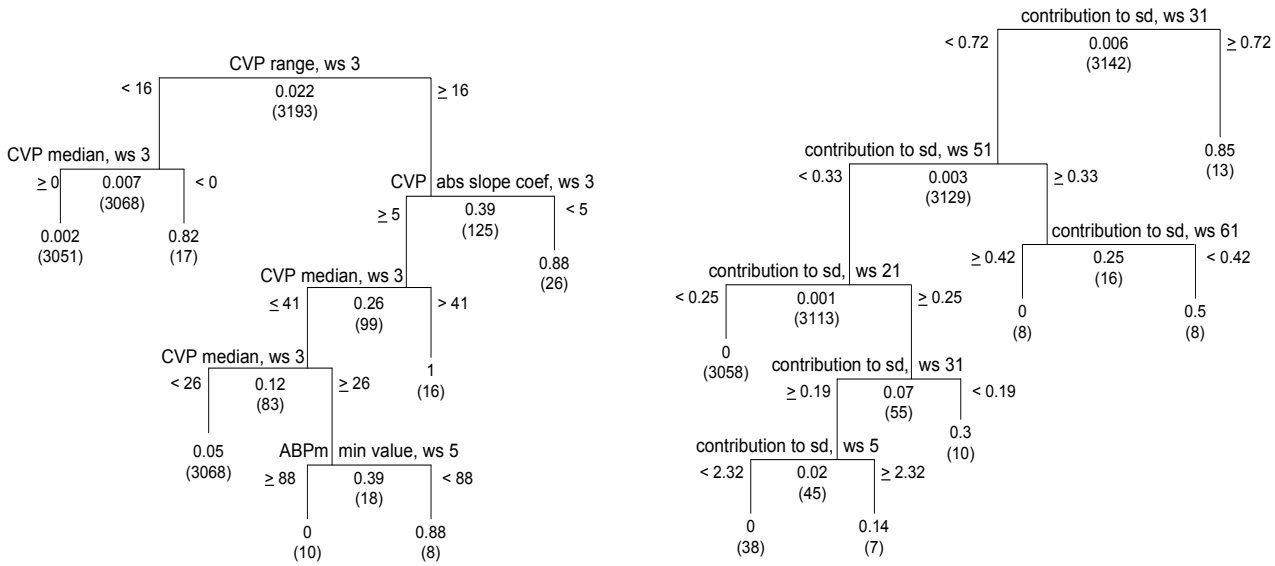


Figure 2: Tree models for filtering CVP time series as resulted from a) the tree induction procedure of C.L. Tsien *et al.*, and b) the combined procedure. The models that are shown here were derived from the entire data set (no cross-validation). Note that the second tree was built after exclusion of 51 data points that were classified as artifacts by the limit-based detector. Each internal node is labeled with the moving statistic that is used for classification and the associated window size (ws). Each leaf node is labeled with the estimated probability of being an artifact, and, between brackets, the number of observations in the relevant subgroup of the data set.

Table 4: Moving statistics and corresponding window sizes (between brackets) included in the tree models resulting from the procedure of Tsien *et al.*, and size of trees (number of leaf nodes).

Variable	Included statistics	Size
ABPm	ABPm: standard deviation (3), median (3), absolute value of slope coefficient (3), mean (3)	8
CVP	CVP: range (3), absolute value of slope coefficient (3), median (3) ABPm: min. value (5)	7
HR	HR: median (3), min. value (10) ABPm: mean (5) CVP: min. value (3)	6

Table 5: Results for the tree models induced in combined procedure, after application of ArtiDetect's limit-based detector. The statistics (absolute error and contribution to the standard deviation) with corresponding window sizes (between brackets) as included in the tree models, and size of trees (number of leaf nodes).

Variable	Included statistics	Size
ABPm	contr to sd (11, 101)	3
CVP	contr to sd (31, 51, 21, 61, 5)	7
HR	abs residual (91, 51) contr to sd (21, 41, 61, 101)	10

4 Discussion and conclusions

We have applied and evaluated three existing procedures and one new procedure for filtering artifacts from ICU monitoring data. None of the procedures was superior in detecting artifacts for all three clinical variables: median filtering outperformed the others on mean arterial blood pressure, ArtiDetect and the combined procedure were best on central venous pressure, and the combined procedure had again the better performance on heart rate. The tree induction procedure of Tsien *et al*. was never superior to all other procedures. ArtiDetect had the largest variation in performance among the three variables.

In a preliminary study on the same data, we compared three different smoothing techniques (kernel smoothing, local regression, and smoothing splines) in a filtering procedure that resembled the moving median filter [Verduijn *et al.*, 2006]. In that study, theoretically impossible (e.g., negative) blood pressures were removed before the filters were applied, and for these variables the results can therefore not be compared directly to the current results. For heart rate, however, both sensitivity and positive predictive value were inferior to the moving median filter that was applied here.

The current study is the first one to externally validate and compare the filtering procedures by Cao *et al.* and Tsien *et al.* External validation, i.e., validation at sites other than the one that was used for development, is important because procedures may be implictly geared towards the local situation in which they were developed [Justice *et al.*,

1999]. A similar implicit source of bias may exist when developers evaluate their own procedure [Friedman and Wyatt, 2006]. Both types of bias may explain the relatively modest performance that was found in this study, compared to the performance reported in the original studies.

A third source of bias in our study is the fact that the time series were selected for their relatively rough course, and stable time series were therefore underrepresented. The results therefore do not represent the performance of the procedures on monitoring data in general. We expect that the two relatively inflexible procedures (moving median filtering and ArtiDetect) will have more trouble on such data.

In contrast to many other studies in the field of artifact detection, our reference standard was not defined by a single expert but based on consensus among four senior ICU clinicians. Because the notion of 'artifact' is vague and inherently subjective, we believe that a consensus-based standard is preferable to single-expert standards. The definition of a consensus-based standard is however laborious, and for this reason our dataset was smaller than in most other studies on artifact detection.

Our data is additionally characterized by absence of *combined probing*, the simultaneous measurement of multiple variables by a single probe. Combined probing is rare in adult ICUs, but customary in neonatal ICUs. It leads to correlations in the occurrence of artifacts in the variables in question. Because C. Cao *et al.* developed their ArtiDetect procedure on neonatal data, they also proposed a *correlation-based* detector in addition to the limit-based and deviation-based detectors. As all variables in our study were measured with separate probes, we have not implemented the correlation-based detector. Our version of ArtiDetect therefore differs from the original one, but we do not expect this has influenced the results.

Also the tree induction procedure of C.L. Tsien *et al.* was slightly modified in our application. In the original study [Tsien *et al.*, 2000], the binary variable indicating the occurrence of artifacts was smoothed in a preprocessing step: measurements were marked with true if the majority of measurements in a surrounding window of five measurements were originally labelled as artifacts. In the smoothed outcome therefore only *artifact episodes* remain, and the procedure is geared towards detecting such episodes. Because artifact episodes were scarce in our dataset, we decided to not apply the preprocessing step. Perhaps that the procedure, which performed relatively poor in our study, was set at a disadvantage by this decision.

To summarize, a reasonable performance was obtained on our data, but no single procedure outperformed the others on all variables. Because of the large differences between variables, we conclude that is wise to employ a well-chosen (e.g. clinically motivated) inductive bias when choosing an artifact detection procedure for a given variable. Furthermore, the performance of ArtiDetect and Tsien et al.'s procedure was substantially lower in our study than in the original investigations, stressing the need for external validation studies in this field. Finally, we believe there is room for improvement in the methods that are based on machine learning. A possible direction for future research is rule induction.

Acknowledgments We would like to thank Marcus Schultz, Erik-Jan van Lieshout and Anne-Cornelie de Pont, senior ICU physicians at the Academic Medical Center, Amsterdam, The Netherlands, for scoring the temporal patterns. Niels Peek receives a grant from the Netherlands Organization of Scientific Research (NWO) under project number 634.000.020.

References

[Cao *et al.*, 1999] C. Cao, N. McIntosh, I.S. Kohane, and K. Wang. Artifact detection in the po_2 and pco_2 time series monitoring data from preterm infants. *J Clin Monit Comput*, 15:369–78, 1999.

[Charbonnier, 2005] S. Charbonnier. On line extraction of temporal episodes from ICU high-frequency data: a visual support for signal interpretation. *Comput Methods Programs Biomed*, 78:115–32, 2005.

[Cunningham *et al.*, 1994] S. Cunningham, A.G. Symon, and N. McIntosh. The practical management of artifact in computerised physiological data. *Int J Clin Monit Comput*, 11:211–6, 1994.

[Friedman and Wyatt, 2006] C.P. Friedman and J.C. Wyatt. *Evaluation Methods in Biomedical Informatics*. Springer, New York, 2nd ed., 2006.

[Hoare and Beatty, 2000] S.W. Hoare and P.C.W. Beatty. Automatic artifact identifaction in anaesthesia patient record keeping: a comparison of techniques. *Med Eng Phys*, 22:547–53, 2000.

[Jakob *et al.*, 2000] S. Jakob, I. Korhonen, E. Ruokonen *et al*. Detection of artifacts in monitored trends in intensive care. *Comput Methods Programs Biomed*, 63:203–9, 2000.

[Justice *et al.*, 1999] A.C. Justice, K.E. Covinsky, and J.A. Berlin. Assessing the generalizability of prognostic information. *Ann Intern Med*, 130:515–24, 1999.

[Mäkivirta *et al.*, 1991] A. Mäkivirta, E. Koski, A. Kari *et al*. The median filter as a preprocessor for a patient monitor limit alarm system in intensive care. *Comput Methods Programs Biomed*, 34:139–44, 1991.

[Michel *et al.*, 2003] P. Michel, F. Roques, S.A.M. Nashef, and The EuroSCORE Project Group. Logistic or additive EuroSCORE for high-risk patients. *Eur J Cardiothorac Surg*, 23:684–7, 2003.

[Miksch *et al.*, 1996] S. Miksch, W. Horn, C. Popow, and F. Paky. Utilizing temporal data abstraction for data validation and therapy planning for artificially ventilated newborn infants. *Artif Intell Med*, 8:543–76, 1996.

[Tsien *et al.*, 2000] C. L. Tsien, I.S. Kohane, and N. McIntosh. Multiple signal integration by decision tree induction to detect artifacts in the neonatal intensive care unit. *Artif Intell Med*, 19:189–202, 2000.

[Verduijn *et al.*, 2006] M. Verduijn, N. Peek, E. de Jonge, and B. de Mol. A procedure for automated filtering of ICU monitoring data using basic smoothing techniques and clinical judgement. Working notes of *Intelligent Data Analysis in bioMedicine and Pharmacology (IDAMAP–06)*, pages 31–6, 2006.

Paper session:

Temporal Datamining / Information Retrieval

Temporal Discretization of medical time series - A comparative study

[1]Revital Azulay, [1]Robert Moskovitch, [1]Dima Stopel, [2]Marion Verduijn, [3]Evert de Jonge, and [1]Yuval Shahar
[1]Medical Informatics Research Center, Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel
{robertmo,stopel,azorevi,yshahar}@bgu.ac.il
[2]Dept of Medical Informatics, [3]Dept of Intensive Care Medicine, Academic Medical Center - University of Amsterdam, P.O.B. 22700, 1100 DE Amsterdam, The Netherlands {m.verduijn,e.dejonge}@amc.uva.nl

Abstract

Discretization is widely used in data mining as a preprocessing step; discretization usually leads to improved performance. In time series analysis commonly the data is divided into time windows. Measurements are extracted from the time window into a vectorial representation and static mining methods are applied, which avoids an explicit analysis along time. Abstracting time series into meaningful time interval series enables to mine the data explicitly along time. Transforming time series into time intervals can be made through discretization and concatenation of equal value and adjacent time points. We compare in this study five discretization methods on a medical time series dataset. Persist, a temporal discretization method yields with the longest time intervals and lowest error rate.

1 Introduction

Time oriented data presents an exceptional opportunity to analyze data, having a better and more natural analysis. Often, features from time series, such as minimal value, are extracted and represented as vectors for further use in static data mining algorithms. This is made through windowing, in which the data is divided to time windows and measurements are extracted from the window. It is very hard to determine the window size and this approach avoids the explicit time representation. Converting time series to time intervals series presents a more compact representation of the time series, which enables an efficient analysis of the data and further mining operations explicitly along time [Moskovitch and Shahar, 2005]. However, to transform time series to time interval series a temporal abstraction method should be applied. This can be made through discretization and concatenation of the discretized values. In this study we present five types of discretization methods, three are static and two consider the time explicitly. For the task of mining time intervals we are interested in *long time intervals* and *low level of error* relative to the original dataset.

We start with a detailed background of time intervals mining, as the motivation for this study. Later we present temporal abstractions and discretization methods. In the methods section we present the methods we used in the study and finally we discuss the results and present our conclusions.

2 Background

2.1 Mining Time Intervals

The problem of mining time intervals, a relatively young field, is attracting a growing attention recently. Generally, the task is given a database of symbolic time intervals to extract repeating temporal patterns. One of the earliest works was made by Villafane et al [1999], which searches for *containments* of intervals in a multivariate symbolic interval series. Kam and Fu [2000] were the first to use all Allen's relations [Allen, 1983] to compose interval rules, in which the patterns are restricted to right concatenation of intervals to existing extended patterns, called *A1* patterns. Höppner [2001] introduced a method using Allen's relations to mine rules in symbolic interval sequences and the patterns are mined using an Apriori algorithm. Höppner uses a k^2 matrix to represent the relations of a k sized pattern. Additionally, Höppner proposes how to abstract the patterns or make them more specific. Winarko and Roddick [2005] rediscovered Höppner's method, but used only half of the matrix for the representation of a pattern, as well as added the option to discover constrained temporal patterns. Similar to Winarko and Roddick [2005], Papapetrou et al [2005] rediscovered the method of mining time intervals using Allen's relations. Their contribution was in presenting a novel mining method consisting on the SPAM sequential mining algorithm, which results in an enumeration tree; the tree spans all the discovered patterns.

A recent alternative to Allen's relations based methods surveyed earlier was presented by Mörchen [2006], in which time intervals are mined to discover coinciding multivariate time intervals, called Chords, and the repeating partially ordered chords called Phrases.

Mining time intervals offers many advantages over common time series analysis methods commonly applied on the raw time point data. These advantages include mainly, a significant reduction in the amount of data, since we mine summaries of the time series, based on temporal abstraction methods. In addition a restriction of short time window is not needed and unrestricted frequent patterns can be discovered. However, in order to enable mining of time series through time intervals the time series have to

be abstracted to time intervals. This can be done based on knowledge acquired from a domain expert [Shahar, 1997] or based on automatic data driven discretization methods.

2.2 Temporal Abstraction

Temporal abstraction is the conversion of a time series to a more abstracted representation. This abstracted representation is usually more comprehensive to human and used as a preprocessing step to many knowledge discovery and data mining tasks. The *Knowledge Based Temporal Abstraction* (KBTA) presented by Shahar [1997], infers domain-specific interval-based abstractions from point-based raw data, based on domain-specific knowledge stored in a formal knowledge-base, e.g. the output abstraction of a set of time stamped hemoglobin measurements, include an episode of *moderate anemia* during the *past 6 weeks*. However, while the KBTA applies the temporal knowledge and creates abstractions that are meaningful to the domain expert, such knowledge is not always available. Moreover, the domain expert knowledge provided is not always the proper one for knowledge discovery and mining tasks, but rather for his routine activities, such as diagnosis. Thus, there are several automatic data driven methods which can be used for this task, which is the focus of this paper.

The task of temporal abstraction corresponds to the task of *segmenting* the time series and characterizing the data in each segment. *Segmenting time series* [Keogh *et al.*, 1993] is the task of representing a time series in a piecewise linear representation, which is the approximation of a time series length n with k straight lines, usually $k<<n$. Three common approaches for segmenting time series are: *Sliding Window* approach, in which a segment is grown until a specified error threshold is reached. *Top Down* approach repeatedly splitting the series according to best splitting point from all considered points, until a stopping criterion is met. *Bottom Up* approach starts by segmenting the series with small segments and then iteratively merges adjacent segments. A survey on temporal abstraction methods is given in [Höppner, 2002].

2.3 Discretization

Many data mining algorithms and tasks can benefit from a discrete representation of the original data set. Discrete representation is more comprehensive to human and can simplify, reduce computational costs and improve accuracy of many algorithms [Liu *et al*., 2002]. Discretization is the process of transforming continuous space valued series $X = \{x_1,...,x_n\}$ into a discrete valued series $Y = \{y_1,...,y_n\}$. The next step is usually to achieve interval based representation of the discretized series. The main part of the discretization process is choosing the best *cut points* which split the continuous value range into discrete number of bins usually referred to as *states*. Discretization methods are mainly categorized as *supervised* vs. *unsupervised* methods.

Unsupervised discretization does not consider class information or any given label. For time series class information is usually not available and unsupervised methods are needed. Two common methods are *equal width discretization* (EWD) and *equal frequency discretization* (EFD). Another method is *k-means clustering* [MacQueen, 1967], in which the time series values are grouped into k clusters (states) represented by centroids, from which the states and the cut points are deduced.

Supervised discretization considers class information, which for time series is often unavailable. There are many supervised discretization methods available in the literature. Known methods for supervised discretization [Dougherty et al, 1995] are, decision tree discretization, heuristic methods and entropy minimization based methods consisting on Shannon entropy [Shannon, 1948]. Two common decision tree algorithms using entropy measure are ID3, [Quinlan, 1986], and C4.5, [Quinlan, 1993] and error based methods [Kohavi and Sahami, 1996]. A good survey and framework for discretization is given in [Liu et al., 2002]. In this study we will focus on the application of unsupervised discretization methods to time series.

2.4 Temporal Discretization

Temporal discretization refers to the discretization of time series, usually made by unsupervised means, as a preprocessing step in transforming the time series into time intervals series. Most of the discretization methods do not consider the temporal order of the values in a time series since most of them were developed for static data. However, recently several methods were proposed, in which the temporal order is considered. Symbolic Aggregate approXimation (SAX) [Lin et al., 2003] is a method for symbolic representation of time series. ***SAX*** was the first method developed explicitly to discretize time series, based on the Piecewise Aggregate Approximation (PAA) [Keogh et al., 2000] which is a time series segmenting algorithm. However, SAX does not explicitly consider the temporal order of the values. Later Mörchen and Ultsch [2005] introduced ***Persist*** which considers the temporal order of the time series and selects the best cut point based on persisting behavior of the discretized series. We will elaborate later on these two methods in the methods section. Another discretization method for time series is suggested by Dimitrova et al. [2005]. The method combines graph theory to create the initial discretization, and information theory to optimize this discretization. The number of states returned is a number determined by the method. The ***Gecko*** [Salvador, 2004] algorithm for identifying states in a time series is a clustering algorithm which dynamically determines the number of states. Another method is ***HMM*** [Bilmes, 1997], hidden Markov model. In HMM the time series is assumed to have been created by states which are hidden, this hidden model is assumed to be a Markov process. HMM is not a discretization method in the sense of resulting in a set of cut points, in HMM the state sequence directly created.

3 Methods

3.1 Discretization methods

We examined the following five discretization methods on medical time series. Apart from Persist, which considers the temporal order of the values in the time series, and

SAX, which was designed for time series discretization, all methods are static data discretization methods.

3.1.1 Equal Width Discretization

Equal Width Discretization (EWD) determines the cut points by dividing the value range into equal width bins, as shown in figure 1. Note the amount of values in each bin is based on the distribution of the values.

3.1.2 Equal Frequency Discretization

Equal Width Discretization (EFD) divides the value range into bins having equal frequency of values in each bin as shown in figure 1.

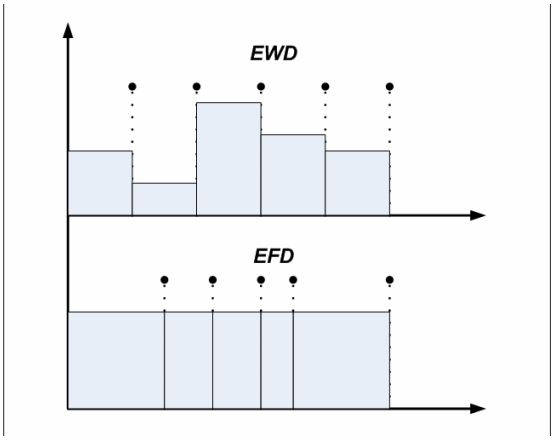


Figure 1. An illustration of the distributed values along the value range after EWD and EFD discretization.

3.1.3 K means Clustering

K-means clustering with Euclidean distance measure is a simple method that was chosen in this study as a representative for discretization method via clustering. K-means clusters the data into *k* clusters which are represented by the centroids. The clustering algorithm begins with a random or more educated choice (more efficient due to the sensitivity of the clustering process to the initial selection) of clusters centroids. The second step is to assign each data point to the cluster that has the closest centroid. After every data point has been assigned, the *k* centroids are recalculated as the mean value of each cluster. These two steps are repeated until no data point is reassigned or the *k* centroids no longer change. The resulting clusters or centroids are used as the states of the discretization process.

All the three methods presented EWD, EFD and K-means are all static methods, which do not consider the temporal order of the time series.

3.1.4 SAX

Symbolic Aggregate approXimation is one of the first discretization methods designed specifically for time series data. SAX consists of two steps, in the first step the time series is converted into a less granular representation and in the second step the abstracted time series is discretized into fixed number of states. The first step is the PAA, Piecewise Aggregate Approximation, in this step the temporal aspect of the data is taken into account. PAA is a representation of time series $X = \{x_1, ..., x_n\}$ by the vector $\overline{X} = \{\overline{x}_1, ..., \overline{x}_m\}$ (m<n), where each $\overline{x}_i$ is the mean value of n/m sequential observations of X. Two important PAA properties are dimensionality reduction and lower bounding. In dimensionality reduction, time series of length n are considered as a point in a n dimensional space, that can be reduced to a m dimensional *space* (m<n) after performing PAA dimensionality. In lower bounding, the distance between two PAA represented series is less or equal to the distance between the original two series, which guarantees no false dismissals; the PAA part of SAX is the time oriented part, which considers the temporal aspect. The second and main step of the SAX method, the discretization of the PAA output, is based on the assumption that normalized time series have a Gaussian distribution and the desire to produce equal probability states. Therefore the time series is normalized and discretized into fixed number of states according to predetermined cut points which produce equal-sized areas under Gaussian curve (the cut points chosen respectively to the selected number of states).

3.1.5 Persist

New univariate discretization method designed specifically for the purpose of knowledge discovery in time series, which for the first time explicitly considers the order of the values in the time series. Given a set of possible (discrete) symbols $S = \{S_1,\ldots,S_k\}$ of a time series of length n, Persist computes the marginal probability $P(S_j)$ of a symbol S_j and the transition probabilities given in a $k \times k$ matrix $A(j,m) = P(s_i=S_j|s_{i-1}=S_m)$, in which the self transitions are the values on the main diagonal of A. In this approach the assumption is that if there is no temporal structure in the time series, the symbols can be interpreted as independent observations of a random variable according to the marginal distribution of symbols, thus, the probability of observing each symbol is independent from the previous state, i.e. $P(s_i=S_j|s_{i-1},\ldots,s_{i-m}) = P(S_j|s_{i-1})$. Based on this Markovian model, if there is no temporal structure the transition probabilities should be close to the marginal probabilities. Otherwise if the states show persistence behavior, which is expected to result in long time intervals, the self transition probabilities will be higher than the marginal probabilities. The Persist algorithm is based on a measure based on the Kullback-Leibler Divergence [Kullback & Leibler, 1951], which indicates which cutoffs lead to a discretization which will result eventually in long time intervals. Persist method was compared to common discretization methods, and showed to achieve relatively good results. However Persist only deals with time series that comes from uniform sampling.

3.2 ICU Dataset

An ICU dataset of patients who underwent cardiac surgery at the Academic Medical Center in Amsterdam, the Netherlands, in the period of April 2002-May 2004. Two types of data were measured: *static data* including details on the patient, such as *age*, *gender*, *surgery type*, whether the patient was mechanically ventilated more than 24 hours, and *temporal data* which were used for the study. The temporal data, included two types of variables: *high frequency variables* (measured each minute): mean arte-

rial blood pressure (ABPm), central venous pressure (CVP), heart rate (HR), body temperature (TMP), fraction inspired oxygen (FiO2) and level of positive end-expiratory pressure (PEEP). FiO2 and PEEP variables are parameters of the ventilator. The variables base excess (BE), creatinine kinase MB (CKMB), glucose value (GLUC), and cardiac output (CO) are *low frequency variables* (measured several times a day). The data contains 664 patients, among which 196 patients were mechanically ventilated for more than 24 hours.

3.3 Evaluation measures

Evaluating unsupervised methods, particular discretization methods is a challenging task since there is no clear objective, such as accuracy in supervised methods. Thus, commonly in the evaluation of unsupervised methods, the evaluation measures are derived from the study objectives.

The time series abstraction task we present here includes the process of discretization, which results in corresponding time series labeled with the states representative value. The following process is the concatenation of adjacent points labeled with the same state label. The output of this state is an interval based time series. We hereby define the evaluation measures we used to evaluate the performance of each discretization method. Generally, our goal was to find the method which results with the longest time intervals, which smoothes the time series towards the task of mining. On the other side we also wanted to minimize the error defined by the difference between the state value and the original value.

3.3.1 Mean and Standard deviation of Time Intervals

To measure the length of the time intervals resulted from each method we calculated the mean and standard deviation of the resulting time intervals, as shown in formula 1.

$$\mu = \frac{\sum_{i=1}^{n} |I|}{n} \quad , \sigma = \sqrt{E\left(|I|^2\right) - \left(E(|I|)\right)^2} \qquad (1)$$

Where $|I|$ is an interval length and $E(|I|)$ is the expected value of $|I|$.

3.3.2 Error measures

To define the error or distance measure which measures the states representation relatively to the original values of the time series, we used the Euclidean distance by which we measure the distance among the original value and the state value, as shown in formula 2.

$$E_D(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (2)$$

Where $X=x_1 \ldots x_n$ is the original time series and $Y=y_1 \ldots y_n$ is the discretized series, the value of every y_i is one of the discrete states representatives values.

Note that implementing this measure is straightforward, but it isn't clear which value represents in the best way the state which includes an interval of values. Due to the sensitivity of this error measure to the state representative value, we defined two different state representative values. **States mean error** called ***Error1***, in which the state representative value is the mean value the two cut points defining the state. **State observation mean error**, called ***Error2***, in which the state representative value is chosen to be the mean value of all the original values within the state.

4 Evaluation and Results

The goal of this study was to perform an extensive evaluation on the discretization methods on the time series data presented in the ICU dataset. We applied all the five discretization methods, presented in section 3.1, with *3, 4 and 5* number of states. Finally, for each method and each amount of states the evaluation measurements were calculated. Additionally, we provide here statistical properties (standard deviation, mean, and number of observations) to characterize the original time series.

Table 1 – The characteristics of each time series. The amount of time points (n), minimal and maximal values and mean and standard deviation of the values.

	N	Min	max	Mean ± std
TMP	416665	31.00	40.00	36.77 ± 0.85
HR	455613	0.00	230.00	79.62 ± 14.50
PEEP	460146	0	20	8.05 ± 2.76
ABPm	452285	26	193.4	77.34 ± 12.10
FiO2	460774	25.8	100	44.67 ± 8.00
CVP	431885	0	44	13.98 ± 4.61
CI	3216	0.99	5.99	2.54 ± 0.68
GLUC	4234	0	26.6	8.50 ± 2.86
CKMB	2028	1.1	465.2	40.08 ± 47.49
BE	4077	-22.3	13.5	-2.54 ± 2.74

4.1 High frequency variables

The high frequency variables (TMP, HR, PEEP, ABPm, FiO2, and CVP) were evaluated with the proposed measurements, after applying the five discretization methods. Generally, the results were similar in terms of the performance of the methods. Moreover, since we were interested in finding the best method and the best amount of states and due to the lack of space in the paper, we present the mean values of the variables.

Table 2 presents the *mean* and *standard deviation* length of the intervals of all the variables for each amount of states (s), having 3, 4 and 5 states, and for each method, as well as *Error1* and *Error2*. EWD and Persist achieved the highest mean interval length. While Persist was the best in the 3 states discretization, EWD was the best in the 4 and 5 states. Usually high mean length had also high standard deviation. In average, while the mean length in the 3 states was higher than in the 4 and 5 states, the last two had the same averaged mean. Error1 and Error2 showed inconsistency in the preferred method, which is reasonable due to the expected sensitivity to the chosen state representative values. The methods having the highest mean length had also the lowest Error1, which is quite surprising since we expected to see a tradeoff. In addition, while in the averaged mean length (according to states) there was no difference between 4 and 5 states, the averaged errors decreased as more states were used.

Table 2 – The mean and standard deviation length, and errors of the high frequency variables. EWD and Persist achieved the highest values of mean length and in general the 3 states had achieved the highest averaged mean.

S	Method	Mean ± std	Error 1	Error 2
	Persist	**132.38 ± 195.75**	**4927.61**	3297.91
	SAX	68.05 ± 115.06	9865.52	2378.70
	EWD	109.73 ± 163.82	6673.67	2804.78
	EFD	66.54 ± 111.27	10209.05	2477.25
	k-means	72.56 ± 122.35	9696.55	**2355.80**
3	**Avg**	68.19 ± 141.65	8274.48	2662.89
	Persist	82.48 ± 152.34	4417.61	2799.02
	SAX	54.49 ± 95.08	8025.43	2042.51
	EWD	**100.64 ± 158.88**	**4212.42**	3013.21
	EFD	45.34 ± 85.96	8880.72	2156.79
	k-means	57.98 ± 99.73	7090.87	**1921.77**
4	**Avg**	55.9 ± 118.40	6525.41	2386.66
	Persist	61.15 ± 121.07	4136.87	2316.51
	SAX	43.53 ± 81.42	6767.22	1767.01
	EWD	**86.48 ± 142.02**	**3527.18**	2658.20
	EFD	38.86 ± 75.22	7892.18	1988.65
	k-means	47.93 ± 86.09	5531.14	**1543.19**
5	**Avg**	55.59 ±101.16	5570.92	2054.71

Table 3 The mean and standard deviation length, and errors of the low frequency variables. EWD, Persist and k-means achieved the highest values of mean length and in general the 3 states had achieved the highest averaged mean.

S	Method	Mean ± std	Error 1	Error 2
	Persist	**342.15 ± 263.90**	**461.73**	**352.93**
	SAX	170.61 ± 193.19	877.05	405.35
	EWD	307.22 ± 216.45	679.18	365.53
	EFD	165.09 ± 193.41	1355.92	475.92
	k-means	177.61 ± 194.21	725.58	370.24
3	**Avg**	232.54 ± 212.23	819.89	394.00
	Persist	**265.15 ± 239.11**	**455.99**	334.36
	SAX	141.89 ± 171.02	721.17	368.23
	EWD	264.52 ± 215.28	472.58	**306.24**
	EFD	120.76 ± 167.59	1132.04	440.34
	k-means	136.26 ± 172.67	690.84	355.23
4	**Avg**	185.72 ± 193.13	694.53	360.88
	Persist	184.28 ± 200.34	399.09	295.55
	SAX	117.74 ± 156.74	657.75	347.18
	EWD	**233.75 ± 209.90**	354.58	268.94
	EFD	99.82 ± 152.48	980.91	413.92
	k-means	115.08 ± 156.70	**333.13**	**221.62**
5	**Avg**	150.13 ± 175.23	545.09	309.44

4.2 Low frequency variables

The low frequency variables were summarized in the same way the high frequency variables, the results shown in table 3. The low frequency variables have two problematic issues, in the context of our work, since they were not measured uniformly, but manually. This is problematic in two phases, the discretization, in which SAX and Persist, which are the more temporal methods are assume the time series have fixed gaps. In addition in the interpolation step we assume the time points can be concatenated.

Persist and EWD achieved the highest mean interval length. However, here Persist outperformed in the 3 and 4 states and EWD in the 5 states. In average, less number of states created longer intervals and higher errors rate, and a tradeoff observed between the mean length and level of error. The longer the mean interval length, the highest the error.

Persist achieved the lowest Error1 in 3 and 4 states and k-means for 5 states. In Error2, Persist achieved the lowest error for 3 states, EWD for 4 states and k-means for 5 states. In general, a higher correlation was observed among the two error measures, unlike in the high frequency variables.

5 Discussion

We presented here the problem of time series discretization, as a preprocessing method in which the time series are transformed to time interval series. Generally, in such process we would like to have the highest mean length of intervals and minimal error when comparing the discretized data to the original values.

We presented five discretization methods. Three are from the traditional static discretization methods, which were not designed specifically for time series and two additional which were designed for time series. We applied the five methods on a dataset from a medical problem aiming in three levels of state abstraction: 3, 4 and 5. We assumed that the more states there will be longer time intervals, which was the desired objective, but also larger error. To measure the error we defined two measures, the first Error1 compares the original values to the middle of the state interval, and the second Error2 compares to the average of the values within the state values intervals. The dataset we used include two types of time series. High-frequency time series which were measured in fixed gaps within each pair of time points, and low frequency in which there were few measurements taken manually in varying gaps. As expected, lower amount of states resulted in longer time intervals and higher rate of error. While in the high frequency there was low correlation between the two error measures, in the low frequency there was a high correlation. This can be explained by the low amount of time points, which probably distribute like the entire state interval. However, we think that Error2 might not be the best measure since it is data driven and thus subjective and influenced by the distribution of the time series. As was shown in the results the Error2 measure was not coherent, although as a state representative mean state value (of Error2) yields smaller distance from the original series.

6 Conclusions and future work

Generally, Persist brought the best outcome. These are very encouraging results indicating that discretization

methods for time series which consider the temporal order of the values are required. However, while Persist [Mörchen and Ultsch, 2005] presents a method that explicitly considers the order of the values, it was not designed for time series having varying gaps. We are currently in the process of performing a wider evaluation on additional datasets. In addition we are developing a temporal discretization method which will take into consideration the varying gaps in any type of time series.

Acknowledgment

We used the implementation of the Persist method and other discretization methods in Matlab, provided by Mörchen on his website.

References

[Allen, 1983] J. F. Allen. Maintaining knowledge about temporal intervals, Communications of the ACM, 26(11): 832-843, 1983.

[Bilmes, 1997] J. Bilmes. A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997.

[Dimitrva *et al*., 2005] E. S. Dimitrova, J.J. McGee, and R.C. Laubenbacher. Discretization of Time Series Data, eprint arXiv:q-bio/0505028, 2005.

[Dougherty et al, 1995] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. International Conference on Machine Learning, pages 194-202, 1995.

[Höppner, 2001] F. Höppner, Learning Temporal Rules from State Sequences, Proceedings of WLTSD-01, 2001.

[Höppner, 2002] F. Höppner. Time Series Abstraction Methods - A Survey. In informatik Bewegt: informatik 2002 - 32. Jahrestagung Der Gesellschaft FÜR informatik E.V. (Gi) (September 30 - October 03, 2002). S. Schubert, B. Reusch, and N. Jesse, Eds. LNI, vol. 19. GI, 777-786, 2002.

[Kam and Fu, 2000] P. S. Kam and A. W. C. Fu, Discovering temporal patterns for interval based events, In Proceedings DaWaK-00, 2000.

[Keogh *et al*., 1993] E. Keogh, S. Chu, D. Hart, and M. Pazzani, Segmenting Time Series: A Survey and Novel Approach, Data Mining in Time Series Databases, World Scientific Publishing Company, 1993.

[Keogh et al., 2000] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowledge and Information Systems 3(3), 2000.

[Kohavi and Sahami, 1996] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, pages 114-119, 1996.

[Kullback and Leibler, 1951] S. Kullback and R.A. Leibler. On information and su_ciency. Annals of Mathematical Statistics, 22:79-86, 1951.

[Lin et al., 2003] Lin, J., Keogh, E., Lonardi, S. and Chiu, B. (2003) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13.

[Liu et al., 2002] H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: An enabling technique. Data Mining and Knowledge Discovery, (6):393-423, 2002.

[MacQueen, 1967] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Symposium on Math, Statistics, and Probability (pp. 281{297). Berkeley, CA: University of California Press, 1967.

[Mörchen, 2005] F. Mörchen, and A. Ultsch, Optimizing Time Series Discretization for Knowledge Discovery, In Proceeding of KDD05, 2005.

[Mörchen, 2006] F. Mörchen, Algorithms for Time Series Knowledge Mining, Proceedings of KDD-06, 2006.

[Moskovitch and Shahar, 2005] R. Moskovitch, and Y. Shahar, Temporal Data Mining Based on Temporal Abstractions, (IEEE) ICDM-05 workshop on Temporal Data Mining, Houston, US, 2005.

[Papapetrou, 2005] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos, Discovering Frequent Arrangements of Temporal Intervals, Proceedings of ICDM-05, 2005.

[Quinlan, 1986] J. R. Quinlan. Introduction to decision trees. Machine Learning, 1:81-106, 1986.

[Quinlan, 1993] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, California, 1993.

[Salvador and Chan, 2005] S. Salvador and P. Chan. Learning States and Rules for Detecting Anomalies in Time Series. Applied Intelligence 23, 3 (Dec. 2005), 241-255, 2005.

[Shahar, 1997] Y. Shahar, A framework for knowledge-based temporal abstraction, Artificial Intelligence, 90(1-2):79-133, 1997.

[Shannon, 1948] C.E. Shannon. A mathematical theory of communication. Bell System Tech. J. 27, 379-423, 623-656, 1948.

[Villafane et al, 2000] R. Villafane, K. Hua, D Tran, and B. Maulik, Knowledge discovery from time series of interval events, Journal of Intelligent Information Systems, 15(1):71-89, 2000.

[Winarko, and Roddick, 2005] E. Winarko, and J. Roddick, Discovering Richer Temporal Association Rules from Interval based Data, Proceedings of DaWaK-05, 2005.

Analysis of ICU Patients Using the Time Series Knowledge Mining Method

[1]Robert Moskovitch, [1]Dima Stopel, [2]Marion Verduijn, [2]Niels Peek, [3]Evert de Jonge, and [1]Yuval Shahar
[1]Medical Informatics Research Center, Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel
{robertmo,stopel,yshahar}@bgu.ac.il
[2]Dept of Medical Informatics, [3]Dept of Intensive Care Medicine, Academic Medical Center,
University of Amsterdam, P.O.B. 22700, 1100 DE Amsterdam, The Netherlands
{m.verduijn,n.b.peek,e.dejonge}@amc.uva.nl

Abstract

Time oriented data presents a more detailed description of problems, while presenting challenges in the computational needs for a successful analysis, in which the time is explicitly analyzed. Commonly temporal datasets are converted into a static representation and being analyzed by common static data mining methods, such as decision trees. Abstracting time series into time intervals, using temporal abstraction, enables to analyze the data explicitly along time. We apply here a mining method, which discovers partially ordered coinciding time intervals, consisting on the Time Series Knowledge Mining (TSKM) method, presented by Mörchen [2006] on temporal data of intensive care patients, using human defined and two types of data driven temporal descretization methods as a preprocessing step. The Persist discretization method results with the best knowledge discovery outcome.

1 Introduction

The reduction in storage cost and the growth in logged temporal data present the opportunity to analyze data along time. Often the analysis of time stamped data is made using a time window, extracting features which describe the time series within the time window and enter them into a "static" data mining algorithms, such as decision trees or naïve bayes. However, determining the right time window size is commonly problematic and extracting features from the time series within a given time window, such as minimal value, or transformations such as wavelets or Fourier transform, do not allow an explicit temporal analysis. Alternatively, abstracting time series to meaningful time intervals enable to mine temporal data in a different approach, which not necessarily enforces to use windowing, which results in explicitly mining the data along the time axis [Moskovitch & Shahar, 2005].

: Representing time series through time intervals yields compact summary representation of the time series. For example, having values within the same range for a period of time can be represented in a time interval in which there is a period of stable values. Thus, a more compact representation is presented, preserving the time axis explicitly and enabling further mining of the time intervals. Time intervals can represent events having duration, or an abstracted time series, which we refer to in this study.

Within the recent half decade a growing interest in mining time intervals was observed. Most of the methods currently use Allen's temporal relations [Allen, 1983] for the representation of temporal knowledge. While Allen's relations were used and applied widely in *temporal reasoning* it has some disadvantages in the task of mining time intervals, as presented by Mörchen [2006b], on which we will elaborate later. These include mainly *ambiguity* and lack of *robustness*, to which he presents an alternative based on partially ordered coinciding time intervals [Mörchen 2006a,b]. In this study we applied Mörchen's method to a set of monitoring data from the intensive care unit (ICU) to examine its capabilities and advantages over Allen's temporal relations. This dataset was previously analyzed in comparative studies on temporal abstraction procedures for predicting the risk of prolonged mechanical ventilation after cardiac surgery [Verduijn et al, 2005; Sacchi et al, 2006].

We start by surveying the background. In the methods section we describe the descretization methods we used, the ICU dataset, the evaluation measures and the experimental plan. Finally, we report the results and discuss Mörchen's method as an alternative to Allen's.

2 Background

2.1 Time Intervals

Time intervals are defined often as a triple $ti=<ti._{start}, ti._{end}, ti._{symbol}>$, having a start-time, end-time and a symbolic value. Time intervals can represent an event in real life which has duration or an abstraction of time series data. Real life events have duration, represented by time interval, e.g., the duration in which the temperature of a patient increases. However, since the measurements are instantaneous resulting in time series, temporal abstraction is used to represent them as time intervals [Shahar, 1997].

2.2 Temporal Knowledge Representation

Temporal knowledge representation is an essential tool in the task of temporal data mining, which influences the entire mining process. Often Allen's thirteen temporal relations *before, meets, overlaps, starts, during, finishes*, and their corresponding *inverse* [Allen, 1983] are used for the representation of the time interval temporal patterns, however, in the context of knowledge discovery Allen's relations have some disadvantages, which we discuss in the following sections.

2.2.1 Mörchen's TSKM

In a very recent paper Mörchen [2006a] criticizes Allen's temporal relations as a tool for temporal knowledge discovery, specifying three main aspects:

(1) *Robustness*, claiming they are not robust since part of them requires the equality of two or more interval endpoints. For example the relations *overlap*, *during* and *finishes* can describe a very similar situation, as shown in examples 1.a,b,c in figure 1, which illustrate the relations among two time intervals A and B.
(2) *Ambiguous*, since the same relation of Allen can visually and intuitively represent very different situations. Examples 2.a,b,c in figure 1 (modified from [Mörchen, 2006b]) illustrates the different situations which are all defined as *overlap*, while the different *overlap* may have different meanings.

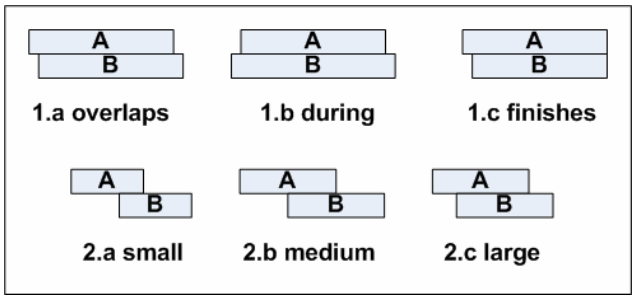


Figure 1 Examples 1.a,b,c illustrating the lack of *robustness* in Allen's relations, in which *overlap*, *during* and *finishes* represent a very similar situation. 2.a,b,c presents examples justifying the *ambiguity* in Allen's temporal relations.

(3) *not easily comprehensible*, representing a temporal pattern of time intervals using Allen's temporal relations requires the definition of all the pair-wise relations among each pair of time intervals [Höppner, 2001] which grows exponentially with the size of the pattern.

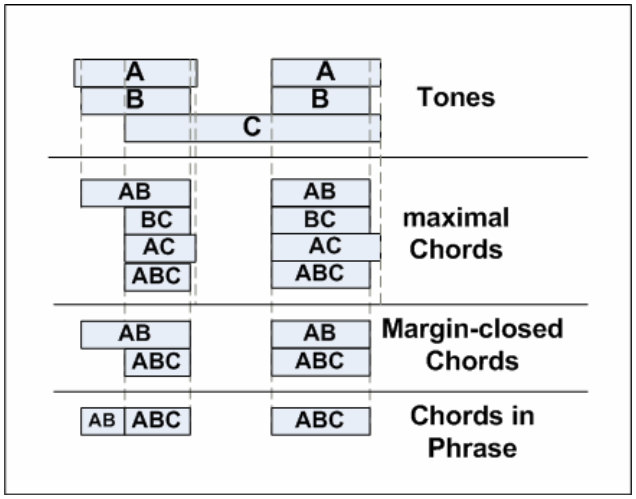


Figure 2 First we have Tones, which are labeled time interval appearing ini' the raw data. Maximal coinciding chords become Chords. A partial order of Chords constructs a Phrase.

Instead of Allen's thirteen temporal relations, Mörchen introduced a new hierarchical language for the representation of temporal knowledge based on time intervals, called the *Time Series Knowledge Representation* (TSKR). The TSKR is a hierarchical interval language describing the temporal concepts of *duration*, *coincidence*, and *partial order* in interval time series. The TSKR consists on three components: *Tones*, *Chords* and *Phrases*. A Tone is the basic primitive component, which is a labeled interval representing duration. Simultaneously occurring Tones form a Chord, representing *coincidence*. According to [Mörchen, 2006a,b] a Chord is defined by the sum of maximal appearances, thus no explicit restriction on the length of a Chord is defined. Several Chords connected with a partial order form a Phrase. A Phrase is defined by the partial order of the Chords, but the durations of the gaps are not restricted. Figure 2 (modified from Mörchen [2006b]) illustrates the process of the knowledge representation in the TSKR method. At the top there are the Tones, labeled time intervals (raw data), which based on their maximal coincidence Chords are constructed. Eventually a Phrase is constructed based on a partial order of Chords. The discovered Phrases are represented by a directed graph of Chords, in which the edges are Chords and their partial order is presented by the curves connecting them, as shown in figure 4.

2.3 Mining Time Intervals

2.3.1 Allen's based Time Intervals Mining

The problem of mining time intervals, a relatively young field, is attracting a growing attention recently. Generally, the task is, given a database of symbolic time intervals, to extract repeating temporal patterns. One of the earliest works was made by Villafane et al [1999], which searches for *containments* of intervals in a multivariate symbolic interval series. A containment model is constructed from the intervals, and rules are mined. Kam and Fu [2000] were the first to use all Allen's relations to compose interval rules. In their search for patterns it is restricted to right concatenation of intervals to existing extended patterns, called *A1* patterns. Additionally, they restrict the length of a pattern to a maximal length. Höppner [2001] introduced a method using Allen's relations to mine rules in symbolic interval sequences, in which the time series are restricted by a sliding window, and the patterns are mined using an Apriori algorithm. In contrast to Kam and Fu [2000] rules are generated and their interestingness is measured based on the intermediate confidence within the time intervals. Höppner uses a k^2 sized matrix to represent the relations of a k intervals sized pattern. Additionally, Höppner proposes how to abstract the patterns or make them more specific. In addition, Papapetrou et al [2005] rediscovered the method of mining time intervals using Allen's relations. However, their contribution was in presenting a new mining method consisting on the SPAM sequential mining algorithm, which results in an enumeration tree which spans all the discovered patterns.

2.3.2 Mining Coincidence of Time Intervals

Recently, an alternative method to Allen's relations based mining methods, was proposed by Mörchen's, which is the technique we focus on in this paper. As we explained earlier, all the mining methods which we described in section 2.3.1, consisting on Allen's temporal relations, suffer from the failures listed earlier. In addition to his *TSKR* temporal knowledge representation method, Mörchen proposed a mining algorithm, called TSKM, to mine Tones, Chords, and Phrases [2006a,b]. The input for mining coincidence

in the form of Chords is a set of Tones with the respective symbolic interval sequence. A Chord is observed if a set of coinciding intervals exceed a minimal length, and frequent if they all exceed a minimal level of support threshold. To mine Phrases, a partial order mining algorithm is required, which is similar to mining *Episodes* of instantaneous temporal data. However, unlike in Episodes in which they consist on time points, here is the input is time intervals. Mörchen extended the existing CHARM algorithm for mining Episodes. The approach was evaluated on a dataset of roller bladders, and analyzed by a sport physician, arguing that this approach is better than Allen's temporal knowledge representation.

In this study we applied Mörchen's mining method on a clinical dataset to examine the method in a new setup. While Morchen in his experiments tried it on a single dataset, which was split into time windows, in our study the dataset is constructed from independent patients in which we want to discover repeating Phrases. Additionally, we wanted to examine Mörchen's method in the light of his criticisms on Allen's representation based mining methods.

3 Methods

3.1 Abstracting Time Series

To transform the time series into time interval (Tones) series we performed *state* temporal abstraction, in which the time series are discretized into several states based on categories given from an expert or from a data driven computational source, as a preprocessing stage. Each time series went through three types of state abstraction: discretization given by a human expert (physician) and data driven methods using *SAX* [Lin et al, 2003] and *Persist* [Mörchen, 2005]. We briefly describe these approaches.

3.1.1 Symbolic Aggregate AproXimation

Recently the Symbolic Aggregate aproXimation (SAX) has been presented [Lin et al, 2003], which is based on the Piecewise Aggregate Approximation (PAA) [Keogh et al, 2001]. The PAA is a *dimensionality*[1] *reduction* method proposed for the problem of similarity search in large time series databases, in which it is crucial to reduce the amount of time units which represents the time series for the purpose of efficient manipulations, while maintaining a proper approximation commonly made through satisfying the *lower bounding* criterion introduced by Faloutsos et al [1994]. In the dimensionality reduction process a time series of n dimensions is transformed into N dimensions ($N<<n$). PAA enables the reduction of the dimension of a time series through splitting the time series into fixed length frames which are represented by the mean of the values within the frame. SAX first uses PAA to reduce the dimensional representation, then after normalizing the resulted values to the mean of zero and variation one, achieves interval representation by discretizing the normalized value range into equal sized areas ender the Gaussian curve, similarly to EFD, resulting in alphabetical symbols for each state. The frame size and the alphabet size create a tradeoff between efficiency and approximation accuracy. SAX is the first symbolic representation of time series with an approximate distance function that lower bounds the Euclidean distance. While SAX is one the first discretization methods designed specifically for time series data, the temporal aspect (order of values) of the data are taking into account only in the preprocessing stage of the PAA, thus not being an explicitly temporal method.

3.1.2 Persist

In a recent study Mörchen and Ultsch [2005] proposed *Persist*, a new *univariate* discretization method designed specifically for the purpose of knowledge discovery in time series, which for the first time explicitly considers the order of the values in the time series. Given a set of possible (discrete) symbols $S = \{S_1,\ldots,S_k\}$ of a time series of length n, Persist computes the marginal probability $P(S_j)$ of a symbol S_j and the transition probabilities given in a $k \times k$ matrix $A(j,m) = P(s_i=S_j|s_{i-1}=S_m)$, in which the self transitions are the values on the main diagonal of A. In this approach the assumption is that if there is no temporal structure in the time series, the symbols can be interpreted as independent observations of a random variable according to the marginal distribution of symbols, thus, the probability of observing each symbol is independent from the previous state, i.e. $P(s_i=S_j|s_{i-1},\ldots,s_{i-m}) = P(S_j|s_{i-1})$. Based on this Markovian model if there is no temporal structure, the transition probabilities should be close to the marginal probabilities, otherwise if the states show persistence behavior, which is expected to result in long time intervals, the self transition probabilities will be higher than the marginal probabilities. The Persist algorithm is based on a measure based on the Kullback-Leibler Divergence, which indicates which cutoffs lead to long time intervals. The method is compared to common discretization methods, such as EQW, SAX, HMM and more simple ones, and results in higher accuracy [Mörchen, 2006].

However, Persist assumes that any time series come from uniform sampling, in which the duration between each time point is fixed, which is not always the situation, especially in "*slow*" domains (sampled infrequently and commonly manually), such as the medical domain or other in which the sampling is made manually in varying periods of time. Thus, a more generalized framework should be developed which considers the distance among the time points with the time series.

3.2 Data Set

An ICU dataset was used of patients who underwent cardiac surgery at the Academic Medical Center in Amsterdam, the Netherlands, in the period of April 2002-May 2004. Two types of data were measured: *static data* in-

[1]Dimensionality in the context of time series refers to the amount of time units a time series is represented by, unlike in statistics and machine learning, in which it refers to the amount of independent variables (features).

cluding details on the patient, such as *age*, *gender*, *surgery type*, whether the patient was mechanically ventilated more than 24 hours during her postoperative ICU stay, and *temporal data*, measured each minute along the first 12 hours of the ICU hospitalization, including: mean arterial blood pressure (ABPm), central venous pressure (CVP), heart rate (HR), body temperature (TMP), and two ventilator variables, namely fraction inspired oxygen (FiO2) and level of positive end-expiratory pressure (PEEP). The data contains 664 patients, among which 196 patients were mechanically ventilated for more than 24hr (29.5%). The objective in this case study on the ICU dataset was to discover common temporal patterns for patients who were mechanically ventilated for more than 24hr for patients who were detubated within 24hr.

3.3 Evaluation Measures

We used the unsupervised TSKM mining method (2.2.1) which results with a set of Phrases, having each a support value above a given threshold. Evaluating knowledge discovered from a mining process is challenging since it is hard to estimate the quality of the discovered knowledge in quantitative terms, such as accuracy in classification. Since in our task the objective was to discover the common patterns of the two types of patients, we defined two measures to estimate the distance among two sets of Phrases. The first measures the pair-wise distances among each pair of Phrases in both sets, and the second is based on the minimal pair-wise distances.

Real examples of Chords and Phrases discovered in this study are presented in figure 3 and 4 respectively. Chord C1 describes the duration, in which ABPm, CVP, and HR at level_2 and TMP level_3 coincide. C4 describes the duration, in which FiO2 at level 2 and PEEP and TMP at level 3. These Chords appear in the examples in Figure 4, which presents two examples of Phrases. We will use these examples for the explanation of the measures.

C1 (#292, 0.80):
hf-ABPmTones is Level_2
hf-CVDTones is Level_2
hf-HRTones is Level_2
hf-TempTones is Level_3

C4 (#251, 0.16):
hf-FiO2Tones is Level_2
hf-MpeepTones is Level_3
hf-TempTones is Level_3

Figure 3 Example of two chords, C1 and C4. The support of C1 is 0.8. The support of C4 is 0.16.

Let $P^+=\{P^+_1, P^+_2,.., P^+_n\}$ and $P^-=\{P^-_1, P^-_2,.., P^-_m\}$ be the set of Phrases discovered from the patients, who were mechanically ventilated for more than 24 hours, and the ones who received ventilation less than 24 hours, respectively. We define a distance measure $d(P_i,P_j)$ among two (single) Phrases, inspired by measures which are commonly used to compare graphs.

$$d(P_i,P_j)=\frac{2\cdot I(|E(P_i)|,|E(P_j)|)}{|E(P_i)|+|E(P_j)|} \quad (1)$$

Where $I(E(P_i)E(P_j))$ is the amount of common directed edges in P_i and Pj, and |E(P)| is the total number of directed edges in Phrase P. We double $I(E(P_i)E(P_j))$ to have a measure in [0,1] range.

As an example for the calculation of $d(P_i,P_j)$ we will use the two Phrases presented in figure 4. The number of mutual directed edges is 2: The edges 1→8 and the edges 8→9. The total number of edges in *P4* is 6 and in *P5* is 4. Thus, $d(P4,P5)$ equals $2*2/(6+4) = 0.4$.

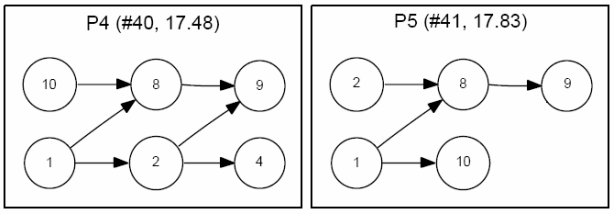


Figure 4 Example of two phrases, P4 and P5. The nodes (directed edges) are the chords, having an id, and connected according to their partial order.

3.3.1 Cross Product Distance

To measure the distance among the two sets of Phrases we measure the pair-wise distances of all the $n\times m$ pairs of Phrases. We started by computing the mean of all of the nxm pair-wise distances, as the distance between P^+ and P^- which resulted in $CPD_1(P^+,P^-)$ presented in equation 2.

$$CPD_1(P^+,P^-)=\frac{\sum_{p^+\in P^+}\sum_{p^-\in P^-} d(p^+,p^-)}{|P^+|\cdot|P^-|} \quad (2)$$

The obvious drawback of D_1 is the lack of consideration of the *support* of each Phrase. The intuition guided us in the extension of CPD_1 was that the support value of a Phrase should be considered as a weight representing its significance among the others. Thus, each pair of Phrases CPD_1 measure is multiplied by their average of supports values, which represents their weight in the final distance measure among the two sets of Phrases presented in CPD_2 measure, as shown in equation 3.

$$CPD_2(P^+,P^-)=\frac{\sum_{p^+\in P^+}\sum_{p^-\in P^-}\left[d(p^+,p^-)\left(\frac{p^+_{\sup}+p^-_{\sup}}{2}\right)\right]}{\sum_{p^+\in P^+}\sum_{p^-\in P^-}\frac{p^+_{\sup}+p^-_{\sup}}{2}} \quad (3)$$

3.3.2 Minimal Distance

In addition to the cross product distance we defined a more compact and extreme measure, called *minimal distance*, in which we measure the distance for each Phrase with the closest Phrase from the other set, as shown in equation 4.

$$MD_1(P^+,P^-)=\sum_{i=1}^{k}\min[d(P^+_i,P^-_j)],1\le j\le m \quad (4)$$

Note that in this measure, several Phrases from P^+_i can be coupled to the same P^-_j. Similar to CPD_2 we extended MD_1 to MD_2, in which the support values are used again to represent the importance of both Phrases.

3.4 Experimental Plan

To compare between the knowledge discovered from the positive and the negative patients we used Mörchen's [2006a] TSKM method in a slightly different approach. While typically the method computes the Tones, Chords, and Phrases from the entire dataset, since here the dataset was split into two classes and in order to be able to compare the discovered Phrases, we discretized the time series into Tones and discovered Chords based on the entire dataset. Then the Phrases were discovered separately for each group of the positive and negative patients, which yielded eventually in two sets of Phrases, P^+ and P^-, which were constructed from the same set of Chords (and Tones).

This discovery process was applied to the datasets after applying three types of preprocessing, in which the high frequency variables abstracted into Tones, including human expert (EdJ, the intensive care physician involved in the study), SAX, and Persist. The human expert discretization is provided in Table 1. The temporal data was abstracted according to the cut points presented in the table.

Finally, each type of a discretization yielded in two sets of Phrases, which we wanted to measure the distance among them to estimate the expected detection accuracy. The discretization which would result with the highest level of distance was expected to be the results in the best separation of the knowledge discovered from both groups.

Table 1: Cut-points determined by human expert.

Variable	Human expert cut-points
hf-ABPm	60,90
hf-CVP	5,17
hf-FiO2	41,60
hf-HR	60,110
hf-PEEP	7.333
hf-TMP	35.5,38.5

4 Results

We first refer to the preprocessing stage, in which the time series were abstracted to Tones. We measured the results of the application of each abstraction method to the time series by the mean and standard deviation of the resulted time intervals (tones), as shown in table 2.

Table 2: The mean length of the intervals for each variable and each discretization method

Variable	Human	SAX	Persist
hf-ABPm	17.03±78.17	8.78±39.98	21.98±92.55
Hf-CVP	15.6±92.85	8.54±38.34	25.1±186.25
Hf-FiO2	48.44±244.85	49.58±225.32	126.97±348.93
hf-HR	30.91±205.98	12.45±83.37	31.04±199.47
hf-PEEP	117.33±346.6	35.89±191.49	76.07±282.06
Hf-TMP	174±498.9	73.03±241.37	179.72±509.43

In most cases the Persist method achieved the longest intervals. However, note that for the PEEP measure the human expert discretization achieved much longer intervals than other discretization techniques.

Table 3 presents the average number of intervals and number of Chords for each discretization method. As expected the methods having low number of time intervals have also the longest length in the results presented in table 2.

While, a relatively significant difference in the average number of intervals was observed, where Persist has the minimal amount, then the human expert and finally SAX with the largest amount of intervals, the amount of the discovered Chords was quite similar.

Table 3: Average number of intervals and number of chords for each discretization method.

Discretization	Avg. # of intervals	# of Chords
Human Expert	13,171.16	12
SAX	25,853.50	11
Persist	9,988.83	10

Table 4 presents the amount of Phrases discovered based on each discretization method. The Phrases where discovered separately from positive and negative patients, thus their number is different for each group.

Table 4: Number of Chords and Phrases for each discretization technique.

Discretization	# of P^+	# of P^-
Human expert	23	32
SAX	17	33
Persist	21	16

Although we defined two types of measures D_1 and D_2, in which the support value is considered to represent the relative importance of a Phrase, we present only the D2 types. This is as a result of the lack of room and the magnitude of the results was the same. Table 5 presents the D_2 values computed for each discretization method. Persist results with the highest distance among the positive and negative patients, according to the D_1 measure. SAX resulted with the lowest distance, and the human expert based abstraction yielded in a relatively high separation level.

Table 5: Distance values $MD_2(P^+,P^-)$ and $CPD_2(P^+,P^-)$ for each discretization method.

Discretization	MD_2	CPD_2
Human expert	111.37	0.21
SAX	38.46	0.08
Persist	172.87	0.30

5. Discussion

We presented the problem of mining time series represented by time intervals through temporal abstraction. We presented Mörchen's TSKM and applied it to the problem presented in the ICU dataset, in which there is a dataset of patients classified into two classes. The temporal abstraction of the data was made based on three inputs: a *domain expert*, and two data driven discretization methods: *SAX* and *Persist*. The TSKM mining method was applied on the resulted time interval series. First, Chords were discovered from the *entire dataset* to enable comparison of the further discovered Phrases, which were discovered separately from each class of patients. according to their

duration being mechanically ventilated. Knowledge was discovered in the form of a set of Phrases from each class of patients. To evaluate the differences in the discovered knowledge we defined a distance measure among two Phrases and two sets of Phrases.

In this study, the cutoffs in the three discretization methods were fixed along the entire time series. This is a limitation of the study, as during the postoperative ICU stay of cardiac surgical patients, the definitions of 'normality' and 'abnormality' change during the twelve-hour period for variables such as PEEP and TMP [Verduijn et al, 2005] (creating a context [Shahar, 1997]). In this study, we used the mean value of these cut-points for these variables in the human discretization method. Discretization with dynamic cut points is an important topic of our future work. However, note that while these contexts are relevant for physicians' for diagnosis and prognosis purposes, it might not be always (and for any variable) relevant in the task of temporal knowledge discovery or classification tasks.

The results of the study indicate that the data which was discretized by the Persist method resulted with the highest level of separation, followed by the human expert and SAX. While the human expert descretization was expected to be more meaningful, it is not necessarily expected to lead to better knowledge discovery since it was defined for diagnosis purposes and not knowledge discovery. SAX while aiming to be a discretization method for time series does not explicitly considers the temporal order of the time series, which might result in the pure results.

Referring to Mörchen's method in the light of his criticisms [Mörchen, 2006a,b] on Allen's relations, our observation is that the TSKM suffers from part of the aspects indicated by Mörchen on Allen's relations. While it does not suffer from robustness, having only two operators representing *coincidence* and *synchronity* (*partial order* of chords), similar to Allen's *equal* and *before* respectively, it is not expressive enough to show the actual relations among the time intervals. Additionally, it is ambiguous as well as Allen's relations since Chords discovered can have different durations which may reflect different meanings, as well as in Phrases, in which the gap duration among Chords in a Phrase may vary, similar to the Allen's *before*.

8. Conclusions and Future Work

In this study we showed the ability to analyze time series using temporal abstraction. We applied the TSKM on a dataset including two classes of patients. Phrases were discovered from each class and the distance among the discovered sets were measured. As future work, we would like to measure the accuracy of the classification using the discovered Phrases. In addition we develop an Allen based time interval mining method which is expected to overcome the criticisms presented by Mörchen.

Acknowledgements

For this study we used the implementation of the TSKM and the discretization methods in Matlab, provided by Mörchen on his website.

References

[Allen, 1983] J. F. Allen. Maintaining knowledge about temporal intervals, Communications of the ACM, 26(11): 832-843, 1983.

[Höppner, 2001] F. Höppner, Learning Temporal Rules from State Sequences, Proceedings of WLTSD-01, 2001.

[Kam and Fu, 2000] P. S. Kam and A. W. C. Fu, Discovering temporal patterns for interval based events, In Proceedings DaWaK-00, 2000.

[Lin et al, 2003] Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13.

[Mörchen and Ultsch, 2005] F. Mörchen, and A. Ultsch, Optimizing Time Series Discretization for Knowledge Discovery, In Proceeding of KDD05, 2005.

[Mörchen, 2006a] F. Mörchen, Algorithms for Time Series Knowledge Mining, Proceedings of KDD-06, 2006.

[Mörchen, 2006b] F. Mörchen, A better tool than Allen's relations for expressing temporal knowledge in interval data, Proceedings of the KDD06 Workshop on Temporal Data Mining 06, 2006.

[Moskovitch and Shahar, 2005], R. Moskovitch, and Y. Shahar, Temporal Data Mining Based on Temporal Abstractions, (IEEE) ICDM-05 workshop on Temporal Data Mining, Houston, US, 2005.

[Papapetrou et al, 2005] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos, Discovering Frequent Arrangements of Temporal Intervals, Proceedings of ICDM-05, 2005.

[Sacchi et al, 2006] L. Sacchi, M. Verduijn. N. Peek, E. de Jonge, B. de Mol, R. Bellazzi. Describing and modeling time series based on qualitative temporal abstraction. Workshop notes of the IDAMAP workshop, 2006.

[Shahar, 1997] Shahar, Y., A framework for knowledge-based temporal abstraction, Artificial Intelligence, 90(1-2):79-133, 1997.

[Verduijn et al, 2005] M. Verduijn. A. Diagliati, L. Sacchi, N. Peek, R. Bellazzi, E. de Jonge, B. de Mol. IC prediction from patient monitoring data: a comparison of two temporal abstraction procedures. Proceedings of the AMIA 2005 Annual Symposium, 2005.

[Villafane et al, 2000] R. Villafane, K. Hua, D Tran, and B. Maulik, Knowledge discovery from time series of interval events, Journal of Intelligent Information Systems, 15(1):71-89, 2000.

Learning susceptibility of a pathogen to antibiotics using data from similar pathogens

Steen Andreassen[1], Alina Zalounina[1], Leonard Leibovichi[2], Mical Paul[2]
[1]Center for Model-based Medical Decision Support, Aalborg University, Denmark
az@hst.aau.dk
[2]Department of Medicine E, Rabin Medical Center, Beilinson Hospital, Petah-Tiqva, Israel

Abstract

Considerations for selecting empirical antibiotic therapy rely on prior knowledge of the *in vitro* susceptibilities of potential pathogens to antibiotics. In this paper the limitations of these a priori antibiotic susceptibilities are outlined and a method that can reduce some of the problems is proposed. Deriving the probabilities for antibiotic susceptibility from a bacteraemia database by the classical maximum likelihood method provides unreliable results when the number of cases is small. Representing a Bayesian approach, we propose hierarchical Dirichlet learning for learning susceptibilities of a pathogen, using data from a group of similar pathogens in the database. A three-fold crossvalidation of the results was performed for eight pathogens belonging to one specific pathogen group (Proteus) and 36 antibiotics. The reduced distances between the estimated susceptibilities in the learning set and the observed susceptibilities in the test set show the improvement in quality of the estimates provided by the Dirichlet estimator relative to the maximum likelihood estimator.

1 Introduction

At the onset of infection, the identity of the infecting pathogen(s) is usually not known, and the clinician must therefore consider the probabilities of the presence of a range of pathogens and weight these with the probability of the pathogen(s) being susceptible to the antibiotics considered for treatment.

Susceptibilities of bacteria to antibiotics differ between hospitals and estimation of susceptibilities from databases of *in vitro* susceptibilities must therefore be based on local data. The size of local databases is limited as susceptibilities change over time. This is aggravated by the observation that community acquired and hospital acquired infections are different enough in their susceptibilities to stratify the databases into these two categories. It is difficult to set a threshold for how large the sample should be to make the classical maximum likelihood estimate useful. If we consider a pathogen that has an estimated susceptibility of 70% to an antibiotic, then the standard deviation (SD), calculated based on the binomial distribution, of that estimate is 9% for a sample size N=25 and 5% for N = 100. So it is probably safe to conclude, that the lower limit for useful estimates is somewhere between N = 25 and N = 100.

This paper will explore a method for partially alleviating the problem associated with limitations of knowledge about antimicrobial susceptibilities. Technically, the method will be based on hierarchical Dirichlet learning [Andreassen *et al*., 2003; Heckerman *et al*., 1999; Filho and Wainer,, 2007; Cestnik, 1990], that allows a systematic approach to strengthening sparse data with educated guesses. The idea of using groups of pathogens will be explored. For example, it might be impossible to obtain enough isolates of *Proteus spp*., which is one of 8 members of the "Proteus group" of pathogens (see Table 1), to derive a reliable estimate for its susceptibility to a certain antibiotic. An educated guess, in the absence of enough data would be to assume that it resembles other members of the Proteus group in terms of susceptibility. The Dirichlet learning then provides a mechanism, that allows the susceptibility estimates for *Proteus spp*. to deviate from the susceptibilities of other bacteria belonging to the Proteus group, if and when data on the actual susceptibility of *Proteus spp*. to this antibiotic becomes available.

The potential benefit of this idea will be evaluated by applying the proposed method to a bacteraemia database. Preliminary results assessing whether our method improves the estimates, relative to the maximum likelihood estimates, will be shown.

2 Materials and methods

2.1 Database structure and maximum likelihood estimates

Prior probabilities used in the model were based on a bacteraemia database collected at Rabin Medical Center, Beilinson Campus, in Israel during 2002-2004. The bacteraemia database included 3350 patient and episode unique isolates among adult inpatients and contains only the clinically significant pathogens. The list of pathogens includes 156 entries. The list of antibiotics has 36 entries.

The bacteraemia database provides the counts of susceptibilities belonging to each pathogen for a range of antibiotics. For example, Table 1 shows the counts of susceptibility (M_{ij}) and the number of isolates tested (N_{ij}) for the antibiotics tobramycin and augmentin and eight hospital-acquired pathogens belonging to the Proteus group. The index i identifies the antibiotic (in this case tobramycin) and j identifies the pathogen (in this case one out of the 8 pathogens in the "Proteus group". Using these counts, maximum likelihood estimates (ML_{ij}) of susceptibility were calculated. For example, the ML estimate for the susceptibility of hospital acquired *Proteus spp.* to tobramycin and its SD were obtained as $ML_{ij} = M_{ij} / N_{ij} = 2/3 = 0.67$ and $SD=\sqrt{ML_{ij}(1-ML_{ij})/N_{ij}} = 0.27$

	tobramycin		augmentin	
Pathogen	M_{ij}	N_{ij}	M_{ij}	N_{ij}
Proteus spp.	2	3	1	3
Proteus mirabilis	39	49	29	50
Proteus vulgaris	1	1	1	1
Proteus penneri	2	2	1	2
Morganella spp.	0	0	0	0
Morganella morganii	19	20	9	19
Providencia spp.	4	10	0	10
Providencia stuartii	5	14	0	14
Sum of Proteus group	72	99	41	99

Table 1: The counts of susceptibility to tobramycin and augmentin for eight hospital acquired pathogens belonging to the Proteus group.

2.2 Hierarchical Dirichlet learning over groups of pathogens

Dirichlet learning is a Bayesian approach for estimation of the parameters in binomial (or polynomial) distributions. In this paper it will be assumed that a priori estimates of the parameters of the binomial distribution for susceptibility can be guessed from the susceptibilities averaged over pathogens that are assumed to be similar.

In the Treat project a decision support system for advice on antibiotic treatment has been constructed [Andreassen *et al*., 2005]. As part of this construction 40 such groups of pathogens with similar susceptibility properties have been identified, and these groups will be used as a starting point for this paper.

Assume that a group of n similar pathogens has been identified, the pathogens being indexed by $j \in \{1, ,\ldots, n\}$. On a number of occasions the susceptibility of these pathogens to a certain antibiotic (indexed by i) has been tested, $N_{i1},\ldots N_{in}$ times respectively, with the counts of susceptibility being $M_{i1,\ldots,}M_{in}$, respectively. The average susceptibility P_i of this group is:

$$P_i=\sum_{j=1}^{n} M_{ij} / N_i, \text{ where } N_i = \sum_{j=1}^{n} N_{ij} . \qquad (1)$$

The maximum likelihood estimator of susceptibility of a pathogen $ML_{ij} = M_{ij} / N_{ij}$ is now replaced by the Dirichlet estimator:

$$P_{ij} = (\beta_i + M_{ij}) / (\alpha_i + N_{ij}) , \qquad (2)$$

where β_i and α_i are imaginary counts, $\beta_i = \alpha_i * P_i$ representing positive outcomes in the binomial distribution and α_i representing the imaginary sample size, inherited from the pathogen group. Thus, α_i indicates how strong the confidence is in the a priori distribution of the parameters, and β_i / α_i can be used as the a priori estimate of the parameter of the binomial distribution, i.e. as an estimate of the susceptibility averaged over the pathogen group. We let all α_i assume the value A, except that we impose an upper limit on each α_i:

$$\alpha_i =\min (A, N_i) , \qquad (3)$$

since it is not reasonable to let the imaginary sample size α_i exceed the number of counts N_i actually available for the group. If A=0, then the Dirichlet estimate becomes equal to the maximum likelihood estimate.

In the next section it will be shown, that a "suitable" value for A can be determined empirically.

2.3 Evaluation of the quality of the estimates

To evaluate the quality of the estimates a three-fold cross-validation procedure is applied. The 3 years of data are divided into 3 periods, each containing data from one year. In turn, one of the 3 periods is designated as the test set and the other 2 periods are designated as the learning set and used for calculation of the estimators.

We wish to evaluate how well the Dirichlet estimator P_{ij}, calculated from the learning set, predicts F_{ij}, the observed frequency of susceptibility, calculated from the test set. F_{ij} is calculated as $F_{ij} = M_{ij} / N_{ij}$. For this purpose we define the distance measure:

$$\text{Dist} = \sqrt{\sum_{ij} (P_{ij} - F_{ij})^2 N_{ij} / N}, \qquad (4)$$

where $N=\sum_{ij} N_{ij}$.

This distance measure calculates the square distance between P_{ij}, and F_{ij}, weighted by the relative frequency of the pathogen.

The procedure followed in the 3-fold cross-validation described above is graphically illustrated in Figure 1.

Dist measures the averaged distance between the Dirichlet estimator from the learning set and the observed frequency in the test set. Since P_{ij} is a function of A (see eqs. (2)-(4)), Dist is also a function of A. The value of A, which minimizes Dist is the optimal size of the imaginary sample to be inherited from a pathogen groups to individual pathogens.

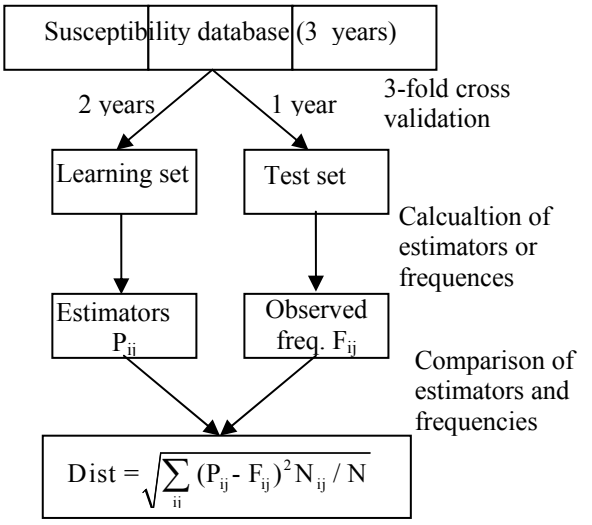


Figure 1: The procedure followed in the 3-fold cross-validation.

3 Preliminary results

3.1 Maximum likelihood estimates

Out of the 156 pathogens in the database, only 10 (6%) of these have been isolated more than 50 times. When community acquired and hospital acquired isolates are considered separately, 5% of pathogens in both cases have counts bigger than 50. The counts available for estimation of susceptibility are even smaller, because susceptibility is only tested for a selection of antibiotics. This indicates, that the maximum likelihood estimates of susceptibility for most combinations of pathogens and antibiotics in this database are too uncertain to be useful.

3.2 Hierarchical Dirichlet learning over groups of pathogens

To illustrate the method we shall focus on one of the groups of pathogens inherited from the Treat project. We chose the Proteus group mentioned above, which has 8 members (see Table 1).

To illustrate Dirichelt learning for a single pathogen, let us consider learning susceptibility of hospital-acquired *Proteus spp*. to tobramycin using susceptibility data available for other members of the Proteus group. (The procedure can be applied to any of eight pathogens in the Proteus group). First we assume a value for A, e.g. A = 4. This gives $\alpha_i = 4$, because for the Proteus group $N_i = 99$ (see Table 1). The average susceptibility of the group is $P_i = 72 / 99 = 0.728$. Next we calculate $\beta_i = \alpha_i * P_i = 4 * 0.728 = 2.91$. Finally we can calculate the Dirichlet estimator as $P_{ij} = (\beta_i + M_{ij}) / (\alpha_i + N_{ij}) = (2.91 + 2) / (4 + 3) = 0.70$.

This result along with the the maximum likelihood estimator and the Dirichlet estimator for the remaining members of the Proteus group are shown in Table 2, assuming that A =4.

Pathogen	ML_{ij}	P_{ij}
Proteus spp.	0.67	0.7
Proteus mirabilis	0.80	0.79
Proteus vulgaris	1	0.78
Proteus penneri	1	0.82
Morganella spp.	NA	0.73
Morganella morganii	0.95	0.91
Providencia spp.	0.4	0.49
Providencia stuartii	0.36	0.44
Sum of Proteus group	0.73	0.73

Table 2: The maximum likelihood estimates and the Dirichlet estimators of susceptibility to tobramycin for eight hospital acquired pathogens belonging to the Proteus group. (NA=not applicable)

An optimal value for A can be determined empirically by minimizing the distance in (4). We have applied the distance measure for tobramycin across the Proteus group (the summation in (4) was performed across one antibiotic and the eight pathogens in the Proteus group). It was found that the distance reaches its minimum ($Dist_{min}$ = 20.2%) at A = 4 (Figure 2a), which is therefore the optimal imaginary sample size to be used for calculation of the Dirichlet estimator. Note, that the maximum value of Dist ($Dist_{max}$ = 25.8%) is observed at A = 0 and corresponds to the Distance achieved by the maximum likelihood estimator. The improvement in quality provided by the Dirichlet estimator relative to the maximum likelihood estimator is $\Delta Dist = (Dist_{max} - Dist_{min}) / Dist_{min} = 28\%$.

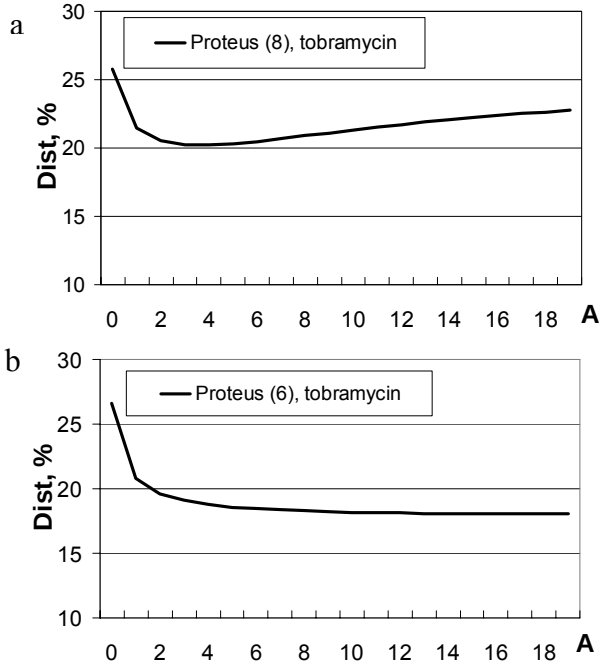


Figure 2: a: The distance measure Dist in percent for tobramycin and the Proteus group (containing 8 pathogens) as a function of A for hospital-acquired infections. b: Dist for tobramycin and the Proteus group, excluding 2 pathogens - *Providencia spp*. and *Providencia stuartii*.

This indicates that the maximum likelihood estimator is inferior to the Dirichlet estimator when applied for the

susceptibility to tobramycin for the Proteus group, but it does not guarantee that this applies to all members of the group and for other antibiotics.

It is apparent from Table 2, that the susceptibility to tobramycin for *Providencia spp.* and for *Providencia stuartii* is smaller for these two species than for the average of the group ($p<0.0001$). This implies that it may be advantageous to place *Providencia spp.* and *Providencia stuartii* in a group of their own.

If we allow *Providencia spp.* and *Providencia stuartii* to form their own group, labelled Providencia, then Dist for the remaining 6 members of the Proteus group becomes smaller, and it becomes possible to use a higher value of A, indicating that the group has become more homogeneous and the Dirichlet learning therefore more robust. (Figure 2b).

The formation of a separate Providencia group also improves Dist calculated for *Providencia spp.* and *Providencia stuartii* (Figure 3). This also applies when Dist is calculated across all the 8 pathogens in the original Proteus group (Figure 4). The Dirichlet estimators in Figure 4 were derived using a 6 member Proteus group and a 2 member Providencia group.

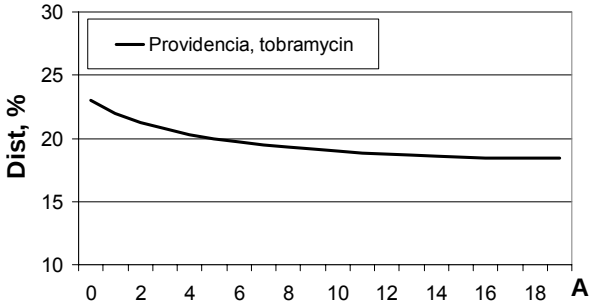


Figure 3: The distance measure Dist in procent for tobramycin and the Providencia group as a function of A for hospital-acquired infections.

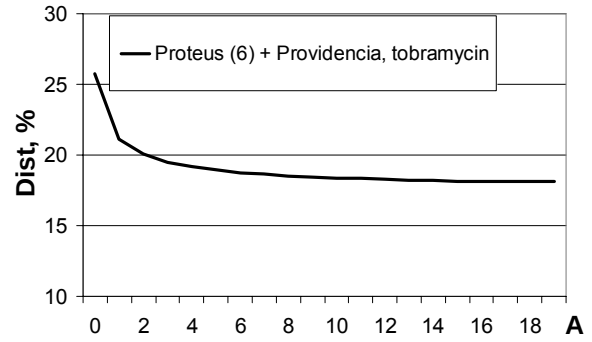


Figure 4: The distance measure Dist in procent for tobramycin and the Providencia group as a function of A for hospital-acquired infections. The Dirichlet estimators were derived using a 6 member Proteus group and a 2 member Providencia group.

Finally we turn our attention to the question of whether the pathogen groups are valid across all antibiotics or if they just apply to a single antibiotic. Augmentin is another antibiotic for which susceptibility is frequently tested in the Proteus group. Figure 5 shows Dist for augmentin and the 8 pathogens in the original Proteus group. The smooth curve in Figure 5 expresses the Dirichlet estimators derived using the original 8 member Proteus group, and the broken curve in the Figure 5 was used for the Dirichlet estimators derived using a 6 member Proteus group and a 2 member Providencia group. The results show that even with the original 8 member Proteus group, there is a small advantage to the Dirichlet learning with the value A = 1. When the Proteus group is split into two, then Dist becomes smaller and the optimal value of A becomes larger. This is qualitatively the same findings as for tobramycin.

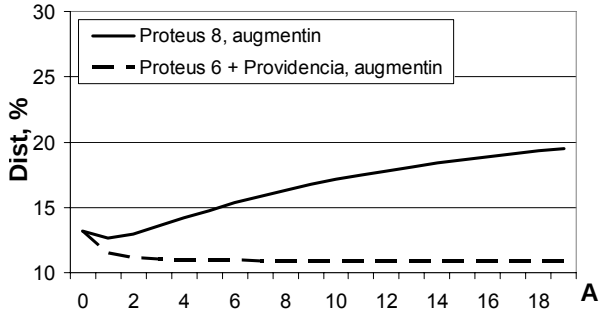


Figure 5: The distance measure Dist in procent for augmentin and the Proteus group as a function of A for hospital-acquired infections.

If we extend the analysis to all antibiotics for which susceptibility testing has been done, then this qualitative finding is confirmed, as illustrated by Figure 6.

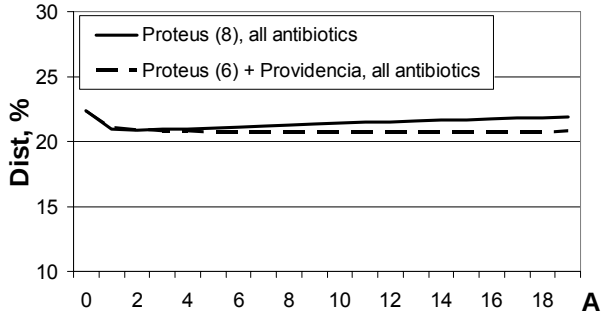


Figure 6: The distance measure Dist in procent for the Proteus group across all antibiotics as a function of A for hospital-acquired infections.

3 Discussion

The preliminary results show, that Hierarchical Dirichlet learning of susceptibility for a pathogen from the data available from a group of similar pathogens can provide estimates of susceptibility with improved quality, i.e. with smaller distance to the frequencies observed in the test set. For the Proteus group of pathogens the original grouping

of 8 pathogens, inherited from the Treat project, turned out to be robust in the sense that on average Dirichlet learning improved estimates across all antibiotics, showing that the formation of a Proteus group seems well founded. The results also showed that as might be expected, the Dirichlet learning works best when the pathogens within the group has a high degree of similarity. Splitting the Proteus group in two, a new smaller Proteus group and a Providencia group actually improved the Dirichlet estimates, not only for a single antibiotic, but averaged over all antibiotics. This again underlines that the concept of groups of pathogens seems to be well founded.

The results also showed that with better matching of the pathogens within a group, the Dirichlet learning could be strengthened, as reflected in the larger values of the size A of the imaginary sample, inherited from the group to the individual pathogens.

The evaluation of the Dirichlet learning is based on calculating the estimators from a two year learning periode and comparing them to a one year test period. Since our dataset only contained data from a three year period, the evaluation became a 3-fold cross validation. Often in cross validation studies a higher fold is preferred to provide higher statistical stability, but in this particular case the 3-fold validation may be an appropriate choice. This is due to the difficulties involved in the collection of a database of susceptibilities. Due to the steady decline of bacterial susceptibility to antibiotics over time, it is not realistic to use databases that cover much more than three year. Likewise it is not realistic to expect a much larger number of bacterial isolates per year, since this database comes from a large university hospital with a large throughput in the department of microbiology. Thus, databases of bacterial susceptibility are unlikely to be substantially larger than what has been used here. On this background it seems realistic to use a database of susceptibility to update the susceptibility estimates every year or every other year. Our choice in this study with a learning set covering a two year period corresponds to an updating of susceptibility estimates every other year.

In this paper we have only explored the properties of the Dirichlet estimators for a single group of pathogens. To produce useful results for clinical practice, the method must be repeated for all pathogen groups in the database.

References

[Andreassen *et al*., 2003] Steen Andreassen, Brian Kristensen, Alina Zalounina, Leonard Leibovici, Uwe Frank and Henrik Schønheyder. Hierarchical Dirichlét learning - filling in the thin spots in a database. In *Proceedings of the 9th Conference on Artificial Intelligence in Medicine*, pages 274–283, Cyprus, October 2003.

[Andreassen *et al*., 2005] Steen Andreassen, Leonard Leibovici, Mical Paul, Anders Nielsen, Alina Zalounina, Leif Kristensen, Karsten Falborg, Brian Kristensen,,Uwe Frank and Henrik Schønheyder. A probabilistic network for fusion of data and knowledge in clinical microbiology. In Husmeier, Dybowski, Roberts (eds.): *Probabilistic Modeling in Bioinformatics and Medical Informatics*, Springer, London, pages 451-472, 2005.

[Heckerman *et al*., 1999] Heckerman David. Tutorial on Learning With Bayesian Networks. In M. Jordan (ed): *Learning in Graphical Models,* MIT Press, Cambridge, MA 1999.

[Filho and Wainer, 2007] Jorge Filho and Jacques Wainer. Using a hierarchical Bayesian model to handle high cardinality attributes with relevant interactions in a classification problem. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 2504–2509, Hyderabad, India, January 2007.

[Cestnik, 1990] Bojan Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the 9th European Conference on Artificial Intelligence,* pages 147-149, Stockholm, Sweden, 1990.

Reusable Semantic Web-based Methods to Query Temporal Patterns: Application to Clinical Trials Management

Martin J. O'Connor, Ravi D. Shankar, and Amar K. Das
Stanford Medical Informatics
Stanford University
Stanford, California 94305, USA
martin.oconnor@stanford.edu

Abstract

Software applications supporting clinical trial planning and monitoring have significant requirements for knowledge management. Modeling the timing of various clinical trial activities is central to these requirements. Often, the encoded temporal patterns found in clinical trial applications may be imprecise and partial reflections of the intentions of the trial designers. To address this problem, we have developed an end-to-end knowledge-based system that permits formal design-time specification of temporal patterns and their automated verification when the system is deployed at clinical trial sites. In this paper, we discuss the use of the Semantic Web Rule Language (SWRL) as a general, reusable mechanism for encoding and executing these temporal patterns on relational databases. We present a set of ontologies and tools that we have developed for these efforts. We show how our approach supports participant and specimen tracking applications for clinical trials undertaken by the Immune Tolerance Network.

1 Introduction

Clinical trials encompass a vast array of formalized, controlled studies to advance and validate new approaches in the prevention and treatment of medical conditions. They require significant knowledge and information management at all stages, from initial planning through data analysis. To assist this process, we are building an ontology based framework to manage clinical trials. This work is driven by our collaboration with the Immune Tolerance Network (ITN), whose goal is to accelerate the development of new therapies for immune disorders [Rotrosen et al., 2002]. In collaboration with ITN, we have created a clinical trial ontology called Epoch [Shankar et al., 2006] to support the management of multi-site clinical trial protocols and the discovery of common tolerance mechanisms across multiple trials.

Reasoning with time-stamped data is central in these clinical trial systems. Trial design and compliance monitoring tasks, for example, typically revolve around specifying temporal patterns and evaluating them. Example patterns include: "Visit 3 for a participant must occur with 3 weeks of visit 2." "Clinical assessments are required twice a week until day 28 or discharge from hospital." "Test is scheduled on weeks 4, 6, and 8 during treatment."

These patterns are usually written as free text and are distributed throughout protocol design documents. Their interpretation is heavily dependent on the context of the protocol being encoded. This unstructured specification process can make it difficult to produce a precise definition for a pattern and can also result in significant gaps in the final specifications. As a result, implemented protocol temporal patterns may be an imprecise and partial reflection of the intentions of the designers and the quality of the trial data may become compromised. These shortcomings are often not noticed until the stage of final data analysis, when many may not be correctible. The consequences are frequently serious: lengthy and expensive data cleanup processes may be required, and some data may have to be discarded because of poor compliance.

2 Semantic Web Methods

To address this problem, we have developed an end-to-end system using the Semantic Web ontology and rules languages (OWL [OWL, 2004] and SWRL [SWRL, 2004], respectively) for design-time encoding of temporal patterns and their execution in a deployed clinical trial. As a first step, we developed a temporal ontology to provide a uniform representation of all temporal information in the Epoch clinical trial model. Using a SWRL development environment [O'Connor et al., 2005][1], we created a set of SWRL rules to encode temporal patterns in terms of the model. As a final step, we developed a mapping from ontology-level concepts to data stored in relational databases.

2.1 Temporal Ontology

We have adopted the valid-time temporal model as commonly used in temporal database research [Snodgrass 1995]. The valid-time model adds a time dimension to a piece of information, which is often referred to as a fact. Facts model one or more associated pieces of information and are analogous to tuples in relational databases. Each fact is considered to be atomic and is held to be true—or *valid*—for one or more times. These times, which can be

[1] http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

instants or intervals, are referred to as the valid-times of the fact and denote the time or times during which it is believed to be true. Using this temporal model, we have developed an OWL-based valid-time ontology, which is the basis for representing temporal knowledge in our system. Following the valid-time model, our temporal ontology allows temporal information to be associated with information that extends over time.

2.2 Temporal Rules

Once temporal information for a set of OWL classes has been consistently represented with the temporal ontology, we can write SWRL rules to temporally reason with information represented using that ontology. For example, the following SWRL rule determines the start dates of a series of treatments with the drug DDI:

```
Participant(?p) ^ hasTreatment(?p, ?t) ^
hasRegimen(?t, ?regimen) ^
swrlb:equals(?regimen, "DDI") ^
temporal:hasValidTime(?t, ?tVT) ^
temporal:hasStart(?tVT,?startTreatment) ->
hasDDIRegimenStart(?p, ?startTreatment)
```

2.3 Temporal Library

Most non trivial rules will require temporal operators. SWRL provides a very powerful extension mechanism that allows user-defined functions to be used in rules. These methods are called *built-ins* and are predicates that accept one or more arguments. A number of core built-ins are defined in the SWRL specification. This core set includes basic mathematical operators and built-ins for string and date manipulations. A few temporal built-ins are included in the current SWRL specification, but they have limited expressive power. To augment this limited set, we have defined an extensive set of temporal built-ins[2]. These built-ins allow the writing of rules that express complex temporal patterns.

2.4 Relational Database Mapping

The temporal model can be used to ensure that temporal information is represented consistently in a system, and SWRL rules can support knowledge level reasoning with this information. However, most data—particularly medical data—will continue to reside in relational databases. To reason with such data using knowledge-based tools, one could map all such relational data to equivalent OWL concepts. While this approach may be appropriate when working with small data sets, it does not scale to significant amounts of data, and for these data sets, an alternate solution is needed.

To support knowledge-driven querying of relational databases, we have developed tools to map data dynamically from relational databases to concepts described in an OWL ontology [O'Connor et al, 2007]. Our tools make extensive use of SWRL to specify the OWL to relational mapping and to provide a knowledge level query interface to the system. We have devised an array of optimization strategies to improve the performance of the underlying relational-to-ontology mapping process. Our primary goal is to offload as much work as possible to the underlying RDBMS by exploiting knowledge of SWRL rules as well as additional information provided by a rule base author. A secondary goal is to reduce the amount of data retrieved from databases during rule processing. These strategies in conjunction with the temporal ontology and associated reasoning mechanism provide an approach to integrate low-level relational data with knowledge-level domain concepts and allows knowledge-level reasoning with clinical trial data.

3 Discussion

The gap between temporal pattern specification and execution is often significant in clinical trial systems. To help close this gap, we developed a system for formally specifying temporal patterns and executing them in terms of this specification. Our system takes temporal patterns that are encoded at the domain level at design time and translates them into an executable form. At run time these patterns operate directly on trial data held in relational databases. The system uses OWL to provide a uniform knowledge model that integrates the temporal representations of relational data with the domain-specific semantics of the temporal patterns used to reason with it. We used SWRL rules written in terms of concepts in this model to express patterns within the tracking application. We are using this system in the development of a visit and specimen tracking application for the Immune Tolerance Network.

References

[OWL, 2004] http://www.w3.org/TR/owl-ref/

[O'Connor et al., 2005] O'Connor M.J., Knublauch, H., Tu, S.W., Grossof, B., Dean, M., Grosso, W.E., Musen, M.A. Supporting Rule System Interoperability on the Semantic Web with SWRL. Fourth International Semantic Web Conference (ISWC2005), Galway, Ireland (2005).

[O'Connor et al., 2007] O'Connor, M.J., Shankar, R.D., Tu, S.W., Nyulas, C., Musen, M.A., Das, A.K. Using Semantic Web Technologies for Knowledge-Driven Querying of Biomedical Data. 11th Conference on Artificial Intelligence in Medicine (AIME07), Amsterdam, Netherlands, 2007.

[Rotrosen et al., 2002] Rotrosen, D., Matthews, J.B., Bluestone, J.A. The Immune Tolerance Network: a New Paradigm for Developing Tolerance-Inducing Therapies. J Allergy Clinical Immunology, 110(1):17-23 (2002).

[Snodgrass 1995] Snodgrass, R.T. The TSQL2 Temporal Query Language, Boston, MA: Kluwer, (1995).

[SWRL, 2004] http://www.w3.org/Submission/SWRL/

[Shankar et al., 2006] Shankar, R.D., Martins, S.B., O'Connor, M.J., Parrish, D.B., Das, A.K. Towards Semantic Interoperability in a Clinical Trials Management System. Fifth International Semantic Web Conference (ISWC2006), Athens, GA (2006).

[2] http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTemporalBuiltIns

An Extensible Software Framework for Temporal Data Processing

Cristiana Larizza[1] and Paolo Ciccarese[1,2]
[1]Laboratory for Bio-Medical Informatics, University of Pavia
Via Ferrata 1, 27100 Pavia, Italy
[2]Harvard Medical School, Charlestown, MA 02129
cristiana.larizza@unipv.it, paolo.ciccarese@gmail.com

Abstract

This paper presents the more recent version of Tempo, a framework for the definition, generation and execution of data processing components combining one ore more pipelines of default/custom modules assembled according to a specific meta-model. Although it has been initially tested in the medical field, Tempo is conceived as a general purpose framework.

1 Introduction

Nowadays, many application domains (financial, scientific, medical and so on) require the collection and processing of huge quantities of temporal data for different purposes. In past literature several tools for temporal data processing have been described [Augusto, 2005], [Boaz and Shahar, 2005], [Hunter, 2006]. The difficulty in identifying universal procedures for analyzing temporal data is due to their very different characteristics requiring the adoption of specific techniques, customized on the basis of the context and on the goals of the analysis. Moreover, very frequently, in order to make the inspection and the processing of temporal information easier and more efficient, it is useful to transform raw temporal data into series of patterns that summarize their evolution. In the following we will present Tempo, a general purpose framework for the definition of components in which different kind of reusable blocks can be assembled into pipelines or combination of pipelines. Each block wraps data filtering or other data processing/visualization algorithms. Tempo already includes a set of reusable blocks for data filtering and temporal patterns extraction based on the artificial intelligence technique named Temporal Abstractions (TAs) [Shahar, 1997], but new algorithms embedded into custom modules can be added to the provided library as plug-ins. The paper describes the latest Tempo release which includes new kind of blocks to be combined into components as will be better explained through an example of application of Tempo to a medical context.

2 The Tempo Model

Our effort within Tempo project has been to propose a data processing model sufficiently flexible and extendible to be applied to different kind of data and contexts.

Tempo components are build around the pipeline concept as already described in [Ciccarese and Larizza. 2006]. The pipeline can be composed by different kind of blocks:
- *Filters*: which give an output defined within the same metric space of the input. Example can be given by blocks embedding a filtering of the outliers in a time series as well as noise reduction algorithm for images;
- *Transformers*: in which the output metric space is different from the input one. Examples can be a mechanism for qualitative abstraction, defined as quantitative data mapping into symbolic values, as well as the transformation at a different colour depth of an image.
- *Boxes*: blocks embedding a sub-pipeline, thus, a series of blocks. This is particularly useful for fostering reuse not only of blocks, but also of already defined sequences of blocks or pipelines.

A descriptor belonging to each block explains which data the block can accept as input and which data can provide as output, as well as the set of accepted/needed parameters. A new feature of Tempo is the capability of processing multiple data streams by means of another kind of blocks:
- *Aggregtors* (see Figure 1) that accept as input the data coming from two different pipelines and give as output a single data stream derived from the application of an operator tuned, as usual, through a set of parameters.

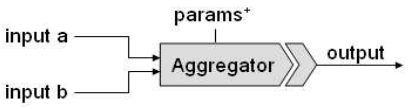


Figure 1. The Aggregator block which implements an operator through a parameterization in P.

When pipelines or pipelines aggregations have been defined, they can be transformed into Tempo *Components* that represent complete data processing elements. A *Component*, when loaded by the Tempo engine, is able to fetch data to feed and run the pipeline and store its results. This is possible by adding to the pipeline two further blocks:
- *Generators*: that provide the input to the pipeline out of binary data (xml, images, text, html, zip) or other kind of sources such as specific tables in a relational database;
- *Serializers*: that represent the end of the pipeline and transform the pipeline output stream into data in different

binary formats (xml, images, text, html, zip, pdf) or into tables of a relational database.
Moreover, in order to be able to inspect data flowing from one block to another of the component, it has been defined the concept of

- *Inspectors*: that provide numeric or graphical views of the data flowing in the pipeline and of the performed abstractions. These blocks don't perform any filtering nor transformation of data, but simply generate a view of the pipeline content.

3 The Core Modules and Applications

Tempo includes a library of reusable blocks that can be assembled to define processing components customized according to the needs. They provide some standard filtering algorithms and a set of mechanisms for the temporal patterns detection (as transformers or aggregators) performed through TAs. When executing a component, its input contains all the parameters needed for the computation (including the configuration setup of Generators and Serializers).
The components generated through the Tempo framework have been already integrated into different applications both under the form of java libraries and web services. Libraries have been already integrated in a case-based retrieval system to support the treatment of end stage renal failure patients [Montani *et al*., 2006] and in a general purpose web application. Web services have been adopted in the Guideline Management System belonging to the Guide project [Ciccarese *et al*. 2005].

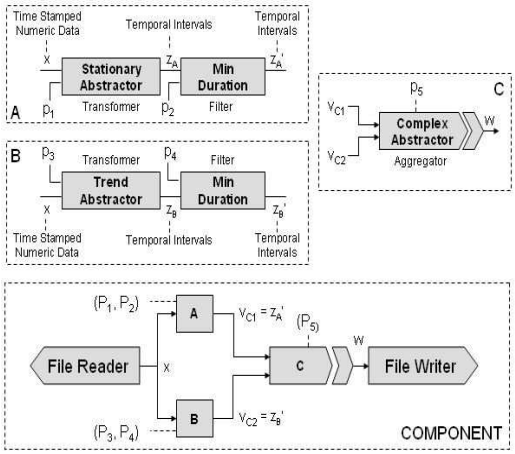


Figure 2. A Tempo component example.

In order to explain the structure of the components let's consider as example the detection of the following pattern: "*heart rate increase for at least 4 minutes immediately followed by stationary heart rate lasting at least 3 minutes*". To detect such pattern it is necessary to define two pipelines and an Aggregator which will be managed by the purposely deployed Tempo component depicted in Figure 2. According to such figure, both pipelines A (detection of *stationary heart rate lasting at least 3 minutes*) and B (detection of *heart rate increase for at least 4 minutes)* accept a time series and combine a Transformer (Stationary and Trend Abstractors respectively) and a Filter for short episodes removal. Aggregator C, implementing a Complex TA [Ciccarese and Larizza. 2006], accepts two interval series (the output of pipelines A and B) and provides the tuples of intervals related through the chosen Allen temporal operator (in this case the MEETS operator to detect the output patterns of pipeline A *immediately after* the output patterns of pipeline B). The final structure of the component is depicted in the bottom of Figure 2. By changing the configuration of the FileReader and the FileWriter, it is possible to run the same component on different data sets and obtain different kinds of output.

4 Conclusions

The paper describes the new release of the Tempo framework for temporal data processing and abstractions. The added value of its architecture are the possibility of composing the data analysis procedures as sequences or combinations of building blocks and the possibility of enriching the available library embedding custom algorithms in new modules developed by third parties through the provided Tempo API. Current version is going to include a graphical tool for a fast definition, validation and deployment of own components.

References

[Shahar, 1997] Yuval Shahar. A framework for knowledge-based Temporal Abstraction. *Artificial Intelligence, 9*0 (1997) 79-133.

[Augusto, 2005] Temporal reasoning for decision support in medicine, *Artificial Intelligence in Medicine* (2005) 33, 1—24

[Boaz and Shahar, 2005] David Boaz Yuval Shahar. A Framework for Distributed Mediation of Temporal-Abstraction Queries to Clinical Databases. *Artificial Intelligence in Medicine*, Volume: 34, Issue: 1, May, 2005, pp. 3-24

[Hunter, 2006] James R W Hunter. TSNet – A Distributed Architecture for Time Series Analysis Niels Peek, Carlo Combi (ed), *In Proc. Intelligent Data Analysis in bioMedcine and Pharmacology (IDAMAP 2006)* (Verona, Italy): pages 85-92

[Ciccarese and Larizza. 2006] Paolo Ciccarese and Cristiana Larizza. A Framework for Temporal Data Processing and Abstractions. *In Proc. AMIA 2006*, p.p. 146-150, (2006).

[Montani *et al*., 2006] S. Montani *et al*., Case-based retrieval to support the treatment of end stage renal failure patients. *Artificial Intelligence In Medicine,* Volume: 37, Issue: 1, May, 2006, pp. 31-42.

[Ciccarese *et al*., 2005] Paolo Ciccarese, Ezio Caffi, Silvana Quaglini and Mario Stefanelli. Architectures and Tools for innovative Health Information Systems: the Guide Project. *International Journal of Medical Informatics*, ed Marius Fieschi, Mario Stefanelli and Casimir A. Kulikowski, vol. 74, (2005) 553 - 562.

SPOT – Utilizing Temporal Data for Data Mining in Medicine

Guenter Tusch[1,2], Martin O'Connor[1], Timothy Redmond[1], Ravi Shankar[1],and Amar Das[1]
[1] Stanford Medical Informatics, Stanford University, Stanford, CA, USA
[2]Grand Valley State University, Allendale, MI, USA
tuschg@gvsu.edu, {martin.oconnor,tredmond,rshankar,das}@stanford.edu

Abstract

Mining large clinical databases often includes exploration of temporal data. For example, in liver transplantation, where parameters are obtained from continuously monitored patients, a researcher might be interested in patients that exhibit an unusual pattern of potential complications of the transplanted organ, each following a typical pattern in time. Standard query languages like SQL are not well suited for this kind of research because of an insufficient time model. A very flexible approach is Knowledge-based Temporal Abstraction, which has been implemented in a number of proprietary systems. Here time-stamped data points are transformed into an interval-based representation that can utilize, e.g., Allen's temporal relationships. For increased availability in clinical research, we extended the knowledge-based temporal abstraction framework by creating an open-source platform, SPOT. It supports the R statistical packages and knowledge representation standards (OWL, SWRL) using the open source Semantic Web tool Protégé-OWL.

1 Introduction

Modern researchers in medicine have access to large clinical databases that have become more readily available recently. One important aspect of data mining those resources is the exploration of temporal data. For example, in liver transplantation, where a wealth of parameters is obtained from continuously monitored patients, a researcher might be interested to select patients or patient episodes that exhibit an unusual pattern of potential complications of the transplanted organ, each following a typical pattern in time. Standard query languages like SQL are not well suited for this kind of research because of an insufficient time model. A very flexible approach is Knowledge-based Temporal Abstraction (KBTA), which has been implemented in a number of proprietary systems. Here time-stamped data points are transformed into an interval-based representation that can utilize, e.g., Allen's approach of temporal relationships [Augusto, 2005]. To make KBTA more readily available for clinical research, we developed SPOT, an implementation using open source and standardized tools: the Web Ontology Language (OWL; `http://www.w3.org/TR/owl-features`), the Semantic Web Rule Language (SWRL; `http://www.daml.org/2003/11/swrl`), Protégé plug-ins; `http://protege.stanford.edu/`, and open source statistical software (R; `http://www.r-project.org/`).

1.1 Medical Example

Liver transplantation is a complex and challenging surgical procedure that is followed by a complex intensive care and clinical monitoring schedule. Potential hepatic complications can be acute or chronic, each following a typical pattern in time. For instance, "acute rejection" can be characterized by increasing AST and ALT (liver enzymes) values, which decrease as soon as the rejection therapy is started. If there is no response to the therapy, it is not considered rejection. The phase with increasing enzymes may vary in length and range of values, or even only one enzyme may be elevated, same for the phase of decreasing values, but still the time pattern holds. In this example a few typical issues are addressed. First, clinical data come with different time granularities, hourly, daily, monthly or yearly. Second, clinical concepts can be expressed in terms of phases or intervals, e.g. increasing or decreasing enzymes, rejection therapy over 3 days, which can be consecutive or overlapping, and establish a typical pattern in time. Third, it is not so much single parameter values but the relationship of intervals that establishes the clinical concept. These aspects are captured in the valid time model as used in temporal database research and in temporal abstraction.

2 Knowledge Based Temporal Abstraction

KBTA is a comprehensive approach to deal with time-oriented data in medicine. A functional approach is used that maps raw data into higher-level concepts like "states" (e.g. peak, rejection therapy) or "trends" (e.g. increasing, decreasing). The goal is to represent complex medical concepts by these primitives using time relationships. [Shahar and Musen, 1996] introduce the KBTA method as a formal model of input and output entities, their relations, and the domain-specific properties that are associated with these entities - called the KBTA ontology. Shahar and Musen describe four different output types: state, gradient, rate, and pattern abstraction. States could be low or high

bilirubin levels of the transplanted liver, gradients increasing or decreasing enzymes, rates could be slow or fast, and a pattern periodic.

There are different implementations of the KBTA method all over the world. Almost all of them focus on describing individual patient courses for clinical therapeutic purposes. An overview on current implementations is found in [Augusto, 2005].

3 SPOT – Architecture and Implementation

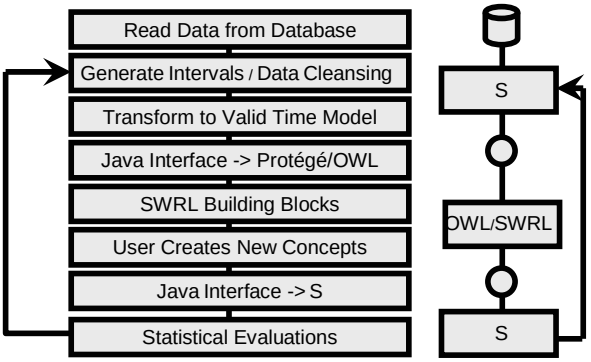


Fig. 1. The SPOT (***S*** – ***P***rotégé – ***O***WL/SWRL – ***T***emporal Abstraction) architecture.

An overview over the SPOT architecture is depicted in figure 1. The researcher (user) can define clinical concepts, e.g., rejection, and then search for patients or episodes with that concept in the clinical database. Two steps are necessary to accomplish this: Training the system to learn concepts from a subset of the clinical database, and searching for the learned concepts in the entire database. The user has to perform the following tasks in order to train the system: Estimation of intervals from a learning sample, e.g., learning thresholds for a running average of the ALT parameter values to model an "increasing ALT" interval, (implemented in S), building of high level concepts (Temporal Abstraction) (implemented in Protégé/OWL/SWRL), and validation of the generated intervals (implemented in S). The user might go through that process several times until the classification error for the clinical concepts he/she models is sufficiently small. Adjustments can be made by changing thresholds or adding additional constraints to the SWRL concepts. Finally, the learned abstractions are submitted to the original database.

Besides using the time stamped data from the clinical database, the user needs to identify intervals, only one parameter at a time (e.g., AST). Several different non-overlapping intervals are allowed, i.e. mark as "increasing", "decreasing", "high", etc. for AST. The interval value is attached to the time-stamped parameter value.

SPOT supports the statistical package R and knowledge representation standards using the open source Semantic Web tool Protégé-OWL. Ontologies are used in OWL to formally specify meaning of annotations by providing a vocabulary of terms. New terms can be formed by combining existing ones. SWRL allows users to write rules that can be expressed in terms of OWL concepts and that can reason about OWL individuals. In the liver transplantation application, we use a temporal ontology implementing the valid time model, and a hierarchical patient ontology with classes: Patient (*has*) Procedure (*has*) Interval/Event (*has*) Valid Time (see figure 2 for an example).

S is an interactive environment for data analysis and at the same time a statistical programming language. R is an open source implementation of S. The Protégé OWL plug-in allows to building ontologies backed by OWL code.

```
Patient(?p) ∧
hasProcedure(?p, ?proc)  ∧
   hasTest(?proc, ?test)  ∧
   hasTestName(?test, ?testName)  ∧
   swrlb:equal(?testName, "BILIRUBIN") ∧
   HasOutputType(?test, ?testType)  ∧
   swrlb:equal(?testType, "INCREASE")  ∧
   temporal:hasValidTime(?test, ?tVT)  ∧
hasTest(?proc, ?test2) ∧
hasTestName(?test2, ?testName2)  ∧
swrlb:equal(?testName2, "BILIRUBIN") ∧
HasOutputType(?test2, ?testType2)  ∧
swrlb:equal(?testType2, "HIGH")  ∧
temporal:hasValidTime(?test2, ?tVT2) ∧
   temporal:overlaps(?tVT, ?tVT2, "days")∧
temporal:hasStartTime(?tVT, ?stTime)  ∧
temporal:hasFinishTime(?tVT, ?fiTime)∧
swrlx:createOWLThing(?hbVT, ?proc)
->temporal:ValidPeriod(?hbVT) ∧
temporal:hasStartTime(?hbVT,?stTime)∧
temporal:hasFinishTime(?hbVT,?fiTime)∧
   hasHighBiliIncrease(?proc, ?hbVT)
```

Figure 2. SWRL Code for the concept of "*High and Increasing Bilirubin*" (?tVT, ?tVT2, and ?hbVT are interval instances)

4 Discussion and Future Aspects

The reported research shows that SPOT is a feasible approach to use open source and standards based software. One challenge is the "translation" of logically represented concepts back into the statistical environment (R). Currently, concept intervals are passed from OWL/SWRL through the Java interface and "relearned" through a classification tool in R, e.g., discriminant analysis. The next step is the development of a GUI using the R and Protégé APIs for easy access and manipulation by the user.

Acknowledgments

Tania Tudorache, PhD, Jeremy Miller, PhD, Craig Webb, PhD, and Mark Musen, MD, PhD.

References

[Augusto, 2005] Augusto, J.C.: Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33: 1-24, 2005.

[Shahar and Musen, 1996] Yuval Shahar and Mark Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine*, 8(3): 267-98, 1996.

Temporal Rules to Predict Renal Flares in Lupus Nephritis

Lucia Sacchi[1], Riccardo Bellazzi[1], Silvana Quaglini[1], Alberto Sinico[2], Gabriella Moroni[3]
[1] Laboratory for Biomedical Informatics, University of Pavia, Pavia, Italy, lucia.sacchi@unipv.it
[2] Division of Nephrology and Dialysis, Ospedale San Carlo Borromeo, Milano, Italy
[3] Division of Nephrology and Dialysis, IRCCS Ospedale Maggiore, Milano, Italy

Abstract

Lupus nephritis is one of the most severe complications of Systemic Lupus Erythematosus (SLE), and it is characterized by acute episodes known as *flares*. Within the available tests for the assessment of disease activity, the evaluation of antiC1q antibodies was recently found to be the most powerful to confirm flare diagnosis. In this paper we evaluate whether it is possible to extract temporal rules to relate four clinical parameters, used to monitor the disease activity, to the occurrence of renal flares. Such rules could be very useful in clinical practice in order to prevent acute episodes. To this aim, we applied an algorithm for temporal association rules extraction on a data set of 228 patients affected by Lupus nephritis and periodically monitored at our two hospitals in Milan, Italy. From the extracted rules antiC1q results to be the most important parameter to indicate the risk of renal flare.

1 Introduction

Lupus nephritis is one of the most frequent and severe complications of Systemic Lupus Erythematosus (SLE). SLE is a chronic autoimmune disease, most common in women of childbearing age, which involves several parts of the body, including a number of organs and systems. Lupus nephritis course is characterized by acute episodes of illness (known as *flares*) and remissions, which are usually induced by the immunosuppressive therapy. In [Moroni *et al.*,1996], flares were shown to be predictive of negative prognosis of the disease, since they are correlated with the development of chronic renal failure, which can ultimately lead the patient to death.

In clinical practice, flare diagnosis is typically based on some specific criteria, which usually include: 30% increase of plasmatic creatinine, proteinuria manifestation or worsening and/or hematuria manifestation or worsening. Moreover, a number of biohumoral tests are available to assess disease activity, among which we recall: C3 and C4 complement fractions, anti-DNA antibodies and antiC1q antibodies. Despite the fact that C3, C4 and anti-DNA are the most used tests to determine disease activity, several studies demonstrated they are not always reliable [Moroni *et al.*, 2001]; some patients were in fact found to show flares even in the presence of normal test values, and vice versa. Rather interesting, in [Moroni *et al.*, 2001] we showed that antiC1q turns out to be the most reliable test to confirm the presence of a flare when compared to C3, C4 and anti-DNA.

Besides flare diagnosis, also flare prediction is of crucial importance in clinical practice, in order to prevent acute renal episodes in SLE patients. In this paper we will evaluate whether it is possible to extract temporal rules to relate four parameters used for disease activity monitoring to the occurrence of renal flares. In particular, we will evaluate whether variations in C3, C4, anti-DNA and antiC1q values during periodical clinical evaluations are frequently temporally related to the occurrence of flares.

2 Material and Methods

2.1 Data

In this work we consider data coming from a group of 228 patients, all affected by Lupus nephritis at different stages and undergoing periodical, albeit not regular, clinical monitoring. For each patient, a set of five time series had been collected, four of them recording the parameters values (C3, C4, anti-DNA, antiC1q), and the other one describing renal disease status (complete remission, partial remission, acute flare, post-flare activity) at each clinical evaluation. The length of time series may vary among patients, depending on the number of clinical checkups each one underwent. Patients with only one measurement were eliminated from the data set, giving origin to a final set of 172 patients, with an average number of 9 measurements per patient.

2.2 Temporal Rules for Renal Flare Prediction

In order to establish whether a variation in one (or more) of the four considered parameters could help clinicians in the prediction of an acute renal episode, it is interesting to evaluate if specific variations into the variables time course are frequently temporally related to the occurrence of flares; to this aim, we chose to resort to an algorithm for temporal rules extraction, which is able to deal with the search for relationships between complex qualitative patterns detected in time series data [Sacchi *et al.*, 2007].

The proposed method enables the user to define patterns of interest, e.g. an increase in a variable lasting for at least

three measurements, thus synthesizing the domain knowledge about a specific process; it is therefore well-suited to deal with the kind of clinical problem at hand. Interesting patterns can be conveniently extracted from the rough quantitative data through the formalism of knowledge-based Temporal Abstractions (TAs) [Shahar, 1997], a technique which allows the description of temporal data in terms of a qualitative and interval-based representation.
From a clinical viewpoint, it was in our case interesting to look for temporal relationships between renal disease activity and an increase or a shift from normal to pathological values in one of the monitored parameters.
To represent shifts from normal to pathological values all the variables were described in terms of *state* TAs, while to detect an increase in the parameters we resorted to a *trend* temporal abstraction representation of the four monitored variables. Starting from this TA representation, we then run on our data an algorithm for the extraction of temporal rules expressing precedence relationships between the detected temporal patterns. Denoting for instance normal values with *N* and pathological values with *P*, an example of such a rule could be "A shift from *N* to *P* in antiC1q PRECEDES a renal flare", where PRECEDES indicates the temporal operator which relates the antecedent to the consequent of the rule. The exploited algorithm implements a search strategy based on an Apriori-like technique, where the quality of a rule is assessed in terms of confidence and support, whose definition had been properly adapted to deal with the temporal domain [Bellazzi *et al.*, 2005].

3 Results and Discussion

Table 1 shows the results obtained by running the rules extraction algorithm on our data set, fixing a threshold for the confidence *min_conf* = 0.4 and for the support *min_sup* = 0.05. In the rules, which are detailed in the following, p_i indicates any of the four monitored parameters:

– "A shift from *N* to *P* in p_i PRECEDES a renal flare"
– "An increase in p_i PRECEDES a renal flare"
– "A remission PRECEDES a shift from *P* to *N* in p_i"

As it can be observed, in the first two rules we want to investigate if a variation in one of the parameters is found to frequently precede a renal flare, while in the third one we evaluate whether a shift from pathological to normal values in a variable frequently occurs after a renal remission.
As it can be noticed from the results obtained, for any of the considered rules, the parameter that shows the best performance is antiC1q. Namely, when the value of antiC1q shifts from normal to pathological ranges in two consecutive evaluations, in the 55% of the cases a renal flare is diagnosed in one of the following controls. Moreover, an increase in the value for antiC1q predicts the occurrence of a renal flare in the 51% of the cases. Eventually, in the 50% of the cases, antiC1q was found to go back to normal values at the achievement of renal remission. No such a behavior was observed for the other parameters, since no rules were extracted by the algorithm in the other cases.
These results suggest that antiC1q, besides being an important parameter to confirm flare diagnosis, results also an indicator for flare prediction. Even if confidence and support assume relatively small values, they are considered significant by the medical experts. In clinical practice, patients verifying the obtained rules will thus undergo more frequent monitoring of the biochemical parameters related to flares.

Rule: A shift from *N* to *P* in p_i PRECEDES a renal flare			
Antecedent	**Consequent**	**Conf (95%CI)**	**Support**
C4	Renal Flare	0.42 (0.35-0.49)	0.2
anti-DNA	Renal Flare	0.44 (0.36-0.52)	0.24
antiC1q	Renal Flare	0.55 (0.46-0.64)	0.17
Rule: An increase in p_i PRECEDES a renal flare			
Antecedent	**Consequent**	**Conf (95%CI)**	**Support**
anti-DNA	Renal Flare	0.48 (0.38-0.58)	0.16
antiC1q	Renal Flare	0.51 (0.41-0.61)	0.2
Rule: A remission PRECEDES a shift from *P* to *N* in p_i			
Antecedent	**Consequent**	**Conf (95%CI)**	**Support**
Remission	antiC1q	0.5 (0.37-0.63)	0.08

Table 1. Rules extracted on the Lupus nephritis data set.

4 Conclusions

In this paper we presented an analysis on the extraction of temporal rules to determine whether it is possible to predict acute renal episodes in Lupus nephritis patients, on the basis of four clinical monitoring parameters. We obtained promising results especially on one of the variables, the antiC1q. Future work will be directed to the inference of the expected time of the next acute event given variations in the monitoring parameters.

References

[Bellazzi et al., 2005] Riccardo Bellazzi, Cristiana Larizza, Paolo Magni, and Roberto Bellazzi. Temporal data mining for the quality assessment of hemodialysis services. *Artificial Intelligence in Medicine*, 34(1):25-39, 2005.

[Sacchi et al., 2007] Lucia Sacchi, Cristiana Larizza, Carlo Combi, Riccardo Bellazzi. Data mining with Temporal Abstractions: learning rules from time series. *Data Mining and Knowledge Discovery*, 2007, to appear.

[Shahar, 1997] Yuval Shahar. A framework for knowledge-based temporal abstraction. *Artificial Intelligence*, 90(1-2): 79-133, 1997.

[Moroni et al., 1996] G. Moroni, S. Quaglini, M. Maccario, G. Banfi and C. Ponticelli. "Nephritic flares" are predictors of bad long-term renal outcome in lupus nephritis. *Kidney Inten* 1996;50:2047-2053.

[Moroni et al., 2001] Gabriella Moroni, M. Trendelenburg, N. Del Papa, S. Quaglini, E. Raschi, P. Panzeri, C. Testoni, A. Tincani, Giovanni Banfi, G. Balestrieri, J.A. Schifferli, P.L. Meroni, Claudio Ponticelli. Anti-C1q antibodies may help in diagnosing a renal flare in lupus nephritis. *Am J Kidney Dis* 37;3:490-498, 2001.