IDAMAP 2006

INTELLIGENT DATA ANALYSIS IN BIOMEDICINE AND PHARMACOLOGY

Niels Peek and *Carlo Combi* (chairs)

August 25-26, 2006

Department of Computer Science, University of Verona, Italy

Organized in collaboration with:

International Medical Informatics Association
Intelligent Data Analysis and Data Mining Workgroup

American Medical Informatics Association
Knowledge Discovery & Data Mining SIG

University of Verona
Department of Computer Science
Faculty of Science

IDAMAP 2006
Intelligent Data Analysis in bioMedicine And Pharmacology

Niels Peek and Carlo Combi (chairs)

1. Introduction

Welcome to IDAMAP-2006, the eleventh workshop on intelligent data analysis in biomedicine and pharmacology, hosted by the Department of Computer Science of the University of Verona, Verona, Italy. This is the first IDAMAP workshop, to last more than one day: IDAMAP-2006 consists of one afternoon (on Friday, August 25) and one full day (on Saturday, August 26).

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 methods for handling temporal data.

Gathering in an informal setting, participants have the opportunity to meet and discuss selected technical topics in a deep way: indeed, ample time is allotted for informal discussion among the participants. All participants are invited to join the workshop dinner on Friday, August 25.

2. Program

The scientific program of the workshop consists of presentations of accepted scientific papers, two invited presentations, a panel, and demonstrations of data analysis tools. More particularly, we will have 11 long presentations, 5 short presentations, and a long presentation within a panel. 5 demos enrich the workshop with a more application-oriented focus. We have also two invited talks. We are happy to have Amar Das from Stanford University and Xiaohui Liu from Brunel University accepting to act as invited speakers.

3. Program Committee

- Ameen Abu-Hanna, Academic Medical Center, Amsterdam, The Netherlands
- Riccardo Bellazzi, University of Pavia, Italy
- Carlo Combi, University of Verona, Italy
- 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
- Silvia Miksch, Vienna University of Technology, Austria
- Lucila Ohno-Machado, Harvard Medical School and M.I.T., Boston, USA
- Niels Peek, Academic Medical Center, Amsterdam, The Netherlands
- Paola Sebastiani, Boston University School of Public Health, USA
- Marco Ramoni, Harvard Medical School, Boston, USA
- Yuval Shahar, Ben-Gurion University of the Negev, Israel
- Stephen Swift, Brunel University, UK
- Allan Tucker, Brunel University, UK
- Adam B. Wilcox, University of Utah, USA
- Blaz Zupan, University of Ljubljana, Slovenia

4. Organization Committee

- Carlo Combi, University of Verona, Italy
- Barbara Oliboni, University of Verona, Italy

5. Acknowledgements

We would like to thank the invited speakers, the authors of papers and demos, the members of the program committee, Barbara Oliboni and all the people helping in the organization of the workshop. Finally, we are grateful to the Department of Computer Science and to the Faculty of Science, University of Verona, and to AMIA for their financial support and sponsorship of IDAMAP 2006.

IDAMAP 2006
Intelligent Data Analysis in bioMedicine And Pharmacology

Schedule

Friday, August 25, 2006

2:00 pm **Opening of IDAMAP Workshop**
Niels Peek and Carlo Combi

2:15 pm **Invited presentation**

Amar Das — *Page 3*
Knowledge-Driven Querying of Time-Oriented Biomedical Data

3:00 pm **Paper session: *Temporal Data Mining***

*** *F Guil, JM Juarez, R Marin* — *Page 7*
Mining Possibilistic Temporal Constraint Networks: A Case Study in Diagnostic Evolution at Intensive Care Units

*** *L Sacchi, M Verduijn, N Peek, E de Jonge, B de Mol, R Bellazzi* — *Page 13*
Describing and Modeling Time Series based on Qualitative Temporal Abstraction

*** *S Badaloni, M Falda* — *Page 17*
Mining Temporal Characterization of Ill-known Diseases

4:15 pm **Break** (including demos)

4:45 pm **Paper session: *Signal and Image Processing***

*** *U Castellani, M Cristani, P Marzola, V Murino, E Rossato, A Sbarbati* — *Page 25*
Cancer Area Characterization by Non-Parametric Clustering

*** *M Verduijn, N Peek, E de Jonge, B de Mol* — *Page 31*
A Procedure for Automated Filtering of ICU Monitoring Data using Basic Smoothing Techniques and Clinical Judgement

* *N Brümmer, J Baumeister, D Riewenherm, F Puppe, J Broscheit* — *Page 37*
Visual Development of Temporal Patterns for Medical Data Abstraction

5:45 pm **End of Day 1**

8:00 pm **Dinner**

IDAMAP 2006
Intelligent Data Analysis in bioMedicine And Pharmacology

Schedule

Saturday, August 26, 2006

9:00 am **Invited presentation**

Xiaohui Liu — *Page 41*
Intelligent Data Analysis for Biomedicine: What have we learned?

9:45 pm **Paper session: *Bioinformatics***

*** *T Curk, U Petrovic, G Shaulsky, B Zupan,* — *Page 45*
Rule-based Clustering for Gene Regulation Pattern Discovery

*** *G Santafé, J A Lozano, P Larrañaga* — *Page 51*
Population substructure determination by means of Bayesian model averaging for Clustering

* *F Ferrazzi, P Sebastiani, R Bellazzi, I S Kohane, M F Ramoni* — *Page 57*
Identification of Feedback Structures from Gene Expression Time Series

* *K Kulovesi, J Muhonen, I Lappalainen, P T Riikonen, M Vihinen, H Toivonen, T A Pasanen* — *Page 61*
Visualisation of Associations Between Nucleotides in SNP Neighbourhoods

11:00 am **Break** (including demos)

11:30 pm **Paper session: *Markov Models***

M A J van Gerven, F J Díez, B G Taal, P J F Lucas — *Page 65*
*** Prognosis of High-grade Carcinoid Tumor Patients using Dynamic Limited-Memory Influence Diagrams

*** *T Charitos, S Visscher, L C van der Gaag, P Lucas, K Schurink* — *Page 71*
A Dynamic Model for Therapy Selection in ICU Patients with VAP

L Peelen, N Peek, R J Bosman — *Page 77*
*** Describing Scenarios for Disease Episodes and Estimating Their Probability: a new Approach with an Application in Intensive Care

12:45 am **Lunch**

2:30 pm **Panel discussion: *An infrastructure for collaboration in time series analysis***

*** *J Hunter* — *Page 85*
TSNet – A Distributed Architecture for Time Series Analysis

3:30 pm **Paper session: *Information Retrieval, Data Mining***

*** *T B Røst, Øystein Nytrø, A Grimsmo* — *Page 93*
Investigating the Value of Diagnosis Codes in the Primary Care Patient Record

* *O Edsberg, S J Nordbø, Øystein Nytrø, A Grimsmo* — *Page 99*
Event Chart Explorer: A Prototype for Visualizing and Querying Collections of Patient Histories

* *A Shillabeer, J F Roddick, D de Vries* — *Page 101*
On the Arguments Against the Application of Data Mining to Medical Data Analysis

4:15 pm **Closing**

IDAMAP 2006
Intelligent Data Analysis in bioMedicine And Pharmacology

Schedule

During the breaks there will be demos of various intelligent data analysis tools and systems.

Demos (during breaks)

Timing of presentations:

Invited talks: 35 minutes + 10 minutes discussion
Long presentations (***): 20+5 minutes
Short presentation (*): 8+2 minutes

Invited presentation

Knowledge-Driven Querying of Time-Oriented Biomedical Data

Amar K. Das

Departments of Medicine and of Psychiatry and Behavioral Sciences
Stanford University School of Medicine

Querying and abstracting time-stamped data are frequently undertaken steps in biomedical data analysis and require extensive use of domain knowledge that is difficult to support at the database level. Prior knowledge-based methods for biomedical data analysis have not adequately addressed the temporal limitations of underlying database technologies. Thus, there is a need for principled methods that can resolve the disconnect between the representation of temporal data in biomedical databases and the specification of domain-relevant concepts used in data analysis. In this talk, I present my group's work on methods for knowledge-level querying of time-oriented data that permit knowledge generated from query results to be tied to the data and, if necessary, used for further inference. We use the Semantic Web ontology and rule languages, OWL and SWRL, respectively, to specify a temporal ontology that can integrate the domain knowledge with the database content. We have created a general bridge-based software architecture to process knowledge-driven queries efficiently using existing time-oriented databases. I demonstrate the applicability of our approach for the discovery of drug resistance patterns in the Stanford HIV Database.

Paper session: *Temporal Data Mining*

Mining Possibilistic Temporal Constraint Networks: A Case Study in Diagnostic Evolution at Intensive Care Units

Francisco Guil
Dept. Languages and Computer Science
University of Almeria
04120 - Almeria, Spain
francisco.guil@ual.es

Jose M. Juarez and **Roque Marin**
Dept. Information and Comm. Engineering
University of Murcia
30100 - Murcia, Spain
{jmjuarez, rmarin}@dif.um.es

Abstract

It is commonly accepted that the large number of temporal associations extracted in the temporal data mining step makes the knowledge discovery process practically unmanageable for human experts. This is the typical second-order data mining problem, where the vast amount of simple sequences or patterns needs to be summarized further. In this paper we propose a method for building possibilistic temporal constraint networks that better summarizes the huge set of mined timed-stamped sequences from a temporal data mining process. This method is based on the Theory of Evidence of Shafer as a mathematical tool for obtaining the fuzzy measures involved in the temporal network. This work also presents a practical example describing an application of this proposal in the Intensive Care Unit domain.

1 Introduction

Temporal data mining can be defined as the activity of looking for interesting correlations (or patterns) in large sets of temporal data accumulated for other purposes. It has the capability of mining activity, inferring associations of contextual and temporal proximity, that could also indicate a cause-effect association. This important kind of knowledge can be overlooked when the temporal component is ignored or treated as a simple numeric attribute [Roddick and Spiliopoulou, 2002]. In non-temporal data mining techniques, there are usually two different tasks: the description of the characteristics of the database (or analysis of the data), and the prediction of the evolution of the population. However, in temporal data mining this distinction is less appropriate, because the evolution of the population is already incorporated in the temporal properties of the analyzed data.

In [Guil *et al.*, 2004] we presented an algorithm, named $TSET$, based on the inter-transactional framework for mining frequent sequences from several kind of datasets, mainly transactional and relational datasets. The improvement of the proposed solution was the use of a unique structure to store all frequent sequences. The data structure used is the well-known set-enumeration tree, widely used in the data mining area, in which the temporal semantic is incorporated. The result is a set of frequent sequences describing partially the dataset. This set forms a potential base of temporal information that, after the experts analysis, can be very useful to obtain valuable knowledge. However, the overwhelming number of discovered frequent sequences may make such task absolutely impossible in practice. This problem can be viewed as a second-order data mining problem, which consists in the necessity of obtaining a more understandable and useful sort of knowledge from a huge volume of temporal associations resulting after the data mining process.

In this paper, we propose an extension of a previous work [Guil and Marín, 2006], which consists on the description of the building of a special model of temporal network formed by a set of uncertain relations amongst temporal points. The temporal model, proposed by HadjAli, Dubois and Prade in [HadjAli *et al.*, 2004], is based on the Possibility Theory as expressive tool for the representation and management of uncertainty in point-based temporal relations. The uncertainty is represented by a vector describing three possibility values, expressing the relative plausibility of the three basic relations between two temporal points, that is, "before", "at the same time" and "after". Thus, the authors define the basic operations (inversion, composition, combination and negation) that allow to infer new temporal information and to propagate uncertainty in a possibilistic way.

Once the sequences base is obtained (characterized by a frequency distribution), we propose a Shafer Theory-based technique which: firstly divides the sequence base into a set of nested subsets and then it normalizes the frequencies of each nested subset so they add to 1. Secondly, for each nested subset, it builds a temporal constraint network calculating, for each pair of temporal points or events, the possibility degrees of the three basic temporal relations. The result is an enumeration of temporal constraint networks that better summarizes the temporal information existing in the dataset. In other words, they permit the qualitative representation of uncertain temporal relations and they are based on formal sound theory for reasoning with uncertainty.

The remainder of the paper is organized as follows. Section 2 describes briefly the $TSET$ algorithm and gives a formal description of the problem of mining frequent sequences from datasets. Section 3 describes briefly the representation aspects of the possibilistic temporal model. In Section 4 we describe the approach for obtaining the uncertain vectors associated with the basic temporal relations

from the divided sequences base. Section 5 presents a practical experience at Intensive Care Unit (hereinafter ICU) that illustrates the proposed approach. Conclusions and future work are finally drawn in Section 6.

2 The $TSET$ algorithm

$TSET$ is an algorithm designed for mining frequent sequences (or frequent temporal pattern) from large relational datasets. It is based on the 1-dimensional inter-transactional framework [Lu *et al.*, 2000], and therefore, the aim is to find associations of events amongst different records (or transactions), and not only the associations of events within records. The main improvement of $TSET$ is that it uses a unique tree-based structure to store all frequent sequences. The data structure used is the well known set-enumeration tree, in which the temporal semantic is incorporated.

The algorithm follows the same basic principles as most apriori-based algorithms [Agrawal *et al.*, 1993]. Frequent sequence mining is an iterative process, and the focus is on a *level-wise* pattern generation. Firstly, all frequent 1-sequences (frequent events) are found, these are used to generate frequent 2-sequences, then 3-sequences are found using frequent 2-sequences, and so on. In other words, (k+1)-sequences are generated only after all k-sequences have been generated. On each cycle, the *downward closure* property is used to prune the search space. This property, also called anti-monotonicity property, indicates that if a sequence is infrequent, then all super-sequence must also be infrequent.

In the sequel, we will introduce the terminologies and the definitions necessary to establish the problem of mining frequent sequences from large datasets.

2.1 Concepts and terminologies

Definition 1 *A* dataset D *is an ordered sequence of records* $D[0]$, $D[1]$,...,$D[r-1]$ *where each* $D[i]$ *can have* c *columns or attributes,* $A[0], ..., A[c-1]$. *The 0-attribute will be the dimensional attribute, the temporal data associated with the record, expressed in temporal units. The rest of attributes can be quantitative or categorical.*

We assume that the domain of each attribute is a finite subset of non-negative integers, and we also assume that the structure of time is discrete and linear. Due to every event registered has its absolute date identified, we represent the time for events with an absolute dating system [Pani, 2001].

In order to simplify the calculations, we transform the original dataset subtracting the date of each record from the date of the first record, i.e. the time origin.

Definition 2 *An* event e *is a 3-tuple* $(A[i], v, t)$, *where* $0 < i < c$, $v \in dom\{A[i]\}$, *and* $t \in dom\{A[0]\}$, *that is,* $t \in \mathbb{N}$. *Events are "things that happen", and they usually represent the dynamic aspect of the world [Pani, 2001].*

In our case, an event is related to the fact that a value v *is assigned to a certain attribute* $A[i]$ *with the occurrence time* t. *The set of all distinct pairs (A[i],v) can be also called* event types. *We will use the notation* $e.a$, $e.v$, *and* $e.t$ *to set and get the attribute, value, and time variables related to the event* e, *and* $e.type$ *to get the event type associated with it.*

Definition 3 *Given two events* e_1 *and* e_2, *we define the* $\leq$ *relation as follows:*

1. $e_1 = e_2$ *iff* $(e_1.t = e_2.t) \wedge (e_1.a = e_2.a) \wedge (e_1.v = e_2.v)$
2. $e_1 < e_2$ *iff* $(e_1.t < e_2.t) \vee ((e_1.t = e_2.t) \wedge (e_1.a < e_2.a))$
3. $e_1 \leq e_2$ *iff* $(e_1 < e_2) \vee (e1 = e2)$

We assume that a lexicographic ordering exists among the pairs (attribute, value), the events types, in the dataset.

Definition 4 *A* sequence (or event sequence) *is an ordered set of events* $S = \{e_0, e_1, ..., e_{k-1}\}$, *where for all* $i < j$, $e_i < e_j$.

Obviously, $|S| = k$. *Note that different events with the same temporal unit can belong to the same sequence. Furthermore, the same events with different temporal unit associated can belong to the same sequence. Nevertheless, in any case will exist two or more pairs (attribute, value) associated to the same temporal unit. So, an attribute cannot take two different values in the same instant.*

Definition 5 *Let* U_{tmin} *be the minimal dimensional value associated to the sequence* S. *In other words,* $U_{tmin} = min\{e_i.t\}$, *for* $e_i \in S$. *If* $U_{tmin} = 0$, *we say that* S *is a* normalized sequence. *Note that any non-normalized sequence can be transformed into a normalized one through a normalization function.*

Let U_{tmax} *be the maximal dimensional value associated to the sequence* S. *This value indicates the maximum distance amongst the events belonging to the normalized sequence* S. *In other words,* $U_{tmax} = e_k.t$, *where* $|S| = k$. *From both, confidence and complexity points of view [Lu* et al., *2000], this value will be always less than or equal to a user-defined parameter called* $maxspan$, *denoted by* ω.

Definition 6 *The* support *(frequency) of a sequence is defined as:*

$$support(S) = \frac{f_r(S)}{|D|},$$

where $f_r(S)$ *denotes the number of occurrences of the sequence* S *in the dataset, and* $|D|$ *is the number of records in the dataset* D, *in other words,* r.

Definition 7 *A* frequent sequence *is a normalized sequence whose support is greater than or equal to a user-specified threshold called minimum support. We denote this user-defined parameter as* $minsup$, *or simply* σ.

Definition 8 *A sequence is a* frequent maximal sequence *if and only if it is frequent and no proper super-sequence (superset) of it is frequent.*

Given a dataset D and the user-defined parameters ω and σ, the goal of sequence mining is to determine in the dataset the set $\mathcal{S}_f^{D,\sigma,\omega}$, formed by all the frequent sequences whose support are greater than or equal to σ, that is,

$$\mathcal{S}_f^{D,\sigma,\omega} = \{S_i | support(S_i) \geq \sigma\}.$$

This set, formed by a large number of time-stamped sequences, is the goal of the temporal data mining algorithm

and the input of the method proposed in this paper for obtaining a temporal constraint network. Basically, the idea is to divide it into a set of nested subsets and, for each subset, obtain a temporal constraint model which summarize better the existing temporal information in the sequences.

3 Representation of Uncertain Temporal Relations

In literature can be found a large amount of work trying to handle uncertainty in temporal reasoning. However, very few work deal with time points as ontological primitives for expressing temporal elements. Basically, two temporal point-based approaches have been recently proposed for representing and managing uncertain relations between events, the probabilistic model done by Ryabov and Puuronen [Ryabov and Puuronen, 2001], and the possibilistic model proposed by HadjAli, Dubois, and Prade [HadjAli *et al.*, 2004]. In this paper, the authors argued the main differences between these two approaches. Mainly, there are two main differences. First, the possibilistic modeling can be purely qualitative, avoiding the necessity of quantifying uncertainty if information is poor. Second, their proposal is capable of modeling ignorance in a non-biased way. In our case, the selection of the possibilistic model is reinforced by the fact that we need a model which make the fusion of mined and expert knowledge easier [Dubois *et al.*, 1999].

The selected model is based on possibility theory [Dubois and Prade, 1988] for the representation and management of uncertainty in temporal relations between two point-based events. Uncertainty is represented as a vector involving three *possibility values* expressing the relative plausibility of the three basic relations (" $<$ "," $=$ ", and " $>$ ") that can hold between these points. Also, they describe the inference rules (that form the basis of the reasoning method) defining a set of operations: inversion, composition, combination, and negation, the operations that govern the uncertainty propagation in the inference process. The authors show that the whole reasoning process can actually be handled in possibilistic logic.

Three basic relations can hold between two temporal points, "before ($<$)", "at the same time ($=$)", and "after ($>$)". An uncertain relation between temporal points is expressed as any possible disjunction of basic relations:

$\leq$	$\Longleftrightarrow$	$< or =$
$\geq$	$\Longleftrightarrow$	$> or =$
$\neq$	$\Longleftrightarrow$	$< or >$
$?$	$\Longleftrightarrow$	$<, =, or >$

The last case represents *total ignorance*, that is, any of the three basic relations is possible. The representation is extended using the Possibility Theory for modeling the plausibility degree of each basic relation. Given two temporal points, a and b, an uncertain relation r_{ab} between them is represented by a *normalized vector* $\Pi_{ab} = (\Pi^{<}_{ab}, \Pi^{=}_{ab}, \Pi^{>}_{ab})$, such that $max(\Pi^{<}_{ab}, \Pi^{=}_{ab}, \Pi^{>}_{ab}) = 1$, where $\Pi^{<}_{ab}$ (respectively, $\Pi^{=}_{ab}$, $\Pi^{>}_{ab}$) is the possibility of $a < b$ (respectively $a = b$, $a > b$).

From the uncertain vector $(\Pi^{<}_{ab}, \Pi^{=}_{ab}, \Pi^{>}_{ab})$, and using the duality between possibility and necessity, namely

$$N(A) = 1 - \Pi(A^c), \quad \text{where } A^c \text{ is the complement of A}$$

we can derive the possibility and necessity degree of each basic relation and their disjunctions.

As,

$$\Pi^{\leq}_{ab} = max(\Pi^{<}_{ab}, \Pi^{=}_{ab})$$
$$\Pi^{\geq}_{ab} = max(\Pi^{=}_{ab}, \Pi^{>}_{ab})$$
$$\Pi^{\neq}_{ab} = max(\Pi^{<}_{ab}, \Pi^{>}_{ab}),$$

we can obtain the necessity degrees of the basic relations,

$$N^{<}_{ab} = N(a < b) = 1 - \Pi^{\geq}_{ab}$$
$$N^{=}_{ab} = N(a = b) = 1 - \Pi^{\neq}_{ab}$$
$$N^{>}_{ab} = N(a > b) = 1 - \Pi^{\leq}_{ab}.$$

In a similar way, we can also obtain

$$N^{\geq}_{ab} = N(a \geq b) = 1 - \Pi^{<}_{ab}$$
$$N^{\neq}_{ab} = N(a \neq b) = 1 - \Pi^{=}_{ab}$$
$$N^{\leq}_{ab} = N(a \leq b) = 1 - \Pi^{>}_{ab}.$$

Moreover, the authors defined the rules that enable us to infer new temporal information and to propagate uncertainty in a possibilistic way. The reasoning tool relies on four operations expressing:

$inversion$	$\Longleftrightarrow$	$\widetilde{r}_{ab} = r_{ba}$
$composition$	$\Longleftrightarrow$	$r_{ac} = r_{ab} \otimes r_{bc}$
$combination$	$\Longleftrightarrow$	$r_{ab} = r_{1_{ab}} \oplus r_{2_{ab}}$
$negation$	$\Longleftrightarrow$	$\neg$

These rules complete the definition of a model for representing and reasoning with uncertain temporal relations that uses the Possibility Theory as an expressive tool for dealing with uncertainty in temporal reasoning.

4 Extracting Uncertain Temporal Relations

In this section, we propose a technique for extract the uncertain temporal relation between each pair of event types from the sequences base. The uncertain temporal relation is represented by an uncertain vector formed by three possibility values, expressing the plausibility degree for each basic temporal relation. We propose the use of Shafer Theory of Evidence [Shafer, 1976] to obtain the plausibility degrees from the frequencies values associated with the set of sequences. The result will be a set of temporal constraint networks, which belong to a a suitable model for representing and reasoning with temporal information where uncertainty is presented.

4.1 Shafer's Theory of Evidence

The Shafer Theory of Evidence, also known as Dempster-Shafer Theory, is a theory of uncertainty developed specially for modelling complex systems. It is based on a special fuzzy measure called *belief measure*. Beliefs can be assigned to propositions to express the uncertainty associated to them being discerned. Given a finite universal set $\mathcal{U}$, the *frame of discernment*, the beliefs are usually computed based on a density function $m : 2^{\mathcal{U}} \rightarrow [0, 1]$ called *basic probability assignment* (bpa):

$$m(\emptyset) = 0, \text{ and } \sum_{A \subseteq \mathcal{U}} m(A) = 1.$$

$m(A)$ represents the belief exactly committed to the set A. If $m(A) > 0$, then A is called a *focal element*. The set of focal elements constitute a core:

$$\mathcal{F} = \{A \subseteq \mathcal{U} : m(A) > 0\}$$

The core and its associated bpa define a *body of evidence*, from where a belief function $Bel : 2^{\mathcal{U}} \rightarrow [0, 1]$ is defined:

$$Bel(A) = \sum_{B|B \subseteq A} m(B)$$

For any given measure Bel, a *dual measure*, $Pl : 2^{\mathcal{U}} \rightarrow [0, 1]$ can be defined:

$$Pl(A) = 1 - Bel(\overline{A}).$$

So, this measure called *plausibility measure*, can be also defined:

$$Pl(A) = \sum_{B|B \cap A \neq \emptyset} m(B).$$

It can be verified [Shafer, 1976] that the functions Bel and Pl are, respectively, a possibility (or necessity) measure if and only if the focal elements form a nested or consonant set, that is, if it can be ordered in such a way that each is contained within the next. In that case, the associated *belief* and *plausibility* measures posses the following properties: For all $A, B \in 2^{\mathcal{U}}$,

$$Bel(A \cap B) = N(A \cap B) = min[Bel(A), Bel(B)]$$
$$Pl(A \cup B) = \Pi(A \cup B) = max[Pl(A), Pl(B)]$$

4.2 Calculating the possibility measures of temporal relations

In our proposal, the sequences base is formed by a set of linked nested set, each one corresponding to a frequent maximal sequence and its subsequences. From an algorithm point of view, each nested set corresponds with a branch of the tree. So the proposed method build the temporal constraint networks in a linear time, just with a depth-first traversal of the tree.

Following the notation of Shafer's Theory, our core is each set of nested sequences $\mathcal{NS} \subseteq \mathcal{BS}^{D,\sigma,\omega}$ which is formed by a set of focal elements or sequences. We normalize the frequencies of each nested subset so they add to 1.

Let Ω be the set of event types presented in the dataset, that is,

$$\Omega = \{(A[i], v)|v \in dom(A[i])\}.$$

Taking into account the *maxspan* constraint, the set of events is defined as an extension of the Ω set in this way:

$$\Omega^{\omega} = \{(A[i], v, t)|v \in dom(A[i]) \wedge 0 \leq t \leq w)\}$$

This set is our frame of discernment, that is, $\Omega^{\omega} = \mathcal{U}$. So, the set of focal elements, the nested sequences base, is defined:

$$\mathcal{NS} = \{S_i \subseteq \Omega^{\omega}|m(S_i) > 0\},$$

where m is the bpa function derived from the frequencies of the sequences, such that $m : 2^{\Omega^{\omega}} \rightarrow [0, 1]$,

$$m(\emptyset) = 0, \sum_i m(S_i) = 1$$

We will denote a temporal relation between two events e_1, e_2 as $e_1 \Theta e_2$. Since we are only interested in the basic temporal relations,

$$\Theta \in \{<, =, >\}.$$

For each pair of event types presented in the nested set, we need to obtain the possibility degree of each basic temporal relation between them. In order to compute the possibility of a temporal relation, it is necessary to consider all focal elements, that is, all sequences which make the temporal relation possible. However, from complexity point of view, we will obtain the possibility degrees from the necessity ones, calculated over the complement of the basic temporal relation, that is,

$$\Theta^c \in \{>=, <>, <=\}.$$

Proposition 1 *Let suppose the qualitative temporal relation $e_1 \Theta^c e_2$. This relation induces a parameterized set:*

$$\mathcal{X}_{e_1 \Theta^c e_2} = \{(e_i e_j)\},$$

where $e_i, e_j \in \Omega^{\omega}, e_i.type = e_1$, $e_j.type = e_2$, and $e_i.t \Theta^c e_j.t$.

Proposition 2 *In order to obtain the set of sequences involved in the temporal relation, we introduce the assessment operator Γ, defined as:*

$$\Gamma(\mathcal{X}_{e_1 \Theta^c e_2}) = \{S_i | S_i \subseteq \mathcal{X}_{e_1 \Theta^c e_2}\},$$

where $S_i \in \mathcal{NS}$.

Proposition 3 *The possibility degree of the temporal relation $e_1 \Theta e_2$ is defined as:*

$$\Pi(e_1 \Theta e_2) = 1 - N(e_1 \Theta^c e_2) = 1 - \sum_{S_i \in \Gamma(\mathcal{X}_{e_1 \Theta e_2})} m(S_i)$$

5 A practical experience at Intensive Care Unit

The Intensive Care Unit (ICU) is a medical service to provide critical attention of medically recoverable patients. One of the fundamental characteristics of this domain is that patients require a permanent availability of monitoring equipment and specialist care. Thus, clinicians work in shifts in order to provide a 24 hours service. In this sense, the temporal evolution of patients is permanently recorded. Physicians at ICU are daily required to provide reports, describing the different diagnosis hypotheses that they assume and the posterior actions (tests, treatments, or requiring new laboratory analysis). In our particular case, the ICU service has a Health Information System (HIS) that stores this information and generates the reports.

Due to the amount of information (different medical areas implied), and the importance of the temporal dimension (implicitly and explicitly analysed in patients' evolution), we consider that the ICU is a suitable domain to apply our second-order temporal data mining proposal.

5.1 Practical Application

In ICU domains, as well as the final diagnosis (like other hospital services), there are *evolutive diagnoses* that state the diagnostic hypotheses. These hypotheses are daily

made by physicians during patient's stay at the ICU service. Furthermore, they can be considered high-level medical information since it is obtained from physician's knowledge and medical observations (like EKGs, tests, or nursing care data).

Despite the importance of other clinical information within the health record, such as treatments or demographic data, we consider in our experiment that the evolution of these diagnosis are a good representation of patient problems and the discovery of temporal pattern diagnosis could be useful in many AI systems for temporal diagnosis or prognosis.

In our experiment, each patient is represented in the database by a temporal sequence of diagnoses (temporal points) and the data mining process results are frequent temporal patterns (or frequent sequences) of diagnosis evolution. In the analysis of this data, different parameters have been empirically stated ($maxspan$ = 24 , and $support$ value = 3, 5, 9) depending of the dataset of 144 patients.

Supp	Patient Patt	Tot Patt
3	N= 936 Max=5	N=379374 Max=12
5	N=122 Max=3	N=115810 Max=11
9	N= 49 Max=1	N=20837 Max=9

Table 1: Practical experiments considering independent patients and complete data. Supp = data mining parameter of minimum support. N = number of sequences obtained. Max = maximum size of the sequences.

In Table 1 is shown a summary of some of the results obtained from the proposed data mining process. In order to count the frequency of possible patterns, two alternative strategies have been adopted. Firstly, if medical data is interpreted, the patterns discovery could be more relevant, considering only the repetitions of the occurrences of diagnosis hypotheses on different patients (Patient Patt column in Table 1). Secondly, considering all possible occurrences without any kind of semantics (see Tot Patt column in Table 1). At present, there is not fully medical evaluation of the patterns obtained yet. However, it must be considered that the current state of the practical part of this research is in an initial step.

5.2 Evolutive Diagnosis Pattern Example

In order to explain the results obtained, this section describes a particular example of the patterns obtained from the complete temporal data mining process applied on temporal diagnosis evolution.

The following maximal sequence (and their subsequences) has been obtained from the ICU database:

Id	Sequence	Frequence
s_4	$\{(d_6,0),(d_7,0),(d_{169},0),(d'_{169},3)\}$	3
s_3	$\{(d_6,0),(d_7,0),(d_{169},0)\}$	4
s_2	$\{(d_6,0),(d_7,0)\}$	6
s_1	$\{(d_6,0)\}$	10

Table 2: Sequences and frequency. (d_i,t) diagnosis i at day t

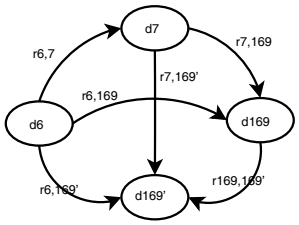


Figure 1: Pattern described by a Possibilistic Temporal Constraint Network

The maximal sequence (s_4 in Table2) describes patients to whom physicians diagnosed at income day:d_6 (Acute Miocardial Infarction of the Pared Inferior -ICD 10 I211), d_7 (haemorrhage complications), and d_{169} (Acute Miocardial Infarction -ICD 10 I252). But also d_{169} again the third day of stay at the ICU.

Thus, a basic assignment (m) can be defined considering Evidence Theory of Shafer and normalizing the sequence's frequency.

$$m(s_1) = 10/23 = 0.434.. \quad (1)$$
$$m(s_2) = 6/23 = 0.260.. \quad (2)$$
$$m(s_3) = 4/23 = 0.173.. \quad (3)$$
$$m(s_4) = 3/23 = 0.130.. \quad (4)$$

Note that sequences describe a nested set ($s_1 \subseteq s_2 \subseteq s_3 \subseteq s_4$) and therefore Shafer Theory can be applied for obtaining the possibilistic values of relations between singleton sets as follows:

$$\Pi_{d_i<d_j} = 1 - N_{d_i \geq d_j} = 1 - Bel_{d_i \geq d_j} \quad (5)$$
$$\Pi_{d_i=d_j} = 1 - N_{d_i <> d_j} = 1 - Bel_{d_i <> d_j} \quad (6)$$
$$\Pi_{d_i>d_j} = 1 - N_{d_i \leq d_j} = 1 - Bel_{d_i \leq d_j} \quad (7)$$

For example, in the particular case of d_6 and d_7:

$$\Pi_{d_6<d_7} = 1 - \sum_{d_6 \geq d_7 \subseteq B} m(B) = 1 - 13/23 = 10/23 \quad (8)$$
$$\Pi_{d_6=d_7} = 1 - \sum_{d_6 <> d_7 \subseteq B} m(B) = 1 - 0 = 1 \quad (9)$$
$$\Pi_{d_6>d_7} = 1 - \sum_{d_6 \leq d_7 \subseteq B} m(B) = 1 - 13/23 = 10/23 \quad (10)$$

These formulae state the possibilistic values of the three temporal relations between d_6 and d_7, describing one of the temporal constraints of the pattern $((\Pi_{d_6<d_7}, \Pi_{d_6=d_7}, \Pi_{d_6>d_7}))$. The relations between $d_6 - d_{169}$, $d_7 - d_{169}$,and $d_{169} - d'_{169}$ can be obtained in the same way (see Figure 1).

Note that the inverse of these relations are easily obtained by the use of the inverse operator defined by Hadjali, Dubois, and Prade temporal model. Thus, the final set of possibilistic temporal constraints is shown in table 3:

Combination of Patterns

This model partially solves the second order problem of data mining, providing a simple representation of the pa-

	d6	d7	d169	d'169
d6	(1,1,1)	(.44,1,.44)	(.7,1,.7)	(1,.87,.87)
d7	(.44,1,.44)	(1,1,1)	(.7,1,.7)	(1,.87,.87)
d169	(.7,1,.7)	(.7,1,.7)	(1,1,1)	(1,.87,.87)
d'169	(.87,.87,1)	(.87,.87,1)	(.87,.87,1)	(1,1,1)

Table 3: All Possibilistic Temporal Constraints of the Network from ICU data

tterns obtained. Thus, one of the advantages of this representation is the capability of **combination** between different patterns. This is useful when some kind of reasoning is required to infer new potential patterns.

Let consider again two patterns obtained given the following maximal sequences from the ICU database:

Id	Sequence	Frequence
s_4	$\{(d_6,0),(d_7,0),(d_{169},0),(d'_{169},3)\}$	3
s_5	$\{(d_6,0),(d_{95},0),(d_{169},0),(d'_{95},2)\}$	4

Table 4: Maximal Sequences.

The possibilistic temporal constraint networks are obtained as shown in previous section. In order to do this combination, we suggest to use the *Minimum Rule*. Then, for each relation present in both patterns (e.g. $r_{d_6-d_{169}}$ in our case), the new relation of the combined pattern is the minimum of both: $min(r_{d_6d_{169}}, r'_{d_6d_{169}}) = ((min(\Pi_{d_6<d_{169}}, \Pi'_{d_6<d_{169}})), (min(\Pi_{d_6=d_{169}}, \Pi'_{d_6=d_{169}})), (min(\Pi_{d_6>d_{169}}, \Pi'_{d_6>d_{169}})))$ In case that one of the relations is not present in both (e.g. $r_{d_6d_{95}}$), the same calculus is done with the trivial relation $(1,1,1)$.

6 Conclusions and future work

In this paper, we propose an initial approach for building qualitative temporal constraint networks from a set of mined frequent sequences with the aim of obtaining a more understandable, useful, and manageable sort of knowledge. The selected temporal model is the proposed by HadjAli, Dubois, and Prade, which uses the Possibility Theory as an expressive tool for representing and reasoning with uncertain temporal relations between point-based events. We propose a Shafer's Theory-based technique to obtain these possibility degrees involved in the network from the frequencies of the sequences.

In order to demonstrate the viability of this proposal we have applied it to the temporal evolution of diagnosis hypotheses at a ICU service. Despite that the clinical validation is not yet performed, the presented results points out the simplicity of representation and the advantage for expert's comprehension.

In future work, we intend to analyze in depth the networks obtained from the set of mined frequent sequences. We also propose to extend the model of temporal network in order to represent not only qualitative but also quantitative temporal relations, taking advantage of the temporal information presented in the time-stamped sequences extracted by $TSET$.

Acknowledgments

This work is supported in part by MEC TIC2003-09400-C04 and the FPU national plan (grant ref. AP2003-4476).

References

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proc. of the ACM SIGMOD Int. Conf. on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[Dubois and Prade, 1988] D. Dubois and H. Prade. *Possibility Theory*. Plenum Press, 1988.

[Dubois *et al.*, 1999] D. Dubois, H. Prade, and G. Yager. Merging fuzzy information. In *Fuzzy Sets in Approximate Reasoning and Information Systems*, pages 335–401. Kluwer Academic Publishers, 1999.

[Guil and Marín, 2006] F. Guil and R. Marín. Extracting uncertain temporal relations from mined frequent sequences. In *Proc. of the 13th Int. Symposium on Temporal Representation and Reasoning (TIME 2006), accepted*, 2006.

[Guil *et al.*, 2004] F. Guil, A. Bosch, and R. Marín. TSET: An algorithm for mining frequent temporal patterns. In *Proc. of the First Int. Workshop on Knowledge Discovery in Data Streams, in conjunction with ECML/PKDD 2004*, pages 65–74, 2004.

[HadjAli *et al.*, 2004] A. HadjAli, D. Dubois, and H. Prade. A possibility theory-based approach for handling of uncertain relations between temporal points. In *11th International Symposium on Temporal Representation and Reasoning (TIME 2004)*, pages 36–43. IEEE Computer Society, 2004.

[Lu *et al.*, 2000] H. Lu, L. Feng, and J. Han. Beyond intra-transaction association analysis: Mining multi-dimensional inter-transaction association rules. *ACM Transactions on Information Systems (TOIS)*, 18(4):423–454, 2000.

[Pani, 2001] A. K. Pani. Temporal representation and reasoning in artificial intelligence: A review. *Mathematical and Computer Modelling*, 34:55–80, 2001.

[Roddick and Spiliopoulou, 2002] J. F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.

[Ryabov and Puuronen, 2001] V. Ryabov and S. Puuronen. Probabilistic reasoning about uncertain relations between temporal points. In *8th International Symposium on Temporal Representation and Reasoning (TIME 2001)*, pages 1530–1511. IEEE Computer Society, 2001.

[Shafer, 1976] G. Shafer. *A Mathematical Theory of Evidence*. Princenton University Press, Princenton, NJ, 1976.

Describing and modeling time series based on qualitative temporal abstraction

Lucia Sacchi[1], Marion Verduijn[2,5], Niels Peek[2], Evert de Jonge[3], Bas de Mol[4,5], Riccardo Bellazzi[1]
[1]Laboratory for Medical Informatics, University of Pavia, Pavia, Italy, lucia.sacchi@unipv.it
[2]Dept. of Medical Informatics, Academic Medical Center (AMC), Amsterdam, The Netherlands
[3] Dept. of Intensive Care Medicine, AMC, Amsterdam, The Netherlands
[4] Dept. of Cardio-thoracic Surgery, AMC, Amsterdam, The Netherlands
[5] Dept. of Biomedical Engineering, University of Technology, Eindhoven, The Netherlands

Abstract

In this paper we address the problem of predicting the risk of prolonged mechanical ventilation in ICU patients on the basis of the temporal behavior of ten monitoring variables. The time course of such variables have been synthesized through four different Temporal Abstraction methods, including State, Trends and their combinations. A comparison of the performances of the different abstraction methods has been run on 644 cardiac surgery patients by using two different classifiers, the Naïve Bayes and the decision tree. Results show that the state and the combined methods are the best ones, in particular with the Naïve Bayes classifier. The Temporal Abstractions approach seems a convenient method to summarize ICU data for classification purposes.

1 Introduction

The analysis of time series collected by monitoring clinical variables is a topic of great interest in biomedical research. Since a huge amount of temporal data is generally collected in a multivariate context, clinicians and analysts usually feel the need to shift from raw data to a new, more synthetic, meaningful and computationally manageable dataset. To this aim, temporal data abstraction turns out to be one of the most interesting steps in the process of intelligent analysis of temporal biomedical data. In this context, the introduction and application of the formalism of knowledge-based Temporal Abstraction (TA) [Shahar, 1997] turned out to be an interesting issue to deal with several problems [Bellazzi *et al.*, 2005; Haimowitz and Kohane, 1996; Miksch *et al.*, 1996; Salatian and Hunter, 1999; Shahar and Musen, 1996; Verduijn *et al.*, 2005]. In this paper we address the challenging problem of predicting the risk of prolonged mechanical ventilation (PMV) for patients admitted at ICU after cardiac surgery by exploiting the formalism of TAs. In particular, we try to answer to two main questions: i) how a qualitative description of temporal data can help in the prediction of the outcome and ii) which is the best strategy to summarize the large number of abstractions that may be extracted from monitoring time series. In particular, focusing on TAs, we propose a comparison between several kinds of representations used as features for the prognostic problem and evaluate the results obtained by running different classification algorithms.

2 Material and Methods

2.1 Data

In this work we consider data coming from 664 patients who underwent cardiac surgery at the Academic Medical Centre in Amsterdam in the period from April 2002 to May 2004. As an ordinary post-surgical procedure, these patients are sent to the ICU, where they receive mechanical ventilation (MV). In normal postoperative courses, patients can be released from MV within 24 hours after ICU admission; in case of complications they instead undergo prolonged mechanical ventilation (PMV), i.e., MV for more than 24 hours.

During the ICU stay several variables are monitored over time; according to the frequency with which these variables are measured, they can be divided into two groups:

- Variables measured every minute (high frequency variables): mean arterial blood pressure (ABPm), central venous pressure (CVP), heart rate (HR), temperature (TMP), fraction inspired oxygen (FiO_2) and respiration pressure (RP). The latter two variables are parameters of the ventilator; they are set and regularly adjusted by the clinician at the lowest possible value and reflect the lung functioning of the patient;
- Variables measured several times a day (low frequency variables): base excess (BE), creatinine kinase MB (CKMB), glucose value (GLC), and cardiac output (CO).

The outcome prolonged mechanical ventilation (PMV) is defined relying on the duration of mechanical ventilation as 1 if the duration is greater than 24h and 0 otherwise. Moreover, since we aim at predicting the outcome within the first 12 hours after the admission at ICU, we consider temporal data in the interval 0-12 hours, leaving out the measurements in the interval 12-24 hours after admission.

Before entering the abstraction step, data needed an initial preprocessing phase, made up of two steps: first, all the clinically unreliable values were removed from all the variables, relying on thresholds defined by a clinical expert; second, high frequency variables were smoothed through a moving average technique in order to reduce the effects of additional noise artifacts in the time series.

For the qualitative representation of the time series we chose to resort to the formalism of knowledge-based TA.

2.2 Knowledge-based Temporal Abstractions

Temporal Abstractions represent a convenient AI technique to extract compact and meaningful descriptions from temporal data; the most interesting feature of such representation is the shift from a time-point (quantitative) to an interval-based qualitative representation of the time series, aimed at extracting specific patterns which are verified in the data. Within TAs, we can distinguish between *basic* and *complex* abstractions. Basic TAs are used to extract simple patterns, and can be specified into Trend TAs, to capture increasing, decreasing or stationary courses in a numerical time series, and State TAs, to detect qualitative patterns corresponding to low, high or normal values in a numerical or symbolic time series. Complex TAs, on the other hand, correspond to intervals in which specific temporal relationships between basic or other complex TAs hold. Such temporal relationships are usually identified with the ones defined in Allen algebra [Allen, 1984].

2.3 Comparison between different TA representations

As already mentioned in the introduction, in this paper we address the problem of the representation of temporal variables recorded during an ICU stay through TAs, aiming at evaluating their capability in predicting the outcome and at establishing a comparison of different descriptions of the variables. In more detail, by exploiting TAs in several ways, we propose the four representations introduced in the following sections.

Representation through State TAs

Through this representation we aim at assigning an abstraction reflecting information on the level of each variable (i.e., low, normal, high) over intervals specified over the time series. In particular, for high frequency variables we aim at defining a label over the four 3-hours intervals: 0-3, 3-6, 6-9, 9-12 hours, while for the low frequency variables the labels are detected over the two 6-hours intervals 0-6 and 6-12 hours after admission at the ICU. To extract such labels, each numerical value of the time series is first replaced by a qualitative label of the kind 'low', 'normal', or 'high', where the thresholds for defining the levels are detected through an automatic 10-fold cross validation procedure. As a second step, the proportion of labels of different type found over each interval is evaluated, in order to establish a label valid over all the period. If a label is found to be significantly overrepresented over a period (p-value of a chi-squared Pearson's statistic <0.05), that label is then assigned to the corresponding interval. If no majority label can be detected, the interval is labeled with a 'varying' TA. According to this strategy, a total number of 32 features for the classification problem is obtained; we have in fact one label for each of the 4 three-hour periods for the 6 high frequency variables and one label for each of the 2 six-hour periods for the 4 low frequency variables.

Representation through trend TAs

Through this representation we aim at assigning an abstraction reflecting information on the temporal trend of each variable (i.e. increasing, decreasing, stationary) over specific intervals. For what concerns the high frequency variables ABPm, CVP, HR and TMP, the same 3-hours intervals identified for the state abstractions are considered. As a first step, trend detection is performed over each of the considered periods, relying on a piecewise linear segmentation of the time series carried out through a sliding window algorithm [Keogh *et al*, 2003]. A trend label reflecting the information on the slope is used to label each segment of the approximating curve. The procedure for trend detection develops then in a similar way as for state abstractions: if more than one type of trend is detected over the three-hour period, a chi-squared Pearson's statistic is computed to determine if one trend label can be assigned as a global label to that specific period (p-value<0.05). If that is not the case, the global label 'varying' is assigned to the trend pattern.

Because of the long steady periods and the little number of measurements respectively, the variables FiO_2 and RP and the four low frequency variables were all considered over the whole twelve-hour period; according to this procedure, one global trend label is assigned to these time series relying on the slope of the regression line obtained over the whole period.

According to this strategy, a total number of 22 features for the classification problem is obtained; we have in fact one label for each of the 4 three-hour periods for ABPm, CVP, HR and TMP plus one label for FiO_2, RP and the 4 low frequency variables.

Combining State and Trend Abstractions

With this representation, features are defined by merging the information coming from both the trend and the state descriptions extracted as described in the previous paragraphs. In particular, state and trend labels are combined over each period defined for each variable, replicating the trend label where necessary (e.g., the state labels of each three-hour period for the variables FiO_2 and RP are combined with the trend label of the twelve hour period of these variables, which is 'spread' over every subperiod). As in the case of the State TA representation, the total number of features for the classification problem is of 32.

Clustering State and trend TAs

To reduce the high number of features (22 for the trend representation and 32 for the state description) obtained by representing time series through trend and state TAs we performed a clustering of the available labels and used the information on both the clustering trends and states as features for the classification problem. The general procedure to cluster the qualitative variables develops in a way which is similar both for trend and state TAs. In particular, when the variable is evaluated on different periods, the labels are first pasted together to form a pattern of trend or state changes. Time series that present similar labels are then clustered together according to an heuristic criterion. The following tables show the clusters that have been defined for each variable.

STATE Tas	
Variables	Clusters
ABPm, CVP, HR, and TMP	'high at least for six consecutive hours' 'low at least for six consecutive hours' 'normal for the last nine hours or for the whole twelve-hour period', 'normal for the last six hours' 'normal at least six consecutive hours not at the end' 'varying'
FiO2 and RP	'high at least for the six last hours' 'normal all twelve hours or low at least for six consecutive hours' 'normal at least for the six last hours 'varying'
BE, CI, CKMB, and GLC	'high at least for the six last hours' 'low at least for the six last hours' 'normal all twelve hours' 'normal at least for the six last hours' 'varying'

Table 1. Labels for the clustered State TAs

TREND Tas	
Variables	Clusters
ABPm, CVP, HR, and TMP	'decreasing at least for six consecutive hours', 'increasing at least for six consecutive hours', and 'varying'

Table 2. Labels for the clustered Trend TAs

As it is clear from the previous tables, the clusters are not mutually exclusive: for example, a variable which is normal in the last nine hours is also normal in the last six hours. To overcome this ambiguity, the variables are assigned to the most specific group (e.g. normal in the last nine hours).

2.4 Prognostic modeling

Once the TA-based features for each representation have been derived, both Naïve Bayes and classification trees were tested on the task of predicting the risk of PMV. The performances of the two classifiers were evaluated through 3-fold cross validation. The analysis were performed in Orange, a data mining environment which is based on visual programming [Demsar et al., 2004].

3 Results

In this Section we will present the results obtained by running Naïve Bayes and classification trees on the problem of predicting the risk of PMV from ICU monitoring temporal variables. For each of the TA representations presented in Section 2 a table is reported. As a reference, also the results for the majority classifier are shown. Classification accuracy (CA), sensitivity, specificity and area under ROC curve (AUC) are shown for a complete evaluation of the performances of the classifiers.

Classification Algorithm	CA	Sensitivity	Specificity	AUC
Majority	0.7048	1	-	0.5000

Table 3. Results for the default classifier, which classifies all the examples according to the majority class in the training set.

STATE TAs				
Classification Algorithm	CA	Sensitivity	Specificity	AUC
NB	0.7803	0.8589	0.5939	0.8163
CT	0.7244	0.8739	0.3703	0.6777

Table 4. Results for the State TA representation. (NB = Naïve Bayes, CT = Classification Tree)

TREND TAs				
Classification Algorithm	CA	Sensitivity	Specificity	AUC
NB	0.6958	0.6378	0.5013	0.6502
CT	0.6494	0.6579	0.5846	0.5785

Table 5. Results for the Trend TA representation. (NB = Naïve Bayes, CT = Classification Tree)

MERGED TAs				
Classification Algorithm	CA	Sensitivity	Specificity	AUC
NB	0.7623	0.8589	0.5324	0.8023
CT	0.7231	0.8547	0.4104	0.6603

Table 6. Results for the Merged State and Trend TA representation. (NB = Naïve Bayes, CT = Classification Tree)

CLUSTERED TAs				
Classification Algorithm	CA	Sensitivity	Specificity	AUC
NB	0.7517	0.8632	0.4680	0.7737
CT	0.7073	0.8377	0.3972	0.6201

Table 7. Results for the Clustered State and Trend TA representation. (NB = Naïve Bayes, CT = Classification Tree)

By comparing the results obtained using different kinds of TA representations, we can point out some interesting observations: first of all, the trend representation as it is seems to be not informative for the prediction of the outcome. The reason for such a behaviour may be that considering only the information about the trend (i.e. without any hint on the level of the variable) might not be enough to describe causes that may lead to a PMV. The observation that, on the other hand, the state representation results in better performances with both the classifiers supports the hypothesis just introduced.

The merging of state and trend TAs into a single label, that can be seen as a complex TA, performs in a way that is comparable to the state representation. From a clinical point of view this description might anyway be more informative that the mere use of state TAs, since it allows a more complete description of the variables, including both level and trend information.

A similar comment can be made also for what concerns the representation through clustered variables; with such a description we have in fact the possibility of synthesizing both the trend and the state features with an improvement in interpretability and clarity of the results for the clinicians.

4 Discussion and conclusions

In this paper we have analyzed how a qualitative representation of time series through knowledge-based TAs can be exploited for the task of predicting the risk of prolonged

mechanical ventilation in patients recovered at the ICU after cardiac surgery.

We have introduced four descriptions of temporal data derived by exploiting temporal abstractions in several ways. One of the main advantages of introducing such a representation of time series is of course the fact that it results in a more intuitive and clear interpretation of the results by the clinicians. The descriptions of the variables obtained in terms of TAs are in fact self-explanatory and don't need to be interpreted by an algorithm-expert.

Another issue that we have explored in this paper is the effect of combining different abstractions (representation through merged and clustered TAs) in terms of predictive capability and to reduce the number of features of the problem.

From the results it turns out that the best performances are obtained when introducing the information on the level of the variables into the problem features. Trend information by itself results in fact in poor performances of both the considered classifiers. The two methods for coupling both trend and state descriptions result rather satisfactory in terms of performances and in terms of synthesis of the information in a lower number of features in terms of interpretability of the results.

Further work can be made on both on the improving of the description through trend TAs, in particular by considering also the information about the rate (e.g. 'slightly increasing' or 'fast decreasing') and also on the improving on sensitivity which results rather low in all the cases.

References

[Allen, 1984] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123-154, 1984.

[Bellazzi et al., 2005] Riccardo Bellazzi, Cristiana Larizza, Paolo Magni, and Roberto Bellazzi. 2005. Temporal data mining for the quality assessment of hemodialysis services. *Artificial Intelligence in Medicine*, 34(1):25-39, 2005.

[Demsar et al., 2004] Janez Demsar, Blaz Zupan, Gregor Leban. Orange: From Experimental Machine Learning to Interactive Data Mining. White Paper (www.ailab.si/orange). Faculty of Computer and Information Science. University of Ljubljana, 2004.

[Haimowitz and Kohane, 1996] Ira J. Haimowitz, Isaac S. Kohane. Managing temporal worlds for medical trend diagnosis. *Artificial Intelligence in Medicine*, 8: 299-321, 1996.

[Keogh et al., 2003] Eamonn Keogh, Selina Chu, David Hart, Michael Pazzani. Segmenting time series: A survey and novel approach. In: M. Last, A. Kandel, H. Bunke (Eds.), Data Mining in Time Series Databases, World Scientfic Publishing Company, pp. 1-22, 2003.

[Miksch et al., 1996] Silvia Miksch, Werner Horn, Christian Popow, Franz Paky. Utilizing temporal data abstraction for data validation and therapy planning for artificially ventilated newborn infants *Artificial Intelligence in Medicine*, 8: 543-576, 1996.

[Salatian and Hunter, 1999] Apkar Salatian, Jim Hunter. Deriving trends in historical and real time continuously sampled medical data. *Journal of Intelligent Information Systems*, 13: 47-71, 1999.

[Shahar and Musen, 1996] Yuval Shahar and Mark A. Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine* 8(3):267-98, 1996.

[Shahar, 1997] Yuval Shahar. A framework for knowledge-based temporal abstraction. *Artificial Intelligence*, 90:79-133, 1997.

[Verduijn et al., 2005] Marion Verduijn, Arianna Dagliati, Lucia Sacchi, Niels Peek, Riccardo Bellazzi, Evert de Jonge, Bas de Mol. IC prediction from patient monitoring data: a comparison of two temporal abstraction procedures. In *Proceedings of the AMIA 2005* Annual Symposium,2005.

Temporal Characterization of Ill-known Diseases

Silvana Badaloni and **Marco Falda**
University of Padova via Gradenigo 6, 35100 Padova, Italy
silvana.badaloni@unipd.it, marco.falda@unipd.it

Abstract

In the identification of unknown diseases the temporal evolution is one of the most important aspects. Very often information about a new disease is imprecise and vague, due to the fact that the disease itself is hardly recognized by studying the symptoms of the patients. To this aim, we have applied a Fuzzy Temporal Reasoning system we have developed to the case of Severe Acute Respiratory Syndrome (SARS). The system is able to handle both qualitative and metric temporal knowledge affected by vagueness and uncertainty. In this preliminary work, we show how the fuzzy temporal framework allows us to represent temporal evolutions of symptoms in different patients thus making possible to deduce characteristic periods of an unknown disease such as SARS was.

1 Introduction

Unpurified drinking water, improper use of antibiotics, local warfare, massive refugee migration and changing social and environmental conditions around the world have fostered the spread of new and potentially devastating viruses and diseases and have made surveillance for infectious diseases a public health need [Berkelman *et al.*, 1994; Berkelman *et al.*, 1996].

Medical examiners and coroners certify approximately 20% of all deaths that occur within the United States and can be a key source of information regarding infectious disease deaths [Wolfe *et al.*, 2004]. A computer-assisted search tool could detect infectious disease deaths from a medical examiner database, thereby reducing the time and resources required to perform such surveillance manually.

Medical diagnosis is a field in which imprecise information about symptoms and events can appear; for example this happens when the physician must interpret the description of a patient, or when a new disease appears and typical patterns have to be discovered. Human reasoning about uncertainty is often poorly coherent, while an automated system can guarantee an homogeneous treatment of vague data.

The framework of Fuzzy Sets can be regarded as the most suitable formalism to deal with imprecision intrinsic to many medical problems [Steimann, 2001] especially when epidemiological studies cannot be developed since statistical data are lacking or insufficient. Therefore, fuzzy-set based approaches allow one the ease of expression offered by symbolic models avoiding the unwieldiness of analytical alternatives, bridging the gap between the discrete world of reasoning and the continuity of reality [Steimann, 2001].

The most common use of fuzzy temporal reasoning in medical diagnosis is for representing the temporal evolution of the manifestations of patient diseases in order to recognize the typical temporal evolution of a specific disease, and then make a diagnosis. In [Wainer and Rezende, 1997; Wainer and Sandri, 1999] it has been shown how diagnostic reasoning can be fundamental to detect infectious diseases on the basis of temporal information. For example, Staphylococcus aureus and short therm Bacillus cereus are the only possible bacterial causes for nausea and vomiting within 1-6 h from ingestion of contaminated matter while a patient with botulism will only have those symptoms only 18-36 h after ingestion. In [Badaloni and Falda, 2005] we have addressed the problem of recognizing an exanthematic disease starting from the approximated knowledge of the temporal sequence of its symptoms and from the imprecise data coming from a patient's description. In exanthematic diseases the temporal evolution and the duration of the symptoms are sufficient to discriminate them, therefore the diagnosis can be based on the identification of typical temporal structures.

In this preliminary work, we address the problem of diagnostic reasoning from a different point of view. We start from a set of data concerning the temporal evolution of symptoms of different patients affected by an unknown or ill-known disease. It should be noticed that, at the moment, we don't treat data extracted from laboratory tests or temporal biomedical patterns that would require the use of advanced methodologies to deal with [Sacchi *et al.*, 2005] but, more simply, we treat temporal data relative to common symptoms of a limited group of patients. Once represented such data in a fuzzy constraint temporal network, we propose a method for abstracting general temporal features characterizing the disease, if they exist (e.g. the incubation period). To this aim we utilize a model [Badaloni *et al.*, 2004] based on Fuzzy Temporal Constraint Networks. The constraint based system allows one to manage in a unified framework temporal information of different types. In fact, temporal information coming from the domain may be both

qualitative such as "the interval I1 with fever precedes the interval I2 with cough" or metric such as "fever lasts one day" or mixed such as "symptom m2 follows symptom m1 and starts at 8pm".

In this paper, we present an application of our temporal reasoning system in a case study: the Severe Acute Respiratory Syndrome (SARS). It is organized as follows: Section 2 describes our approach to integrate temporal information in presence of vagueness and uncertainty, Section 3 defines the medical problem under study, Section 4 reports the considered temporal data and shows how the problem can be modeled. Finally, the results are discussed in Section 5.

2 Qualitative and quantitative fuzzy temporal constraints

Let's first describe the different components of our integration model. To deal with qualitative temporal information the most famous approach is the Allen's Interval Algebra [Allen, 1983]; in this algebra each constraint is a binary relation between a pair of intervals, represented by a disjunction of *atomic relations*:

$$I_1 \ (rel_1, \ldots, rel_m) \ I_2$$

where each rel_i is one of the 13 mutually exclusive atomic relations that may exist between two intervals (such as *equal*, *before*, *meets* etc.).

Allen's Interval Algebra has been extended in [Badaloni and Giacomin, 2006] with the Possibility Theory [Dubois *et al.*, 1996] by assigning to every atomic relation rel_i a degree α_i, which indicates the *preference degree* of the corresponding assignment among the others

$$I_1 \ R \ I_2 \text{ with } R = (rel_1[\alpha_1], \ldots, rel_{13}[\alpha_{13}])$$

where α_i is the preference degree of $rel_i \ (i = 1, \ldots, 13)$; preferences can be defined in the interval $[0, 1]$. If we take the set $\{0, 1\}$ the classic approach is obtained.

Intervals are interpreted as ordered pairs $(x, y) : x \leq y$ of $\Re^2$, and soft constraints between them as fuzzy subsets of $\Re^2 \times \Re^2$ in such a way that the pairs of intervals that are in relation rel_k have membership degree α_k.

If temporal entities are points, the Point Algebra [Vilain *et al.*, 1989] and its fuzzy extension [Badaloni and Giacomin, 2006] have been considered. In the classical case, i.e. when temporal information is not affected by uncertainty and vagueness, a Qualitative Algebra QA that includes all the combinations that can occur between temporal points and intervals is defined in [Meiri, 1996], containing all the algebras: the Point Algebra PA , the Interval Algebra IA, the Point-Interval and Interval-Point Algebras PI and IP, referring to point-point, interval-interval, point-interval, interval-point relations. In order to build the fuzzy Qualitative Algebra QA^{fuz}, we have considered the corresponding fuzzy extensions PA^{fuz} and IA^{fuz} [Badaloni and Giacomin, 2002; Badaloni and Giacomin, 2006], PI^{fuz} and IP^{fuz} [Badaloni *et al.*, 2004].

Dealing with temporal metric information, traditional Temporal Constraint Satisfaction Problems (TCSPs) [Dechter *et al.*, 1991] have been extended to the fuzzy case [Marín *et al.*, 1997; Godo and Vila, 2001]. In most cases trapezoidal distributions have been used, since they seem enough expressive and computationally less expensive. We too adopt trapezoidal distributions: each trapezoid is represented by a 4-tuple of values describing its four characteristic points plus a degree of consistency α_i denoting its height.

$$T_k = \ll a_k, b_k, c_k, d_k \gg [\alpha_k]$$

with $a_k, b_k \in \Re \cup \{-\infty\}$, $c_k, d_k \in \Re \cup \{+\infty\}$, $\alpha_k \in (0, 1]$, $\ll$ is either `(` or `[` and $\gg$ is either `)` or `]`.

The points b_k and c_k determine the interval of those temporal values which are likely, whereas a_k and d_k determine the interval out of which the values are absolutely impossible.

As an example, let's consider the following sentence:

> "Patient P_1 had fever on February 27; within approximately 5 days he developed cough."

By setting the origin of time on February 27 and assuming an imprecision of one day, we can model this sentence as

$$P_1 : \quad \{(4, 4.5, 5.5, 6)\}$$

in Figure 1 its graphical representation is shown.

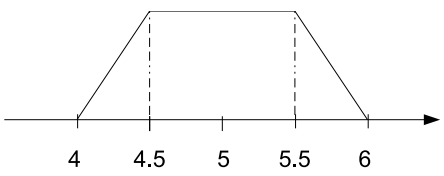


Figure 1: Example of trapezoidal possibility distribution

In our approach the following trapezoids can be modeled:

open triangle:$(a_i, a_i, a_i, d_i)[\alpha_i]$

open trapezoid:$(-\infty, -\infty, c_i, d_i)[\alpha_i]$

closed left semiaxis:$(-\infty, -\infty, d_i, d_i][\alpha_i]$

in this way the expressiveness of the language is increased with respect to e.g. [Barro *et al.*, 1994]. Besides, these trapezoids allow us to integrate qualitative constraints.

As far as operations between metric constraints are concerned, the usual operations i.e. inversion, conjunctive combination, disjunctive combination and composition have been defined.

2.1 About the integration

Fuzzy qualitative constraints and fuzzy metric constraints have been integrated together in a single framework [Badaloni *et al.*, 2004] defining two transformation functions $QUAN^{fuz}$ and $QUAL^{fuz}$ that allow to switch from the qualitative to the metric plane and vice versa. In the conversion from the metric to the qualitative plane a lot of information is lost, due to the fact that for example any metric constraint that represents a positive distance between temporal events is transformed into the "before" qualitative constraint; nothing instead is lost in the inverse conversion. For this reason, when the constraints to be represented are

of different nature the operations between qualitative and metric constraints are made, as far as possible, in the metric plane, in order to loose as less information as possible; therefore we always try to transform the qualitative constraints into metric constraints. There is a case however where it is not possible to remain in the metric plane; this happens when the composition operation involves a mixed qualitative relation. In this case it is necessary to transform the metric operand and to operate in the qualitative plane. Once the operations have been extended to the fuzzy case usual algorithms to solve CSPs can be easily generalized.

This way, we can manage temporal networks where nodes can represent both points and intervals, and where edges are accordingly labeled by qualitative and quantitative fuzzy temporal constraints. A more detailed description of our approach can be found in [Badaloni *et al.*, 2004].

2.2 Algorithms

The notions of local consistency have been extended too. In particular, local consistency has been expressed as the degree of satisfaction which denotes the acceptability of an assignment with respect to the soft constraints involved in the relative sub-network. According to [Dubois *et al.*, 1996], this degree of satisfaction corresponds to the least satisfied constraint.

Moreover, Path-Consistency and Branch & Bound algorithms have been generalized to the fuzzy case adding some relevant refinements that improve their efficiency. Path-consistency allows to prune significantly the search space while having a polynomial computing time.

In our integrated system embedding both qualitative and metric constraints composition and conjunction operations used in the algorithms depend on the type of operands, therefore they change according to the kind of constraints to be processed (qualitative or metric).

3 The SARS case

In the following we will consider the case of SARS, a kind of pneumonia spread from Far East in 2003. We take as reference one of the first articles written in that period [Poutanen *et al.*, 2003]; it is about the cases in Toronto. We will study only four patients of ten (Patients 1, 2, 7 and 8), because they are better described from the temporal point of view. In this initial study the scenario has been simplified, but it is sufficient to show the flexibility of the constraints that can be used to model a problem, but the system could be applied to tens of patients thus outperforming a human analysis in terms of coherence and consistency.

3.1 Description of the outbreak

The Toronto index case (Patient 1) and her husband traveled to Hong Kong to visit relatives from February 13 through February 23, 2003. They returned to their apartment in Toronto on February 23, 2003. Patient 1, a 78-year-old woman, had fever, anorexia, myalgias, a sore throat, and mild nonproductive cough two days after returning home. Two days later, she noted the development of increasing cough with dyspnea. She died three days later, on March 5, at home, nine days after the onset of her illness.

The index patient's 43-year-old son (Patient 2), had fever and diaphoresis on February 27. Within approximately five days he became afebrile, but concurrently, a nonproductive cough, chest pain, and dyspnea developed. Because of persistent symptoms, 4 days later he was assessed at a hospital and noted to have a fever (temperature, 39.8C) and an oxygen saturation of 82 percent while breathing room air. Despite intensive physiological support, multiorgan dysfunction syndrome developed, and he died on March 13, 2003, 6 days after admission, and 15 days after becoming ill.

As a result of media attention, three additional cases of SARS were identified. The first case was in a previously healthy 37-year-old female family physician of Asian descent (Patient 7) who saw Patient 2 and his wife on March 6, when they were both symptomatic. Patient 7 had a severe headache on March 9, followed by fevers (temperatures of up to 40C), myalgias, and malaise. Four days later, a nonproductive cough developed, and she was noted to have fever (temperature, 38.5C) and tachypnea with an oxygen saturation of 100 percent on room air.

The second additional identified case was in a 76-year-old man of non-Asian descent (Patient 8). Patient 8 was assessed in the emergency department on March 7 for atrial fibrillation and observed overnight on a gurney separated by a cotton curtain 1 to 2 m from Patient 2. Patient 8 was discharged home on March 8, and two days later he had fever (temperatures of up to 40C), diaphoresis, and fatigue. Despite receiving broad-spectrum antibiotics, oseltamivir, intravenous ribavirin, and intensive support, he died on March 21, 5 days after admission and 12 days after the onset of his illness.

4 Modelling the scenarios

The four patients were suspected to have SARS because the lived in the same apartment in Toronto and the symptoms were similar to those reported in Hong Kong, where Patient 1 spent a week before returning home.

Our aim is to characterize the incubation period, that is the period between the contagion and the first symptoms. To do this, we take into account the period during which the disease could have been got, the fever (as initial symptom), the cough, the contagion and the death.

For example the following temporal evolution can be abstracted from the previous observations about Patient 1:

- in travel from February 13 to February 23;
- 2 days later, fever;
- 2 days later, cough;
- 3 days later, death.

These temporal descriptions allow to build timetables for each patient in exam, as depicted in Figure 3. Note that metric information need to be specified as relative distances between temporal events, since this is the semantics of metric constraints.

Four distinct networks have been build for each patient starting from the previous data. In a preliminary phase the symptoms common to all patients have been identified by the physicians. There are seven significant points plus an interval (V_6) that represents the period during which the disease could be got, in the following called I.

The seven points become the network vertices and are identical in each network (Figure 3):

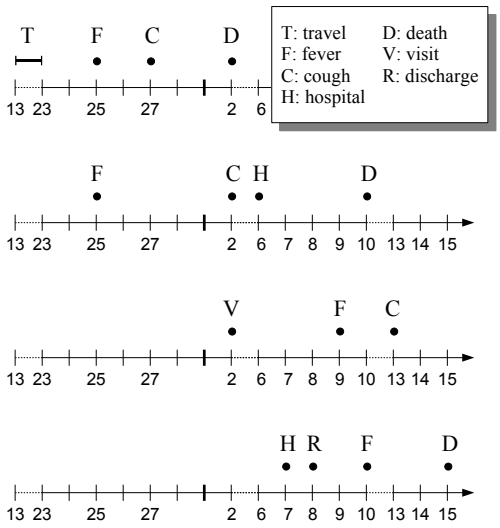


Figure 2: Timelines for the patient's data

1. V_0: t_0, the "origin of time";
2. V_1: begin of I;
3. V_2: end of I;
4. V_3: fever;
5. V_4: cough;
6. V_5: death;
8. V_7: contagion.

The "origin of time" has been set on February 23, that is the day in which Patient 1 returned home and infected her family. The end of period I coincides respectively with the death, the admission to hospital, the discharge to home and the one-day medical visit. The constraints that refer to a patient have been defined as in the following example, where we assume an uncertainty of half a day:

- about -10 days from V_0 to V_1:

$$V_0\{[-11, -10.5, -10, -9.5]\}V_1$$

- 0 days from V_0 to V_2

$$V_0\{=\}V_2$$

- about 2 days from V_2 to V_3

$$V_2\{[1, 1.5, 2.5, 3]\}V_3$$

- about 2 days from V_3 to V_4

$$V_3\{[1, 1.5, 2.5, 3]\}V_4$$

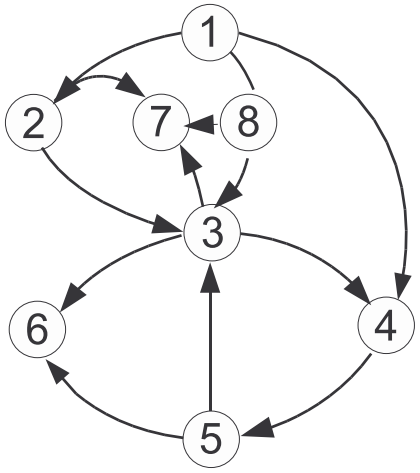


Figure 3: Graph of the problem

- about 3 days from V_4 to V_5

$$V_4\{[2, 2.5, 3.5, 4]\}V_5$$

Additionally in all four cases there are the following constraints that model the facts that the contagion must be before the first symptom and that the I period is characterized by a begin and an end, and that the contagion must be contained in period I.

$$V_7\{<\}V_3$$

$$V_6\{si\}V_1$$

$$V_6\{fi\}V_2$$

$$V_7\{d, s, f\}V_6$$

In this example all qualitative constraints have a degree of preference equal to 1, because here they are used only to link qualitative intervals with metric points. Metric constraints instead present, as said before, a trapezoidal possibility distribution (that is the set of all preference degrees in the domain) that sets the maximal plausibility to the assignments in the core between b and c, and states as impossible the values outside the range (a, d).

5 Results

The previous networks are then coded as XML files based on a XML schema designed to represent Fuzzy Temporal Networks, and then passed to the solver.

A consistency analysis of the temporal data gave the following incubation estimates in terms of constraints between the contagion V_7 and the fever V_3, being P_i the patients

P_1 : $V_3\{(-13.5, -12.5, -1.5, -1.0)\}V_7$
P_2 : $V_3\{(-4.5, -4.0, 0.0, 0.0]\}V_7$
P_7 : $V_3\{(-4.5, -3.5, -2.0, -1.0)\}V_7$
P_8 : $V_3\{(-5.0, -3.5, -1.5, -1.0)\}V_7$

That can be interpreted as:

P_1: approximately from 1 to 12 days;

P_2: approximately from 0 to 4 days;

P_7: approximately from 2 to 4 days;

P_8: approximately from 1 to 4 days.

The output of the solver is still a valid XML file that could be possibly modified and processed again. In this prototype version the application has a command line interface, but since it works with XML the output could be easy formatted to be shown on a Web interface using XML stylesheets.

In Figures 4, 5, 6 and 7 the period I is represented as a hatched rectangle and the incubation period as an interval between the begin of the period I and the onset of the first symptoms.

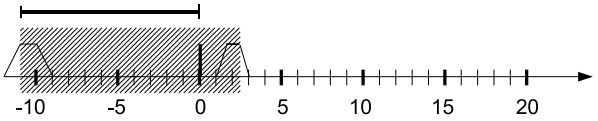


Figure 4: Evolution in Patient 1

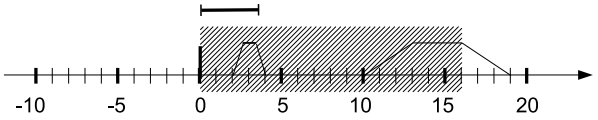


Figure 5: Evolution in Patient 2

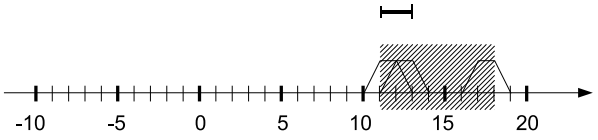


Figure 6: Evolution in Patient 7

In this paper to merge information coming from different sources the intersection operations has been considered: therefore real incubation period is the intersection of the four estimated periods, that is about 2-4 days. In this case the deduction has been not too difficult, but in a more realistic scenario with a lot of temporal data to analyze an automated reasoning system could be very helpful for a physician that has to figure out the temporal evolution of the symptoms of a new disease. Besides, a more robust technique should take into account the weight of data coming from different sources in order to reduce the influence of less common cases (possibly wrong).

6 Conclusions

In this paper we have shown an application of our temporal constraint solver in a medical domain; this application could support the physician to characterize temporally an unknown disease by aid him to deduce its temporal evolution. Our solver extends the classic temporal constraints by allowing to specify uncertainty and vagueness; this is fundamental in the context of unknown illnesses.

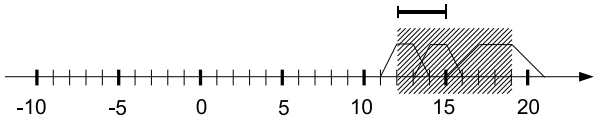


Figure 7: Evolution in Patient 8

In order to make the system more useful a first enhancement will be the management of constraint classes; in this way, it will be possible to merge automatically the deduced durations in order to identify the most plausible.

Moreover, as in [Wainer and Sandri, 1999; Keravnou, 2002], a real diagnosis expert system should consider also atemporal aspects of diseases. As future work we intend to enrich our system by addressing also these aspects.

References

[Allen, 1983] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(1):832–843, 1983.

[Badaloni and Falda, 2005] S. Badaloni and M. Falda. Discriminating exanthematic diseases from temporal patterns of patient symptoms. In *LNAI 3581*, pages 33–42, Springer, Germany, 2005.

[Badaloni and Giacomin, 2002] S. Badaloni and M. Giacomin. Fuzzy extension of interval-based temporal subalgebras. In *Proc. of IPMU 2002*, pages 1119–1126, Annecy, France, 2002.

[Badaloni and Giacomin, 2006] S. Badaloni and M. Giacomin. The algebra IA^{fuz}: a framework for qualitative fuzzy temporal reasoning. *Artificial Intelligence*, 170(10):872–908, 2006.

[Badaloni *et al.*, 2004] S. Badaloni, M. Falda, and M. Giacomin. Integrating quantitative and qualitative constraints in fuzzy temporal networks. *AI Communications*, 17(4):183–272, 2004.

[Barro *et al.*, 1994] S. Barro, R. Marín, J. Mira, and A. Paton. A model and a language for the fuzzy representation and handling of time. *Fuzzy Sets and Systems*, 175:61–153, 1994.

[Berkelman *et al.*, 1994] R. L. Berkelman, R. T. Bryan, M. T. Osterholm, J. W. LeDuc, and J. M. Hughes. Infectious disease surveillance: a crumbling foundation. *Science*, 264:368–370, 1994.

[Berkelman *et al.*, 1996] R. L. Berkelman, R. W. Pinner, and J. M. Hughes. Addressing emerging microbial threats in the united states. *JAMA*, 275:315–317, 1996.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[Dubois *et al.*, 1996] D. Dubois, H. Fargier, and H. Prade. Possibility theory in constraint satisfaction problems:

Handling priority, preference and uncertainty. *Applied Intelligence*, 6:287–309, 1996.

[Godo and Vila, 2001] L. Godo and L. Vila. Possibilistic temporal reasoning based on fuzzy temporal constraints. In *Proc. of IJCAI95*, pages 1916–1922, 2001.

[Keravnou, 2002] E. T. Keravnou. Temporal constraints in clinical diagnosis. *Journal of Intelligent and Fuzzy Systems*, 12(1):49–67, 2002.

[Marín *et al.*, 1997] R. Marín, M. A. Cárdenas, M. Balsa, and J. L. Sanchez. Obtaining solutions in fuzzy constraint network. *International Journal of Approximate Reasoning*, 16:261–288, 1997.

[Meiri, 1996] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence*, 87:343–385, 1996.

[Poutanen *et al.*, 2003] Susan M. Poutanen, Donald E. Low, Bonnie Henry, Sandy Finkelstein, David Rose, Karen Green, Raymond Tellier, Ryan Draker, Dena Adachi, Melissa Ayers, Adrienne K. Chan, Danuta M. Skowronski, Irving Salit, Andrew E. Simor, Arthur S. Slutsky, Patrick W. Doyle, Mel Krajden, Martin Petric, Robert C. Brunham, and Allison J. McGeer. Identification of severe acute respiratory syndrome in canada. *The New England Journal of Medicine*, 348(20):1995–2005, 2003.

[Sacchi *et al.*, 2005] L. Sacchi, R. Bellazzi, C. Larizza, R. Porreca, and P. Magni. Learning rules with complex temporal patterns in biomedical domains. In *LNAI 3581*, pages 23–32, Springer, Germany, 2005.

[Steimann, 2001] F. Steimann. On the use and usefulness of fuzzy sets in medical AI. *Artificial Intelligence in Medicine*, 21:131–137, 2001.

[Vilain *et al.*, 1989] M. Vilain, H. Kautz, and P. van Beek. Constraint propagation algorithms for temporal reasoning: a revised report. In J. de Kleer D. S. Weld, editor, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381, San Mateo, CA, 1989. Morgan Kaufmann.

[Wainer and Rezende, 1997] J. Wainer and A. Rezende. A temporal extension to the parsimonious covering theory. *Artificial Intelligence in Medicine*, 10:235255, 1997.

[Wainer and Sandri, 1999] J. Wainer and S. Sandri. Fuzzy temporal/categorical information in diagnosis. *Journal of Intelligent Information Systems*, 11:9–26, 1999.

[Wolfe *et al.*, 2004] M Wolfe, K. B. Nolte, and Yoon S. S. Fatal infectious disease surveillance and the medical examiner database. *Emerg. Infect. Dis.*, 2004.

Paper session: *Signal and Image Processing*

Cancer area characterization by non-parametric clustering

U. Castellani, M. Cristani, P. Marzola, V. Murino, E. Rossato*, A. Sbarbati†

Abstract

The application of machine learning techniques to open problems in different medical research fields appears to be stimulating and fruitful, especially in the last decade. In this paper, a new method for MRI data segmentation is proposed, which aims at improving the support of medical researchers in the context of cancer therapy. In particular, our effort is focused on the processing of raw output obtained by Dynamic Contrast-Enhanced MRI (DCE-MRI) techniques. Here, morphological and functional parameters are extracted, which seem indicate the local development of cancer. Our contribute consists in organizing automatically these output, separating MRI slice areas with different meaning, in a histological sense. The technique adopted is based on the Mean-Shift paradigm, and it has recently shown to be robust and useful for different and heterogeneous segmentation tasks. Moreover, the technique appears to be predisposed to numerous extensions and medical-driven optimizations.

1 Introduction

Segmentation is a vast and complex domain, both in terms of problem formulation and resolution techniques. It consists in formally translating the delicate visual notions of homogeneity and similarity, and defining criteria which allow their efficient implementation [Petitjean, 2002]. The goal is to partition the source data into meaningful pieces, i.e. those parts corresponding to the different entities, in the physical and semantical sense of the application envisioned. Roughly speaking, the segmentation methods can be categorized into two main classes: *edge-based* and *region-based* [Petitjean, 2002]. In the former, features corresponding to part boundaries are first detected and then regions are built, each one formed by sets of points delimited by the same boundary. In the latter, points sharing the same similarity property are grouped together. In particular, three are the most popular approaches to region-based segmentation: *split-and-merge* methods, identified by a top-down paradigm; *region-growing* methods, that adopt a bottom-up paradigm, and *clustering-based* methods, based on the projection of the points onto a higher dimensional space where the clusters (i.e., segments) are recovered by defining some particular distance functions [Jain *et al.*, 1999a].

In this paper, we apply a recently proposed clustering-based technique for the analysis of data, which considers as leading framework the Mean Shift (MS) clustering paradigm, proposed in [Comaniciu and Meer, 2002]. The main underlying idea of such non parametric approach is that the data space is regarded as an empirical probability density function to estimate. The MS procedure operates by shifting a fixed size estimation window, i.e., *the kernel*, from each data point towards a local mode, denoted, roughly speaking, as a high concentration of points. The points converging to the same mode are considered as belonging to the same region.

Although MS has shown to be a powerful technique for several research fields such as image and video segmentation, tracking, clustering and data mining [Comaniciu and Meer, 2002; Collins, 2003; Georgescu *et al.*, 2003], very few work has been derived from it in the context of medical multidimensional data segmentation.

In this paper, the MS paradigm is applied to perform segmentation of multidimensional data, obtained using Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI). Briefly speaking, DCE-MRI techniques represent non-invasive ways to discover symptoms of local tumor growth, based on a manually-driven feature extraction step that operates on the MRI imagery.

As explained in the following, our method bring two advantages to the current state of the DCE-MRI analysis. First, it permits a more accurate feature extraction step, that here operates in an *automatic* fashion. Second, it permits to fasten the analysis itself, ensuring a higher throughput, that turns out to be useful in the case of massive analysis.

The rest of the paper is organized as follow. In Section 2, an overview of the previous work done in the context

*U. Castellani, M. Cristani, V. Murino and E. Rossato are with the Dipartimento di Informatica, University of Verona, Strada le Grazie 15, 37134 Verona (Italy). Contacts: U. Castellani, e-mail umberto.castellani@univr.it.

†P. Marzola and A. Sbarbati are with the Department of Morphological and Biomedical Sciences, Anatomy and Histology Section, University of Verona, P.le Scuro 10 - Policlinico B.go Roma - 37134 Verona (Italy). Contacts: P. Marzola, e-mail pasquina.marzola@univr.it.

of medical data segmentation is provided; subsequently, in Section 3, the necessary medical background is provided, considering the classical DCE-MRI experimental methodology, in the contest of the tumor development monitoring. This section will elucidate the nature of the data managed; moreover, here it will be possible to deeply understand the advantages brought by our method. In Section 4, the Mean Shift procedure is explained, connecting it with a classical pattern recognition procedure, i.e. the Parzen Windows estimation method. In Section 5, the technical details of the proposed method are reported. Results are shown in Section 6, also compared with a state of the art method, and, finally, Section 7 concludes the paper.

2 Previous Works

In the realm of medical data segmentation, several works have been introduced, especially for MRI clustering and classification [Windishberger *et al.*, 2003; Dimitriadou *et al.*, 2004; Zhang and Chen, 2004; Wismuller *et al.*, 2006; Arulmurgan *et al.*, 2005; Wei and Yang, 2005; Jain *et al.*, 1999b; Scarth *et al.*, 1995; Castellani *et al.*, 2005]. Most proposed methods are based on the *K-Means* algorithm [McQueen, 1967; Han and Kamber, 2000]. In [Windishberger *et al.*, 2003; Zhang and Chen, 2004], a variant of the K-Means has been implemented, called *fuzzy C-Means* (FCM) [Scarth *et al.*, 1995; Jain *et al.*, 1999b; Dimitriadou *et al.*, 2004]. Such variant takes advantages of fuzzy logic algorithms to enhance clustering performance. In particular, the FCM algorithm assigns pixels to fuzzy clusters without labels. Unlike the hard clustering methods which force pixels to belong exclusively to one class, FCM allows pixels to belong to multiple clusters with varying degrees of memberships. In [Windishberger *et al.*, 2003] the clustering of MRI time series have been performed for the identification and separation of artifacts as well as quantification of expected novel information on brain activities. In [Zhang and Chen, 2004] the authors focused on the methodological aspect of the *fuzzy C-Means* by introducing a kernel-induced distance metric and a spatial penalty on the membership functions. The proposed approach has proved to be more robust to noise and other artifacts with respect to standard algorithms. In [Castellani *et al.*, 2005] the authors proposed a DCE-MRI clustering approach, coupled with a Information Visualization module, in which a Bayesian development of the K-Means was applied. Here the add-on is that the number of the clusters is automatically computed; the algorithm is similar in spirit to the X-Means algorithm proposed by [Pelleg and Moore, 2000]. In [Dimitriadou *et al.*, 2004], a quantitative comparison of MRI cluster analysis has been reported.
With respect to the proposed evaluation, the results clearly show that approaches based on k-means algorithm perform significantly better than all the other methods.
More complex techniques have been proposed in [Wismuller *et al.*, 2006; Arulmurgan *et al.*, 2005; Wei and Yang, 2005] which are based on neural networks or genetic algorithms [Jain *et al.*, 1999b; Han and Kamber, 2000], but they are time consuming and therefore are not suitable for interactive applications.

3 The DCE-MRI analysis

The main purpose of DCE-MRI analysis is to accurately monitor the local development of cancer, eventually subject to different treatments.
The traditional criteria to assess the tumor response to treatment is based on the local measurement of tumor size change. This phenomenon is due to the local *angiogenesis*, i.e., the process of growth of new vessels which provide the tumor tissue with nutrients. In consequence, various angiogenesis inhibitors have been developed to target vascular endothelial cells and to block tumor angiogenesis.

Recently, a different and more appealing indicative symptom of the cancer development has been analyzed, i.e. the tissue vascularization [Marzola *et al.*, 2004]. Roughly speaking, vascular effect may precede, by a remarkably long time interval, the effect on tumor growth. For these reasons, the assessment of antiangiogenic compounds requires imaging methods that can detect early vascular alterations.

DCE-MRI techniques play a relevant role in this field [Marzola *et al.*, 2004]. The final aim is to provide quantitative measures that indicate the level of vascularization in the cancer tissue, eventually treated with antiangiogenic compounds, in a *non-invasive* way.
Roughly speaking, the DCE-MRI analysis can be divided in the following steps (see Fig.1)[1]: 1) injecting macromolecular contrast agents in the tissue being analyzed; 2) producing MRI image sets of different slices of the tissue; 3) extracting morphological and functional parameters such as *fractional plasma volume* (fPV) and *transendothelial permeability* (kPS), that model the tissue vascularization; in practice, to each point of the MRI image is associated a pair consisting of fPV and kPS values; 4) manually selecting a Region Of Interest on the MRI slices, in order to isolate the highly vascularized local tumoral area; usually this area is ring-shaped and separates a necrotic area (that lies in the center of the ring) from the external healthy portion of the tissue; 5) averaging the values of fPV and kPS in such ring-shaped area, obtaining for each slice a couple of fPV and kPS mean values, that indicate the overall level of vascularization. This process has been recently tested using a well-known anti-cancer treatment [Marzola *et al.*, 2005], evidencing that the fPV and kPS parameters well describe the effectiveness of the treatment, as checked by additional histological analysis; as we see in the following, we take as experimental data-set the one coming from this research.

In this paper, we strongly improve the process above, providing an automatic method of data segmentation; the proposed technique is applied to this particular kind of analysis, but we suppose it can be also applied in general in the DCE-MRI context. In detail, we focus on steps 3), 4) and 5); our method takes as input the functional parameters fPV and kPS obtained in step 3); in an automatic fashion, it is able to segment areas that experimentally corresponds to the tumoral area extracted by hands in step 4); note that originally this step was driven by histological and physio-

[1] The procedure listed above comes from the investigation detailed in [Marzola *et al.*, 2005], that in turn presents additional similar researches

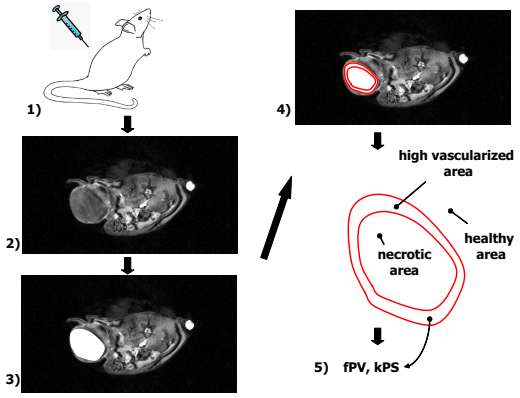


Figure 1: DEC-MRI analysis: example of DEC-MRI analysys procedure; 1) contrast agent injection; 2) MRI image acquisition 3) kPS,fPV extraction; for clarity, here the bright zone highlights the tumor4) ring shape ROI drawing by hand; 5) mean kPS,fPV values computation

logical a-priori considerations, being the ring-shaped zone segmented by hand by an human operator.

The advantage brought by the proposed approach is twofold: firstly and mostly important is that, given a DCE-MRI slice, we provide a region of points composed by an ensemble of fPV and kPS values that individuate separate groups; note that the partition is histologically meaningful, and not relies on a-priori manual settings. Secondly, such a segmentation is produced automatically and quickly (5 seconds, versus the 4-5 minutes needed for an accurate manual setting), thus fasting the analysis process listen above.

4 Mean Shift

The Mean Shift procedure is a dated non-parametric density estimation technique [Fukunaga, 1990; Comaniciu and Meer, 2002]. The main underlying idea is that the data feature space is regarded as an empirical probability density function to estimate: therefore, a big concentration of points that fall near the location $\mathbf{x}$ indicates a big density near $\mathbf{x}$.

The theoretical framework of the mean shift arises from the Parzen Windows [Duda *et al.*, 2001] basic expression, i.e. the kernel density estimator, that is

$$\hat{f}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}_i) \tag{1}$$

where $\hat{f}(\mathbf{x})$ represents the approximated density calculated in the d-dimensional location $\mathbf{x}$, n is the number of available points and

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x}). \tag{2}$$

Here above, $K_{\mathbf{H}}$ can be imagined as a weighted window used to estimate the density, dependent on the kernel K and the symmetric positive definite $d \times d$ bandwidth matrix $\mathbf{H}$. The function K is a bounded function with compact support (for full details, see [Comaniciu and Meer, 2002]); the bandwidth matrix codifies the uncertainty associated to the whole feature space.

In the case of particular radial symmetric kernels (see [Comaniciu and Meer, 2002]), K can be specified using only a 1-dimensional function, the *profile* $k(\cdot)$, equal for each dimension. Moreover, if we assume independence among the feature dimensions and equal uncertainty over them, the bandwidth matrix can be rewritten as proportional to the identity matrix $\mathbf{H} = h^2\mathbf{I}$. Under such hypotheses, Eq. 2 can be rewritten as:

$$\hat{f}_{h,k}(\mathbf{x}) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \tag{3}$$

where $c_{k,d}$ is a normalizing constant, n is the number of points available, and $k(\cdot)$ is the kernel profile; in Eq.(3) it is easy to note that $k(\cdot)$ models how strongly the points are taken into account for the estimation, in dependence with their distance h to $\mathbf{x}$.

Mean Shift extends this "static" expression, differentiating (3) and obtaining the gradient of the density, i.e.:

$$\hat{\nabla} f_{h,k}(\mathbf{x}) = \frac{2c_{k,d}}{nh^d}\left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i-\mathbf{x}}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}_i-\mathbf{x}}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i-\mathbf{x}}{h}\right\|^2\right)} - \mathbf{x}\right] \tag{4}$$

where $g(x) = k'(x)$. In the above equation, the first term in square brackets is *proportional* to the normalized density gradient, and the second term is the *Mean Shift* vector, that is guaranteed to point towards the direction of maximum increase in the density [Comaniciu and Meer, 2002]. Therefore, starting from a point $\mathbf{x}_i$ in the feature space, the mean shift produces iteratively a trajectory that converges in a stationary point $\mathbf{y}_i$, representing a mode of the whole feature space.

5 The proposed method

Our segmentation method can be thought as a clustering process, derived from the approach proposed in [Comaniciu and Meer, 2002]. Briefly speaking, the first step of such process is made by applying the Mean Shift procedure to all the points $\{\mathbf{x}_i\}$, producing several convergency points $\{\mathbf{y}_i\}$. A consistent number of close convergency locations, $\{\mathbf{y}_i\}_l$, indicates a mode μ_l. The labeling consists in marking the corresponding points $\{\mathbf{x}_i\}_l$ that produces the set $\{\mathbf{y}_i\}_l$ with the label l. This happens for all the convergency location $l = 1, 2, \ldots, L$.

In this paper, we consider each point of the MRI as a d-dimensional entity, living in a *joint domain*. In specific, each $\mathbf{x}_i$ is composed by the pair $\mathbf{x}_s \in \mathcal{R}^2$ of spatial coordinates relative to the x,y image axes (the *forming the spatial sub-domain*) and the pair $\mathbf{x}_c \in \mathcal{R}^2$ of fPV and kPS coefficients (forming the *coefficients sub-domain*). For each sub-domain we assume Euclidian metric.
In order to explore the joint domain, a multivariate kernel is used [Comaniciu and Meer, 2002; Wang *et al.*, 2004], that has the form

$$K_{h_s,h_c}(\mathbf{x_i}) = \frac{C}{h_s^2 h_c^2} k\left(\left\|\frac{\mathbf{x}_{i,s}}{h_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}_{i,c}}{h_c}\right\|^2\right) \tag{5}$$

where $\mathbf{x}_{i,s}$ indicates the spatial coordinates of the $i-$th point and so on for $\mathbf{x}_{i,c}$; C is a normalization constant, and h_s,h_c are the kernel bandwidths for each sub-domain. These values give to each feature domain the intuitive concept of "importance": strictly speaking, the bigger the related kernel bandwidth, the less important that feature. In other words, a big amplitude of the kernel tends to agglomerate points in few convergence locations, while a small kernel highlights better local modes, encouraging cluster separations.
In this paper, we use the Epanechnikov kernel [Comaniciu and Meer, 2002], that can be described by the profile

$$k(x) = \begin{cases} 1 - x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

that differentiated leads to the uniform kernel, i.e. a d-dimensional unit sphere.

6 Experiments

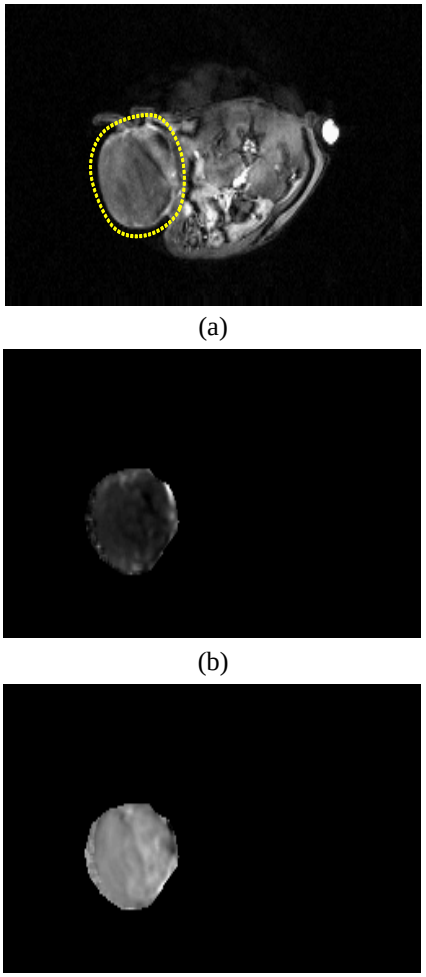
The experiments performed in this paper are related to a series of investigations on the effects of a particular tumor treatment, using DCE-MRI techniques. Here, human mammary carcinoma fragments (13762 MAT B III) were subcutaneously injected in the right flank of 42 female rats at the level of the median-lateral. The details about the experiment outstand the scope of the paper (see [Marzola *et al.*, 2005] for details); anyway, the interesting aspects are the following: 1) after the injection of a contrast compound in the animals, MRI images were acquired for tumor localization and good visualization of extratumoral tissues. The dynamic evolution of the Signal Intensity in MR images is analyzed using a two compartments tissue model in which the contrast agent can freely diffuse between plasma and interstitial space. The kPS and fPV values are obtained pixel by pixel by fitting the theoretical expression to experimental data. After that, data were transferred on a PC for analysis. Images were analyzed on a ring-like region-of interest (ROI) basis to obtain the average value of kPS and fPV within it: in each animal, the central 5 slices of the 3D data set were analyzed.

In our case, we select a reasonable section of the MRI slice, (Fig.2 (a); in principle, the analysis can be applied to the entire slice); in this area, we calculate the related kPS and fPV coefficients (Fig.2 (b) and (c)) and we perform MS segmentation using a uniform kernel for each subdomain.

After the normalization of the data, that brought all the values between 0 and 1, the kernel bandwidth widths have been easily chosen. In particular, after some (less than 10) trials the bandwidth values have been set to $[0.3, 0.3, 0.03, 0.06]$ for the spatial (first pair of values), and the coefficient sub-domain (second pair of values), respectively.

The current implementation of the proposed method is working under the Matlab 7 environment. The segmentation process takes ~ 5 sec. each for each MRI slice.

A result obtained for the slice shown in Fig.2 is shown in Fig.3 (b).
As comparative test, we perform the same analysis using the approach based on the Bayesian development of the K-Means, presented in [Castellani *et al.*, 2005]; the result is

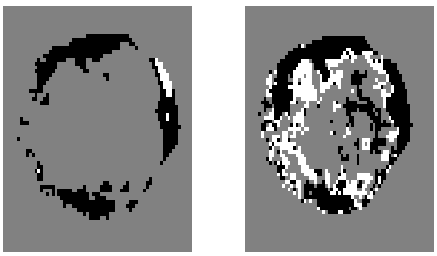


(a) (b) (c)

Figure 2: DEC-MRI: (a) example of MRI slice, where a contrast agent has been introduced into the tissue before the image acquisition; a rough section of the tissue was selected in order to apply our algorithm, highlighted by the dotted circle; (b) intensity image representing the fPV values; (c) intensity image representing the kPS values; in b) and c), the higher the values of the parameters, the brighter the color of the correspondent pixels.

shown in Fig.3 (a). As one can see, our approach identify two clusters: both of them have a different histological meaning; the darker cluster, roughly forming a ring, indicates effectively the zone of the tumor more affected by vascularization. This zone corresponds to the one segmented by hand at steps 4) and 5) of the DCE-MRI analysis discussed in Sect.3. The second cluster, that spreads over the center, indicates another different zone, affected by high permeability with respect to the contrast agent. The result obtained using the X-Means based approach shows slightly only the circular high vascularized ring.

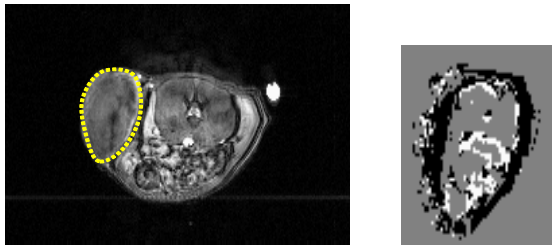
With the same experimental setup, we perform another two tests on the same DCE-MRI data set. As shown in Fig.4 (b) and (d), in both the cases the resulting segmentations show 2 clusters, i.e., an external high vascularized



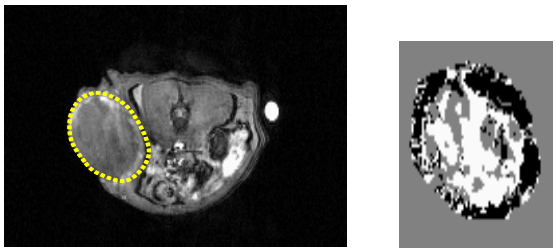
(a) (b)

Figure 3: Comparative results: (a) the segmentation obtained using the X-means method; (b) our approach

ring and a central necrotic spread zone, with precise histological meanings, as written above.



(a) (b)



(c) (d)

Figure 4: DCE-MRI results: on the left, the MRI images related to two different experiments, with the tumoral zone highlighted. On the right, the resulting segmentations

7 Conclusions

In this paper, we introduce a multidimensional segmentation technique derived by the Mean Shift (MS) procedure, aimed at improving the analysis and the characterization of tumor tissues. Briefly speaking, the multidimensional output obtained by a recent and non invasive tissue analysis, namely, the Dynamic Contrast-Enhanced MRI (DCE-MRI) technique is considered; the output of this technique, composed by spatial, morphological and functional tumor parameters is projected in a joint space, where an *automatic* clustering-based segmentation is performed; this process results in a histologically meaningful partition, that individuates tissue zones differently involved with the development of the tumor. The goals of the proposed method are two: 1) we permit an analysis of the tissue more precise and 2) fast than the manual analysis currently performed; these two results assess that the non-parametric paradigm derived from the MS strategy well behaves with medical segmentation issues, related to the DCE-MRI context. Further research is currently under study, specially devoted to make automatic the phase of kernel selection.

References

[Arulmurgan *et al.*, 2005] S. Arulmurgan, T. Selvi, and S. Alagappan. Mri image segmentation using unsupervised clustering techniques. In *Proceedings of International Conference on Computational Intelligence and Multimedia Applications*, pages 105–110, 2005.

[Castellani *et al.*, 2005] U. Castellani, C. Combi, P. Marzola, V. Murino, A. Sbarbati, and M Zampieri. Towards information visualization and clustering techniques for mri data sets. In *Conference on Artificial Intellicence in Medicine*, pages 315–319, 2005.

[Collins, 2003] R.T. Collins. Mean-shift blob tracking through scale space. In *CVPR (2)*, pages 234–240, 2003.

[Comaniciu and Meer, 2002] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[Dimitriadou *et al.*, 2004] E. Dimitriadou, M. Barth, C. Windshberger, K. Hornik, and E. Moser. A quantitative comparison of funtional mri. *Artificial Intelligence in Medicine*, 31:57–71, 2004.

[Duda *et al.*, 2001] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, second edition, 2001.

[Fukunaga, 1990] K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, second edition, 1990.

[Georgescu *et al.*, 2003] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 456–463, 2003.

[Han and Kamber, 2000] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann. August 2000.

[Jain *et al.*, 1999a] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[Jain *et al.*, 1999b] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[Marzola *et al.*, 2004] P. Marzola, A. Degrassi, L. Calderan, P. Farace, C. Crescimanno, E. Nicolato, A. Giusti, E. Pesenti, A. Terron, A. Sbarbati, T. Abrams, L. Murray, and F. Osculati. In vivo assessment of antiangiogenic activity of su6668 in an experimental colon carcinoma model. *Clin. Cancer Res.*, 2(10):739–50., 2004.

[Marzola *et al.*, 2005] P. Marzola, S. Ramponi, E. Nicolato, E. Lovati, M. Sandri, L. Calderan, C. Crescimanno, F. Merigo, A. Sbarbati, A. Grotti, S. Vultaggio, F. Cavagna, V Lo Russo, and F. Osculati. Effect of tamoxifen in an experimental model of breast tumor studied by dynamic contrast-enhanced magnetic resonance imaging and different contrast agents. *Investigative radiology*, 40(7):421–429., 2005.

[McQueen, 1967] J. B. McQueen. Some methods of classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[Pelleg and Moore, 2000] Dan Pelleg and Andrew Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000.

[Petitjean, 2002] Sylvain Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Comput. Surv.*, 34(2):211–262, 2002.

[Scarth *et al.*, 1995] G. Scarth, M. McIntyre, B. Wowk, and R. L. Somorjai. Detection of novelty in functional images using fuzzy clustering. In *Third Scientific Meeting of the International Society for Magnetic Resonance in Medicine*, pages 238–245, 1995.

[Wang *et al.*, 2004] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *ECCV (2)*, pages 238–249, 2004.

[Wei and Yang, 2005] L. Wei and Y. Yang. A study on several machine-learning methods for classification of malignant and benign clustered microcalcifications. *IEEE Transaction on Medical Imaging*, 24(3):371–380, 2005.

[Windishberger *et al.*, 2003] C. Windishberger, M. Barth, C. Lamm, L. Shroeder, H. Bauer, R. Gur, and E. Moser. Fuzzy cluster analysis of high-field functional mri data. *Artificial Intelligence in Medicine*, 29:203–223, 2003.

[Wismuller *et al.*, 2006] A. Wismuller, A. M. Baese, O. Lange, M. F. Reiser, and G. Leinsinger. Cluster analysis of dynamic cerebral contrast-enhanced perfusion mri time-series. *IEEE Transaction on Medical Imaging*, 25(1):62–73, 2006.

[Zhang and Chen, 2004] D. Q. Zhang and S. C. Chen. A novel kernelize fuzzy c-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine*, 32:37–50, 2004.

A procedure for automated filtering of ICU monitoring data using basic smoothing techniques and clinical judgement

Marion Verduijn[1,4], **Niels Peek**[1], **Evert de Jonge**[2], **Bas de Mol**[3]
[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
m.verduijn@amc.uva.nl

Abstract

This paper presents the first results from a study on automated filtering of monitoring data that are automatically recorded in information systems. Monitoring data often contain erroneous measurements and artifacts. In practice, experienced clinicians ignore particular measurements that they consider as unreliable when inspecting and using monitoring data. In this study, we investigate to what extent this clinical filtering of monitoring data can be imitated by basic smoothing techniques. We developed a procedure for automated filtering of monitoring data in which smoothing is used to classify measurements as artifacts and non-artifacts. In the procedure, clinical judgement of monitoring data is used as gold standard. The procedure is applied to ICU monitoring data, and evaluated for three different smoothing techniques.

1 Introduction

The intensive care unit (ICU) is one of the most data intensive environments in medicine. The treatment aims to keep a close watch to a patient's physiological condition and to intervene immediately when needed. Therefore, automated monitoring systems measure many physiological variables with high frequency to continuously check the patient's status. In modernly equipped ICUs, these measurements are automatically recorded in an ICU information system, a replacement of the paper-based patient record: it provides an comprehensive overview of the patient's condition and recent medical history, and is regularly inspected by the ICU physician to adjust treatment and instruct the nursing staff. ICU information systems are increasingly equipped with computerized medical assistants for supporting these tasks [Miksch *et al.*, 1996; Michel *et al.*, 2003; Charbonnier, 2005].

A necessary condition for optimal functioning of these information systems is that reliable patient data be recorded in these systems. Monitoring data, however, often contain inaccurate and erroneous measurements, also called *artifacts*. These measurements can, for instance, be due to movements of the patient, or to equipment malfunction. These measurements hamper clinical interpretation of the data. In practice, experienced clinicians ignore particular measurements that they consider as unreliable when inspecting and using monitoring data. Computerized medical assistants as implemented in information systems do not discern artifacts in monitoring data, though, and may therefore provide inaccurate support based on these measurements. Therefore, the monitoring data should be cleaned from artifacts and measurement errors prior to data usage.

Except for measurements that take theoretically impossible values (e.g., negative blood pressures), it is often not evident which measurements are artifacts. Although the general course of monitoring data is a smooth pattern without large changes within short periods of time, sudden changes cannot be considered as artifacts by definition. Sudden changes may reflect actual clinical events and may therefore contain information of patient's health state.

This paper presents the first results from a study on automated, retrospective filtering of monitoring data. In this study, we investigate to what extent basic smoothing techniques can be used to imitate filtering of monitoring data by experienced clinicians, without taking account of context information and domain knowledge. We develop a procedure for automated filtering of monitoring data in which a smoothing technique is used. The intended application of this procedure is prior to periodical consultations of the data in the ICU information system by ICU physicians, nursing staff, and computerized medical assistants. In the procedure, clinical judgement of monitoring data is used as gold standard: time series that are manually filtered by physicians are used to tune the procedure for particular types of physiological variable.

The paper is organized as follows. In Section 2, some preliminaries of the automated data filtering procedure are described. In Section 3, we introduce the monitoring data that is used in this study. Subsequently, the procedure of manual filtering by the ICU physicians and the procedure for automated filtering are described. Section 4 describes the results of manually filtering, and the results of applying the automated filtering procedure to the monitoring data. We conclude the paper with a discussion and conclusions in Section 5.

2 Preliminaries

Artifact data are often recognizable by their large deviation from the average level of the variable in question. A common approach to detect such data points is therefore to compare point-based values to averages of a series over

time. The main difficulty with this approach is that also mean values may change over time, and that such changes are sometimes rapid and sometimes slow. The estimator of the series average should therefore have the flexibility to adapt to the data. However, it should not be so flexible that it adapts to artifacts, as it is then no longer possible to detect them.

In practice, (time-dependent) averages are usually estimated by flexible regression techniques that originate from the field of data visualization and smoothing. These techniques are based on least-squares regression and obtain flexibility by utilizing a notion of locality (in time). In this study, we applied three conventional smoothing techniques for filtering time series data: kernel smoothing, local regression, and smoothing splines. We briefly describe these techniques in this section.

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a series of successive measurements, and $\mathbf{t} = (t_1, \ldots, t_n)$ be the associated measurement times, i.e., $t_i < t_{i+1}$, $i = 1, \ldots, n-1$. A *smoothing estimate* of $\mathbf{x}$ is a vector $f(\mathbf{t}) = (f(t_1), \ldots, f(t_n))$ that is 'close' to $\mathbf{x}$ (i.e., the difference between x_i and $f(t_i)$ is generally small), but varies less than $\mathbf{x}$ (i.e., the difference between $f(t_i)$ and $f(t_{i+1})$ is generally smaller than that between x_i and x_{i+1}).

Kernel smoothing and *local regression* are different generalizations of the simple moving average [Cleveland and Loader, 1996]. In the simple moving average, the smoothed series $f(t_1), \ldots, f(t_n)$ is composed of local averages of the time series. That is, for each measurement x_i, a local average is calculated based on the measurement itself and its neighborhood measurements. Increasing the neighborhood size tends to increase the smoothness of the series $f(t_1), \ldots, f(t_n)$.

In kernel smoothing, the neighborhood measurements of x_i are weighed using a *kernel function* to obtain a locally weighted average. The kernel function gives higher weight to the measurements in the neighborhood that are closer to t_i and lesser weight to those that are further away. In local regression, a polynomial regression function is fitted for each measurement x_i based on its neighborhood measurements. The polynomial degree of these functions can be zero (i.e., local linear regression), or higher.

Smoothing splines are regression functions of piece-wise polynomials that minimize a compromise between the data fit and the degree of smoothness by calculating the penalized residual sum of squares:

$$RSS(f, \lambda) \quad = \quad \sum_{i=1}^{n} \{x_i - f(t_i)\}^2 + \lambda \int \{f''(t_i)\}^2 dt$$

where λ is a smoothing parameter. The first part of the equation measures the closeness of function f to the data, and the second part penalizes its curvature based on the second derivative of the function; λ establishes a tradeoff between the two parts [Hastie *et al.*, 2001].

In this study, we applied these three smoothing techniques as implemented in the S-plus statistical software package. For each smoothing technique, some parameters can be chosen. We optimized the complexity of each smoothing function for only one parameter: the number of degrees of freedom for smoothing splines, and the neighborhood size for kernel smoothing and local regression. Common settings were used for the other parameters of these latter two techniques, namely the gaussian kernel function for kernel smoothing, and a polynomial degree of two for local regression. Compared to the neighborhood size, the choice of these parameters are less important [Wand and Jones, 1995].

3 Data and methods

3.1 Monitoring data

In this study, monitoring data were used of the department of Intensive Care Medicine of the Academic Medical Center in Amsterdam, the Netherlands. At this department, the critically ill patients are monitored by Philips monitoring systems. During patient care, these monitoring data are sampled with a frequency of one measurement per minute to be recorded in the Metavision ICU information system developed by iMDsoft[1].

Our study is restricted to three physiological variables that concern the cardiovascular system: mean arterial blood 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 (a duration 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 2693, 3145, and 2005 minutes, respectively.

3.2 Manual filtering procedure

Four senior ICU physicians from the Academic Medical Center (where the data were recorded) were asked to inspect the 30 time series and point out individual data points that should be removed. The results of this manual filtering were used as gold standard during tuning and evaluating of the automated filtering procedure. The manual filtering procedure was carried out in two steps.

First, all four physicians were provided with paper versions of the 30 time series, and asked individually to mark data points they judged to be 'questionable'. The formal rule was to mark data points that they suspected to not reflect the actual health status of the patient at the time of measurement, and that they would therefore neglect in clinical practice. Removal of these points would therefore not result in a loss of information with respect to the patient's health status, but rather clean the data from disturbances that would be ignored by clinicians anyway. During this

[1] www.imdsoft.com

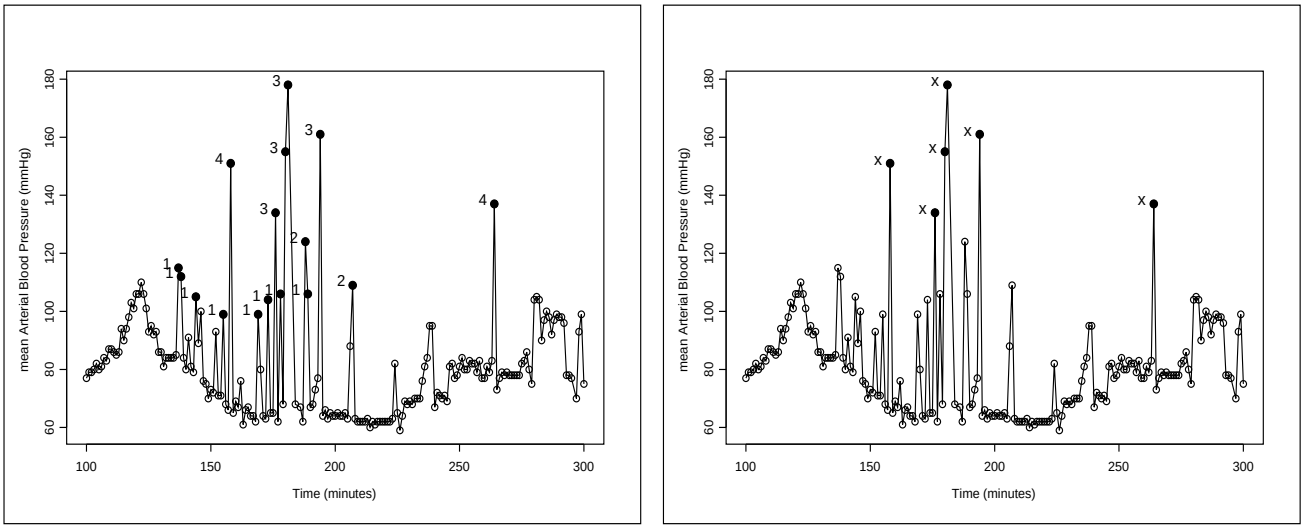


Figure 1: Results of the manual filtering procedure on a series of 200 mean arterial blood pressure measurements. Shaded circles represent data points that were judged to not reflect the actual health status of the patient, (a) by the four ICU physicians individually, where the associated numbers correspond to the number of physicians having that judgement, and (b) after reaching agreement during the consensus meeting.

task the physicians were provided with context information of the time series to be judged, such as measurements of other physiological variables that were recorded simultaneously on the same patient, and data of concurrent therapy (e.g., medication) and fluid administration.

Second, a consensus meeting was organized in which the four ICU physicians involved were asked to harmonize their individual judgements. During this meeting, two additional researchers (MV, NP) were present to guard consistency in the consensus judgements. Again, context information was provided, as well as the initial judgements of all four physicians.

3.3 Automated filtering procedure

In the automated filtering procedure, the measurements in the time series were classified as artifacts and non-artifacts by their deviance from the time-dependent average level of the variable. For that purpose, a smoothing technique was applied to the time series to obtain smoothed estimates of the measurements, and the deviance of the measurements was quantified in terms of the squared residuals. Based on these residuals, we examined to what extent measurements with a large deviance are judged as artifacts by physicians.

We use the following notation to describe the procedure in more detail. Let each time series of variable type v (i.e., ABPm, CVP, HR in this study) again be denoted by $\mathbf{x} = (x_1, \ldots, x_n)$, and the associated measurement times by $\mathbf{t} = (t_1, \ldots, t_n)$. Furthermore, let $\mathbf{y} = (y_1, \ldots, y_n)$ denote a binary variable with the associated clinical judgement of the time series as obtained in the manual filtering procedure; 1 is used to encode the artifacts and 0 to encode the non-artifacts.

The procedure was tuned and evaluated in five steps. In the first step, we obtained smoothed estimates of the time series of variable v, and calculated the corresponding squared residuals. These analyses were performed at the level of time series. So, we analyzed each of the time series of variable v as follows. We applied a given smoothing technique to the time series for a large number of parameter settings (i.e., the neighborhood size in kernel smoothing and local regression, and the number of degrees of freedom in smoothing splines in our study), and obtained a vector $f(\mathbf{t}) = (f(t_1), \ldots, f(t_n))$ with smoothed estimates for each parameter setting. For each measurement x_i in the time series, we defined the squared residual as

$$r_i^2 = (x_i - f(t_i))^2,$$

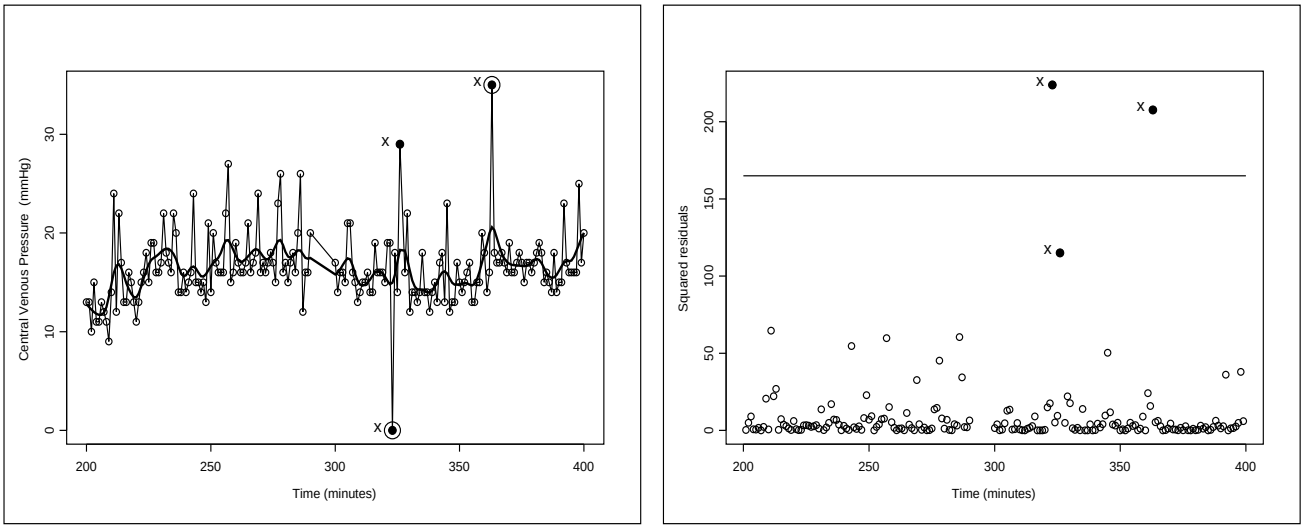
and obtained a vector $\mathbf{r}^2 = (r_1^2, \ldots, r_n^2)$ with residuals for each parameter setting.

The analysis was subsequently continued at the level of variable v. For that purpose, these vectors with residuals and the associated clinical judgement of all time series of variable v were combined in a single data set, and the resulting set was randomly split into a training and test sample.

In the third step of the analysis, the optimal smoothing parameter was chosen based on the training sample. For each parameter setting, the probability of being judged as artifact in the manual procedure was estimated for each residual r_j^2 with a logistic regression function:

$$P(y_j = 1 | r_j^2) = \frac{e^{\beta_0 + \beta_1 r_j^2}}{1 + e^{\beta_0 + \beta_1 r_j^2}},$$

where β_0 and β_1 were optimized on the training sample. Based on the estimated probabilities, the log likelihood of the residuals of each parameter setting with respect to the clinical judgement of the time series was calculated. We applied this in a 10-fold cross validation procedure. The parameter setting with maximal cross validated log likelihood was selected as optimal parameter setting for the



a b

Figure 2: Results of the automated filtering procedure on a series of 200 central venous pressure using smoothing splines: (a) shows the time series, the smoothed function (in bold), and the data points considered as artifacts by the manual filtering procedure (shaded circles) and by the automated procedure (large circles); (b) shows the corresponding residuals and the classification threshold.

given smoothing technique as applied to the time series of the variable v.

Subsequently, the optimal threshold value to classify measurements as artifacts and non-artifacts was chosen on the training sample. For each squared residual r_j^2 in the training sample, we defined

$$c_j = \begin{cases} 1, & \text{if } r_j^2 \geq \tau_v, \\ 0, & \text{otherwise.} \end{cases}$$

Threshold τ_v was chosen by minimizing the classification error of c_j with respect to y_j. We assigned equal weight to the two different types of error, false positives (i.e., non-artifacts incorrectly classified as artifacts) and false negatives (i.e., artifacts incorrectly classified as non-artifacts). In case of a tie, the classification threshold was chosen conservatively (i.e., the highest threshold value). This amounts to assigning additional weight to the false positive errors to minimize the risk of classifying non-artifacts in future time series as artifacts and removing them incorrectly from the series.

Finally, after tuning of this procedure on the training sample, the procedure was applied and evaluated on the test sample; in practice, these will be new monitoring data.

3.4 Experiments and evaluation

We applied our procedure for automated filtering to the 10 ABPm, 13 CVP, and 7 HR time series. The procedure was applied to these variables using kernel smoothing, local regression, and smoothing splines as smoothing technique with the aim to compare the performance of the procedure for these different smoothing techniques. So, we performed nine experiments.

To make optimal use of the available data, we evaluated the performance of the procedure in each experiment using 10-fold cross validation. We used the clinical judgement of the measurements as obtained in the manual filtering procedure as gold 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; the number of incorrectly classified non-artifacts can be derived from the reported positive predictive value.

4 Results

In this section, we describe the results of filtering of the 30 time series by the ICU physicians. Subsequently, the results of applying the automated filtering procedure to the time series are described.

4.1 Manual filtering procedure

The four ICU physicians judged respectively 12, 57, 46, and 49 of the 2693 measurements that compose the 10 ABPm time series as 'questionable', 18, 18, 51, and 73 of the 3145 measurements in the 13 CVP time series, and 46, 58, 42, and 16 of the 2005 measurements in the 7 HR time series. In the consensus meeting, the individual judgements were harmonized to a consensus judgement, and 22 measurements (0.8%) in the ABPm time series were judged as artifacts, 22 measurements (0.7%) in the CVP time series, and 46 measurements (2.3%) in the HR time series.

Figure 1 illustrates the results of the manual filtering procedure for a mean arterial blood pressure series. In Figure 1a, the shaded circles represent data points that one or more ICU physicians considered as 'questionable'; the associated numbers correspond to the number of physicians

Table 1: Results of applying the automated filtering procedure to the ABPm, CVP, and HR time series using kernel smoothing, local regression, and smoothing splines. The performance is quantified in terms of the 10-fold cross validated sensitivity and positive predictive value (PPV) in the set of time series.

Variable (*unit*)	**Smoothing technique**	**Neighborhood size / degrees of freedom**	**Threshold**	**Sensitivity**	**PPV**
ABPm (*mmHg*)	Kernel smoothing	10	1477	13/22	13/14
	Local regression	30	1544	13/22	13/15
	Smoothing splines	30	1648	12/22	12/14
CVP (*mmHg*)	Kernel smoothing	5	123	12/22	12/15
	Local regression	10	103	13/22	13/16
	Smoothing splines	40	165	9/22	9/11
HR (*beats/min*)	Kernel smoothing	200	1727	18/46	18/23
	Local regression	200	1197	13/46	13/19
	Smoothing splines	2	1425	17/46	17/25

having that judgement. All non-shaded circles represent data points that were judged to be reliable. In Figure 1b, the same series is shown, now after consensus was reached among the four physicians: the shaded circles represent data points that were considered as artifacts.

4.2 Automated filtering procedure

We applied the automated filtering procedure to the ABPm, CVD and HR time series. Figure 2 illustrates the application of the procedure to a central venous pressure series using smoothing splines. Figure 2a represents the time series, and the data points considered as artifacts by the manual filtering procedure (shaded circles), and by the automated procedure (large circles); the bold line represents the smoothed curve. The corresponding squared residuals are shown in Figure 2b. Based on these residuals, we would be able to perfectly classify all measurements in this series, e.g., by using a classification threshold of 100. However, in the procedure, the threshold was optimized on all CVP time series (horizontal line). Using this threshold, one artifact in this series was incorrectly classified as non-artifact.

Table 1 lists the results of the nine experiments. For each experiment, the optimized parameter settings are reported, i.e., the neighborhood size for kernel smoothing and local regression, the number of degrees of freedom for smoothing splines, and the classification threshold. Furthermore, the cross validated sensitivity and positive predictive value are listed. So, 9 of the 22 artifacts in all CVP time series were correctly classified using smoothing splines. In total, 11 measurements were classified as artifact in this experiment, of which 2 measurement incorrectly.

5 Discussion and conclusions

Effective data filtering of monitoring data is an important requirement for reliable use of computerized medical assistants. This is especially the case for applications that focus on extreme values of the data. Extreme values are, for instance, used in several scoring models that quantify the severity of illness of intensive care patients (e.g., the lowest heart frequency in the first 24 hours of ICU stay) [Knaus *et al.*, 1991; Le Gall *et al.*, 1993]. With the introduction of ICU information systems, the task of extracting these items from monitoring data can be performed automatically. It has been shown, however, that this may result in more extreme values, resulting in higher severity scores [Bosman *et al.*, 1998]. The application of these techniques is therefore highly dependent on preparatory artifact removal.

In this study, we investigate to what extent basic smoothing techniques can be used to imitate filtering of monitoring data by experienced clinicians, without taking account of context information and domain knowledge. The first results show that about half of the artifacts in the time series were correctly classified by our procedure (i.e., at most 13 of the 22 artifacts in the ABPm and CVP time series, and at most 22 of the 46 artifacts in the HR series). As the measurements were classified by their absolute deviance from the smoothed time series, these were data points with largest deviances. The number of measurements that were incorrectly classified as artifacts was found to be low, especially for the ABPm and CVP time series. The three different smoothing techniques were found to perform roughly equally well. In the procedure, their parameter setting was optimized per monitoring variable. It turned out that more smoothing was required for the HR series (larger neighborhood sizes and lower number of degrees of freedom) than for the ABPm and CVP series.

This study is related to the work of M. Imhoff *et al.* [Imhoff *et al.*, 1998] in which time series of monitoring data were analyzed with low-order autoregressive models and phase space models to detect outliers, level changes and trends. They evaluated the performance of their approach with clinical judgement of the time series, and both types of model were found to be able to identify all outliers. Compared to our study, a large number (134) of time series were analyzed. These series were judged by one senior ICU physician, for whom a high intrarater reliability was found. However, the interrater reliability among physicians in judging monitoring data is relatively low, as appeared in our study. Therefore, the evaluation in this study is more subjective than in our study. Their approach was found to be useful in retrospective analysis of monitoring data, but not for routine bedside clinical use, as the applied models are semiautomatic and require that the user has an explicit knowledge of statistics. Our procedure for automated fil-

tering using simple smoothing techniques does not suffer from these disadvantages, and is expected to be easily implemented in an ICU information system.

In the patient monitoring and therapy planning system VIEVENT [Miksch *et al.*, 1996], an extensive procedure for data validation is implemented to detect and repair erroneous measurements [Horn *et al.*, 1997]. The procedure employs data filtering in a hybrid approach using (clinical) knowledge of the (course of the) variables and their interrelation in addition to statistical methods. Manually filtered time series are sufficient for tuning our procedure, and there is no further necessity to make knowledge of the variables explicit.

A general problem in studies on artifact detection is that the concept of 'artifact' is vague and hard to define. This explains the low interrater reliability that we found among the physicians' judgements and has motivated harmonization of the judgements through a consensus meeting. Yet the resulting consensus judgement is at best an intersubjective, and not an objective, standard. We note however that (inter)subjective standards may be equally well reproducible by automated filtering techniques, and that is precisely what this study aims at.

One may suggest that the procedure will improve by applying the smoothing techniques in a leave-one-out design to avoid that data artifacts attract the smoothing function. In a leave-one out design, for each measurement in the time series, the smoothing technique is applied to the time series of which this measurement is excluded from, and the resulting function was subsequently applied to obtain a smoothed estimate of the particular measurement. We also performed our experiments in a leave-one-out design, but did not found an improvement of the performance of the filtering procedure, though.

In our procedure, the stability of the time series was not explicitly taken into account. The variance over time contains important additional information for data filtering, though, as physicians generally judge sudden small changes in stable parts of the series as artifacts with more certainty, than sudden larger changes in unstable patterns. Therefore, in the future part of this study, we will derive confidence intervals around the time series from bootstrap samples, and perform filtering based on these intervals.

The time series that were used in this study were selected for their relatively rough course, and stable time series were underrepresented. Because of the aforementioned feature of the current procedure, the sensitivity of the procedure may therefore be overestimated in our experiments. To determine the performance of the procedure for automated filtering of monitoring data in general, the procedure should be applied to randomly selected time series.

To conclude, the procedure for automated data filtering using basic smoothing techniques and based on the resulting residuals turned out to be highly predictive, but moderately sensitive, and will therefore be refined in the future part of this study.

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

[Bosman *et al.*, 1998] R. J. Bosman, H. M. Oudemans van Straaten, and D. F. Zandstra. The use of intensive care information systems alters outcome prediction. *Intensive Care Medicine*, 24:953–958, 1998.

[Charbonnier, 2005] S. Charbonnier. On line extraction of temporal episodes from ICU high-frequency data: a visual support for signal interpretation. *Computer Methods and Programs in Biomedicine*, 78:115–132, 2005.

[Cleveland and Loader, 1996] W. S. Cleveland and C. Loader. Smoothing by local regression: Principles and methods. In W. Haerdle and M. G. Schimek, editors, *Statistical Theory and Computational Aspects of Smoothing*, pages 10–49. Springer, New York, 1996.

[Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, Berlin, 2001.

[Horn *et al.*, 1997] W. Horn, S. Miksch, G. Egghart, C. Popow, and F. Paky. Effective data validation of high-frequency data: time-point-, time-interval-, and trend-based methods. *Computer in Biology and Medicine, Special Issue: Time-Oriented Systems in Medicine*, 27:389–409, 1997.

[Imhoff *et al.*, 1998] M. Imhoff, M. Bauer, U. Gather, and D. Löhlein. Statistical pattern detection in univariate time series of intensive care on-line monitoring data. *Intensive Care Medicine*, 24:1305–1314, 1998.

[Knaus *et al.*, 1991] W. A. Knaus, D. P. Wagner, E. A. Draper, J. E. Zimmerman, M. Berger, 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(6):1619–36, 1991.

[Le Gall *et al.*, 1993] J. Le Gall, S. Lemeshow, and F. Saulnier. A new Simplified Acute Physiology Score (SAPS II) based on a European/North American multicenter study. *Journal of the American Medical Association*, 270:2957–2963, 1993.

[Michel *et al.*, 2003] A. Michel, A. Junger, M. Benson, D. G. Brammen, G. Hempelmann, J. Dudeck, and K. Marquardt. A data model for managing drug therapy within a patient data management system for intensive care units. *Computer Methods and Programs in Biomedicine*, 70:71–79, 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. *Artificial Intelligence in Medicine*, 8:543–576, 1996.

[Wand and Jones, 1995] M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall, London, 1995.

Visual Development of Temporal Patterns for Medical Data Abstraction

Norman Brümmer[*], **Joachim Baumeister**[*], **Daniel Riewenherm**[†], **Frank Puppe**[*], **Jens Broscheit**[†]
[*]Department of Computer Science, University of Wuerzburg, Am Hubland, 97074 Wuerzburg
{bruemmer,baumeister,puppe}@informatik.uni-wuerzburg.de
[†]Department of Anesthesiology, University of Wuerzburg, Josef-Schneider-Str. 2, 97080 Wuerzburg
{riewenherm_d,broscheit_j}@klinik.uni-wuerzburg.de

Abstract

In this paper we present a visual representation of temporal patterns in abstractions of numerical and timestamped data. We provide a curve-like acquisition tool which supports domain specialists to develop and refine temporal knowledge in an intuitive and effective manner. The resulting patterns can be used to detect artifacts as well as more complex phenomena, e.g., in order to derive intelligent alarms.

1 Introduction

The temporal development of numerical data and its interpretation, respectively, is of prime importance when monitoring patients in the medical domain, e.g., during surgeries or in the context of an ICU. Here, the automatic abstraction and interpretation of these continuously received parameter values can support the medical staff, e.g., anesthetists, with the tracking of the patient's status.

Furthermore, the interpretation of parameter values and their development is often difficult because they are superimposed by artifacts, e.g., an accidently dropped pulse sensor. In consequence, the validation of received parameter values preceding the actual interpretation is a crucial issue.

In this paper we present an approach for a visual representation of temporal abstraction and validation knowledge allowing for an intuitive and precise formalization. The applied visual patterns were adopted from knowledge engineering interviews taken with the domain specialists and were refined in order to allow for a formal and precise interpretation of the modeled temporal knowledge.

The visually acquired patterns will be translated to a textual representation in order to be integrated into a rule-based formalism. This enables a combination of temporal patterns and non-temporal rule conditions including conjunctions, such as *and*, *or* and *not* expressions.

The context of our work is an intelligent monitoring and alarm system to be used during surgeries or in IC units, there supporting the work of anesthetists.

The paper focuses on the development of high–level abstractions, i.e., deriving meaningful alarms or artifacts, although the handling of basic abstractions derived from raw data streams is also an important task.

The derivation of intelligent alarms is two-folded in our system: In the first step we try to detect defined states of artifacts within previously abstracted data streams and annotate the data with possibly found artifacts (thus enabling for a high-level data validation). In the second step we investigate the annotated data streams for defined alarm states, e.g. *insufficient anesthesia* and *hypovolemia*.

2 A Visual Representation for High-Level Abstractions

The manual definition of complex patterns of the particular parameter changes over time is a difficult and error–prone task. For this reason we introduce an intuitive and visual representation for describing such patterns, i.e., *Abstract Temporal Curves (ATC)*, which can be interpreted as conditions for temporal rules for deriving high-level abstractions, i.e. artifacts and alarms. The representation offers a curve-like description of the temporal behavior of parameters, thus describing certain phenomena.

2.1 Abstract Temporal Curves

In the following, we introduce simple graphical elements that enable a description of basic events occurring in abstract parameter courses. Thereafter, we define temporal constraints that can be applied to events in order to describe the temporal behavior. A more complex temporal pattern is described by a set of events, attached constraints and a maximum duration restricting the entire pattern.

The modeling basis of an ATC contains layers for each involved parameter (Figure 1). Horizontal lines denote the corresponding abstract parameter values. There exist two basic elements that can be combined in different ways in order to describe the possible events.

Edges are horizontal lines describing a persistent value the specified parameter may take. Parallel edges of the same parameter define alternative and possible values for the parameter.

Nodes are markers placed on edges at arbitrary positions. They define changes of parameter values and thus basically declare temporal constraints.

Changes in the specified parameter behavior must be separated by nodes. Additionally, further nodes can be placed at any position, as it has been done with the first defined node in Figure 1. Here the decrease of the abstracted parameter value *AP* (arterial blood pressure) is specified

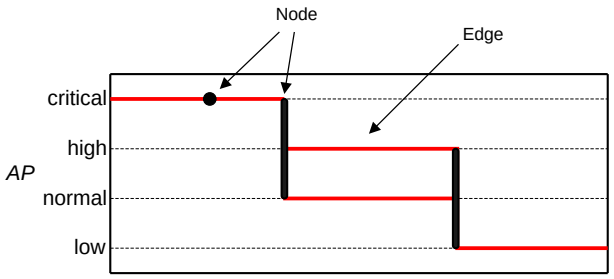


Figure 1: Edges and nodes in a parameter sequence in an ATC. Nodes can be defined at arbitrary positions on edges. They denote events expressing value changes and value persistence.

starting with value *critical* then falling to either *high* or *normal* and finally decreasing to the value *low*.

We extend the notation by *temporal constraints* between the nodes. A temporal constraint consists of a pair of nodes connected by a period. It denotes that the enclosed events need to occur within the specified time range. There are three alternatives to connect nodes by temporal constraints; Figure 2 depicts different types of constraints: A *sequence (c)* defines a time span for a certain event flow, occurring on a single parameter. It is required to occur within the given period. A second alternative for a definition of temporal constraints are *intervals (a)*. An interval connects two nodes of different parameter courses, which means that the corresponding events need to occur in the given time span. With the third alternative, i.e., the *point-interval (b)*, we directly connect nodes in order to express that the corresponding events are required to occur simultaneously. The

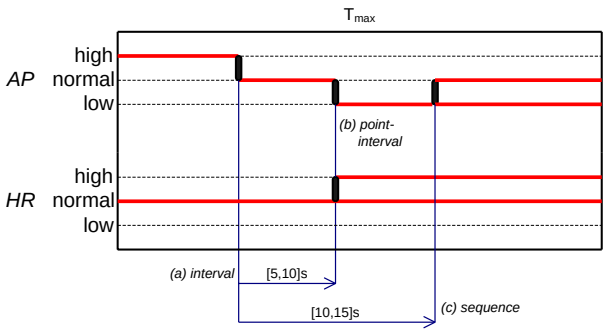


Figure 2: An ATC with temporal constraints defined between some nodes on the courses of the parameters *AP* (arterial blood pressure) and *HR* (heart rate).

additional global constraint T_{max} on top of the pattern description implies that the duration of the entire scenario is restricted to the time-interval $(0, T_{max}]$.

2.2 Translation of ATCs to Temporal Rules

The presented visual notation allows for an intuitive description of temporal events. These events will be compiled to a textual representation consisting of temporal events and expressions. The resulting temporal patterns can be combined with non-temporal rule conditions. Thus, they can be integrated into a rule-based formalism. For example, the rule framework of *d3web* [Baumeister, 2004] allows for a definition of complex rule conditions built up by non-terminal constructs, e.g. *and*, *or* and *not*. This enables the intuitive definiton of complex temporal patterns.

3 Results and Discussion

The presented work was implemented in the context of a medical project aiming for an intelligent detection of artifacts and alarms during surgeries. Two medical experts were involved with the formalization and refinement of the temporal knowledge including a variety of typical artifact and alarms. It turned out that the domain specialists got familiar very quickly with the meaning of the visual knowledge representation. This was not surprising since the representation was adopted from the observations made during knowledge acquisition interviews conducted with the domain specialists. In fact, similar curves were drawn on paper during the interview sessions to explain the particular events. Consequently, almost no training phase was required and the initial model was discussed and implemented in about 5 minutes for each pattern. However, the main effort consisted in testing and refining the collected patterns. The most complex task was the appropriate definition and refinement of temporal constraints included in the patterns. These adaptations required frequent testing cycles. For simple patterns, e.g., mostly artifacts, the refinement phase took about 10 minutes, whereas more complex patterns required more than 30 minutes for adaptation and testing.

In the literature, representations for high-level abstractions are mostly graph-based. A related approach to our visual representation can be found in [Chittaro and Combi, 2003]. There, three alternative visual vocabularies representing intervals and their relations (as subset of Allen-Relations), based on objects from the physical world, are presented. But, explicit temporal periods (e.g. $[10s, 30s]$) can not be modeled, as it is possible in our approach.

In the future we are planning to integrate a fuzzy definition of abstraction thresholds and temporal expressions, since an exact definition turned out to be one of the most difficult tasks. Visualizations of temporal uncertainty has been accomplished e.g. in [Kosara and Miksch, 1999]. Furthermore, we consider the semi-automation of the refinement process of the patterns by adapting discovery algorithms for this task.

References

[Baumeister, 2004] Joachim Baumeister. *Agile Development of Diagnostic Knowledge Systems*. AKA Verlag, DISKI 284, 2004.

[Chittaro and Combi, 2003] Luca Chittaro and Carlo Combi. Visualizing Queries on Databases of Temporal Histories: New Metaphors and Their Evaluation. *Data and Knowledge Engineering*, 44(2):239 – 264, 2003.

[Kosara and Miksch, 1999] Robert Kosara and Silvia Miksch. Visualization Techniques for Time-Oriented, Skeletal Plans in Medical Therapy Planning. *Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99)*, pages 291–300, 1999.

Invited presentation

Intelligent Data Analysis for Biomedicine: What have we learned?

Xiaohui Liu
Intelligent Data Analysis Group
School of Information Systems, Computing and Mathematics
Brunel University, London
www.ida-research.net

Intelligent Data Analysis (IDA) is the interdisciplinary study concerned with the effective analysis of data. The ever increasing and variety of data in biomedicine have provided one of the most fertile grounds for testing existing analysis techniques and providing great motivations for investigating new IDA methods. For over ten years, the IDA group at Brunel have been working on the interface between artificial intelligence, dynamic systems, image and signal processing, pattern recognition and statistics, and have been applying this research to challenging problems in biology and medicine, e.g. microarray data analysis, managing glaucoma and muscular dystrophy. In this talk, I shall introduce some of our work in this area, particularly those involving temporal data analysis. Key research issues covered will include data quality control, multivariate time series analysis, and biomedical evaluations. Finally some useful lessons we have learnt will be discussed.

Paper session: *Bioinformatics*

Rule-based clustering for gene regulation pattern discovery

Tomaz Curk,[a] Uros Petrovic,[b] Gad Shaulsky,[c] Blaz Zupan[a,c]
a) University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia
b) J. Stefan Institute, Department of Biochemistry and Molecular Biology, Ljubljana, Slovenia
c) Baylor College of Medicine, Department of Molecular and Human Genetics, Houston, TX
tomaz.curk@fri.uni-lj.si, blaz.zupan@fri.uni-lj.si, gadi@bcm.tmc.edu, uros.petrovic@ijs.si

Abstract

Sequence and structure of gene regulatory promoter regions determine the genetic response programs of cells according to their internal state and environment. Computational inference of the relation between structure of promoter regions and gene expression can help us to understand the underlying genetic programs and can greatly assist in experiment planning. It most often relies on a large data base of regulatory elements (*i.e.* putative or known transcription factor binding sites) and then infers rules that relate promoter structure and gene expression. The principal obstacle it faces is combinatorial explosion of rules to test. We developed a rule-based clustering method that uses gene expression distance to guide rule inference and searches only the most promising part of the vast and expressively rich rule-space. We also developed a set of effective visualizations to present and explore the shared structural promoter features of discovered gene clusters. Cross-validation confirms the ability of the proposed rule-based clustering method to find rules with good predictive power.

1 Introduction

Regulation of gene expression is a complex key mechanism in the biology of eukaryotic cells. Cells carry their function and respond to the environment by an orchestration of signaling molecules and transcription factors that influence gene expression. Resulting products regulate expression of other genes thus forming diverse set of regulatory pathways. To better understand gene functions and gene interactions we need to discover and analyze the programs of gene regulation. Computational analysis of gene regulatory regions can greatly speed-up and to a certain extent automate the normally tedious discovery process performed by geneticists.

Gene's regulatory (promoter) region is defined as a stretch of DNA, normally located upstream of the gene's coding region. Transcription factors are special proteins that can bind to sequence-specific binding sites in regulatory regions, and by doing so inhibit or excite gene expression of their target genes. Regulation by binding of transcription factors is just one of the many mechanisms of gene expression regulation. Expression is also determined by chromatin structure, epigenetic effects, post-transcriptional, translational, post-translational and other forms of regulation [Wasserman and Sandelin, 2004]. Because there is a lack of these kinds of data, most current studies focus on inference of relations between gene regulatory content and their expression as measured using microarray technology.

Determining the regulatory region and putative binding sites are the first crucial steps in such analyses. Regulatory regions differ from coding regions in nucleotide and codon frequency. This fact is successfully used by many prediction algorithms [Bajic *et al.*, 2004]. Genome promoter sequences are readily available for download for most organisms (for yeast see www.yeastgenome.org).

The next more important and well studied step is to determine transcription factors' putative binding sites in promoter regions. These are 4 to 20 nucleotide long DNA sequences [Wasserman and Sandelin, 2004] which are highly conserved (with low sequence variation) in the promoter regions of regulated genes. A matrix representation of binding sites is normally used in computational analysis. The matrix defines the frequency of the four nucleotides (A, T, G, C) at each position in the binding site (see Table 1 and Figure 2 for example). The binding site is often also presented as a single consensus line (see Table 1), or graphically as a logo (see Figure 2). Experimentally confirmed and computationally inferred putative binding sites are now available in data bases such the TRANSFAC data base [Wingender *et al.*, 1996]. When analyzing genes with unknown regulators, one can find candidate binding sites using local sequence alignment programs such as the MEME program [Bailey and Elkan, 1994] that can identify short, frequent sequences. A detailed description and evaluation of such tools is presented in [Tompa *et al.*, 2005].

Most of contemporary methods to relate gene structure and expression start with gene expression clustering. Next, they determine cluster-specific binding sites. The success of such approach is heavily dependent on number (usually a parameter to clustering method) and composition of gene clusters. A slight change in initial conditions or parameter values for clustering can lead to different groupings with substantially different sets of binding sites. Another problem with such approach is its inability to discover overlapping subgroups; it is biologically known

position	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
A	0	0	0	0.2	0.1	0	0	0	0	0.1	0.6	0.1
C	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0.5	0	0.4	0.4	0.1	0	0.8	0.4	0.8	0	0.6
T	1.0	0.5	1.0	0.4	0.5	0.9	1.0	0.2	0.6	0.1	0.4	0.3
single line consensus	T	K	T	K	K	T	T	G	K	G	A	K

Table 1. Matrix presentation of an example putative binding site. A single line consensus sequence presentation, giving the most frequent nucleotide at each position, is also used (coded with standard FASTA codes, *e.g.* K indicates G or T).

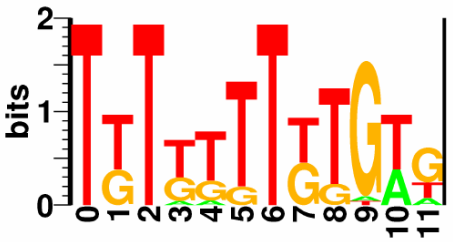


Figure 2. Logo representation of the putative binding site from Table 1. Information of nucleotides at each position is given.

that same gene can respond in many different ways and perform various functions.

An alternative to above is to start with information about binding sites and search for descriptions shared between similarly expressed genes. An example is the approach proposed by [Chiang *et al.*, 2001] where for each binding site and corresponding set of genes their method calculates the correlation of gene expression profiles. This correlation is then statistically compared with the one obtained from the same-sized randomly drawn set of genes. If the observed difference is statistically significant, the method reports on the rule for the particular binding site. The method fails to model combinations of two, three or more putative binding sites. This could be regarded as a major deficiency as it is biologically known that regulation of gene expression can be highly combinatorial and requires the coordinated presence of many bound transcription factors. More advanced methods try to infer rules that describe the structure of regulatory regions with more than one, but rarely more than two putative binding sites [Pilpel et al., 2001; Beer and Tavazoie, 2004]. The principal bottle-neck is the complexity of exhaustive combinatorial search that these methods employ, which quickly becomes prohibitive when exploring combinations of binding sites. For example, the number of all possible combinations of three binding sites, from a base of thousand binding sites available for modeling, quickly grows into hundreds of millions. Transcription is also affected by absolute or relative orientation and distance between binding sites and other landmarks in the promoter region (*i.e.* the translation start ATG), making exhaustive search that would include such models unpractical.

To overcome these limitations we have developed a heuristic rule search method that is able to efficiently identify complex structural descriptions of gene regulatory regions. The proposed rule-based clustering method is guided by the information on the similarity of gene expression and explores only the most promising (coherent) subgroups of genes with similar regulatory content.

2 Rule-based clustering method

We devised a rule-based clustering approach with the goal to find potentially complex rules that describe the shared regulatory structure of genes with similar expression profiles. Similarity in gene expression is assessed using Pearson correlation, but other distance measures could be used instead. Rules are of the form "IF *structure* THEN *expression profile*", where *structure* is an assertion over the binding sites in the gene promoter sequence and uses a description language defined below, and *expression profile* is a average profile of genes that match the *structure*.

2.1 Descriptive language and rule search

We have defined a rich descriptive language which, besides being able to describe presence of a binding site, can be used to define conditions on the distance of putative binding sites from transcription and translation start site (ATG) and other landmarks, the distance between putative binding sites and their relative and absolute orientation relative to a given reference point (see Figure 3).

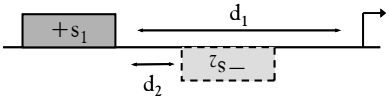


Figure 3. The descriptive language for gene's structure can include conditions on distance (d_2) between two binding sites (s_1 and s_2), distance from ATG (d_1) and orientation of binding sites in the sense ($+s_1$) or non-sense reading direction ($-s_2$).

We based our heuristic rule search method on the approach of clustering trees developed by [Blockeel *et al.*, 1998]. We developed a general rule search method that is guided by the information on example distance (gene distance in our applications) and only the most promising parts of the vast rule-space are searched. Every next step in the group (or cluster) discovery process is selected based on the correlation of gene expression in currently discovered groups. Further refinements of rules are performed on only those rules describing the most promising groups.

The method requires a "target" set of genes for which we want to find rules describing their promoter regions and cluster them into subgroups. The target set can be the full genome or can be a set of differentially expressed genes in one or many experimental conditions (mutation of a gene, outside chemical, temperature or other influence) measured with DNA microarrays. The algorithm starts with the set of all genes, which is represented by the conditional part of the initial rule "True." The algorithm then tries to refine the current rule set B by adding conditional parts. For example, the conditional part of rule

"M_1" requiring only the presence of binding site M_1 can be refined into "M_1+" requiring the orientation of M_1 to be in sense direction. The initial condition "M_1" can be refined to require binding site M_1 to be at a distance −100 to −80 nucleotides relative to ATG, which is stated by the rule's condition "M_1@−100..−80(ref:ATG)." Another binding site (*e.g.* M_2) can be used as a reference point, which we write as "M_1@−100..−80(ref:M_2)." The initial rule can also be refined into "M_1 and M_2" requiring both binding sites to be present in the promoter region.

Every refined rule has to cover fewer genes than the original but at least N target genes (parameter N is set by user). If not, it is completely discarded from further consideration. Also, the similarity in gene expression within the newly refined group must significantly increase compared to the similarity in the original group. The significance in decrease of variance is tested using F-test. Rules that pass this test are added into set B and used for further refinements (size of set B is limited to L best rules, parameter L set by user. Set B is usually referred as "beam".) Otherwise the rule is scored based on the group average intra-distance, which must be lower than parameter D (parameter set by user), and moved into the final set of rules R where only K best rules are kept and returned as result of the rule-based clustering algorithm (parameter K set by user). For the basic step of the search algorithm see Figure 4, for the algorithm see Algorithm 5.

Note that because the algorithm starts with the entire genome, the discovered rules can cover genes outside the target set. The method can be applied to search for genes that were initially left out of a target set but should have been included based on their regulatory content and gene expression.

The proposed rule-inference method differs from classic rule-coverage algorithms (*e.g.* CN2 described by [Clark and Niblett, 1989]) because it allows the discovery of overlapping groups of genes. The basic CN2 algorithm has a relatively small beam (size of beam set B in our algorithm is bigger) and iteratively removes examples (genes in our case) that can be described by the best rule discovered in current iteration. The procedure is then repeated on the reduced set until no examples are left to cover. Our proposed method uses a larger beam and searches for rules until refinements are possible. No actual gene coverage is considered in these steps. After the search is completed, all discovered rules are sorted by their score. We then traverse the ordered list and check the average cumulative gene group coverage of each rule. If average cumulative coverage is less than parameter M (set by user) then the rule is selected and the genes' cumulative coverage updated accordingly, otherwise the rule is discarded. This procedure selects the final set of best rules that are presented to the user.

Exhaustive search of even relatively simple rules can quickly grow into a prohibitively hard problem due to combinatorial explosion. The main distinctive feature of our method is its ability to efficiently derive rules describing a higher combinatorial regulation involving three or more binding sites by starting from a base set of thousands putative binding sites. We give an example of the number of rules search by our heuristic method and compare it with the number of all possible combinations that would be inspected by exhaustive search.

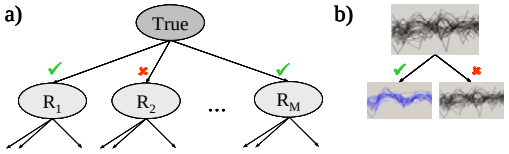


Figure 4. a) Rule refinement is the basic step of the search algorithm. b) The intra-group gene expression similarity of genes covered by the refined rule (bottom nodes) must increase significantly compared to the group described by the original rule (top node). Different refinements of same rule lead to differently homogeneous groups (check mark indicates a significant increase in the group intra-similarity, cross represents a not significant increase compared to expression of original group).

```
1 beam B is a set of maximum L best rules for
  further refinement: B = {True}
2 R = {} is a set of K best discovered rules
3 WHILE B ≠ {}
4   take best rule Rb from set B
5   FOR EACH k IN 1..number of binding sites
6     refine Rb by using binding site Mk,
      create new rule Rn
7     IF rule Rn acceptable AND significant
      increase between Rb in Rn THEN
8       Add Rn into B (keep best L rules in B).
9 add rule Rb into R if among K best
10return K best discovered rules in set R
```

Algorithm 5. The search algorithm. Rule R_b is refined into R_n (line 6) by adding conditions on presence, orientation and distance of a new binding site (M_k) relative to binding sites already described in other terms of rule R_b.

2.2. Visualization of results

The rich descriptive language and the method's ability to discover overlapping gene groups can result in a large number of discovered rules. We implemented three graph-based visualizations that facilitate a better insight into the common structural features of discovered rules and gene groups (see Figure 6). Gene network graph is the simplest way to visualize the discovered groups (Figure 6a). Nodes represent genes and we connect two nodes if the two genes are covered by same rule. Presence of many overlapping groups can quickly render this visualization saturated. The next level of abstraction is group graph (Figure 6b). Nodes represent groups of genes (that can be described by one or more rules) and we connect two nodes (groups) if they share an arbitrary number of genes (parameter set by user). This visualization is useful for exploration of regulatory or functional overlaps in discovered subgroups of genes. By varying the threshold one can observe how the initial grouping of genes breaks into less connected subgroups. The last level of abstraction is motif graph (Figure 6c). Nodes represent terms (parts of rules requiring specific characteristics for a binding site; binding site is also called "motif," hence the graph name) and two nodes are connected if they appear in same rule. This can be used to identify common and rule-specific binding sites appearing in discovered rules indicating potential general and group-specific regulators respectively.

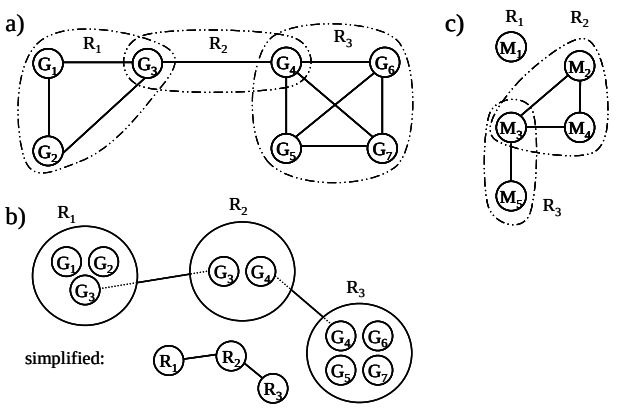


Figure 6. Three visualizations for presenting results. Conditional parts for three simple rules are visualized in this example. Rule "$R_1 = M_1$" requires the presence of binding site M_1. Rule "$R_2 = M_2$ and M_3+ and M_4" requires the presence of three binding sites (M_3 must be in sense direction) and rule "$R_3 = M_3$ and M_5@-200..-180(ref:ATG)" requires two sites one of which must be relative to ATG. Rules' gene coverage is shown in graphs. a) Gene network graph. b) Group graph. c) Motif graph.

3 Case study and experimental validation

We tested the proposed rule-based clustering method on the data set from a microarray transcription profiling study where budding yeast *S. cerevisiae* cells were induced to proliferate peroxisomes – organelles found in most organisms and cell types that compartmentalize several oxidative reactions – as a result of cell's regulated response to absence of glucose or glycerol and exposure to fatty acid oleate as the sole carbon source [Smith *et al.*, 2002]. Each gene in the data set is described with a transcription profile that consists of six microarray measurements from oleate induction time course and two measurements in "oleate vs. glucose" and "glucose vs. glycerol" growth conditions. In total, we used eight microarray measurements of gene expression to calculate the distance between genes. We defined the distance function to be 1.0-Pearson correlation in gene expression for the given gene pair.

For the target group we selected a set of 224 genes that were identified in the study to have similar expression profiles to those of genes involved in peroxisome biogenesis and peroxisome function. The goal of our analysis was to further divide the target group into smaller subgroups of genes with common elements in promoter structure and possibly identify genes that were inadvertently left out but should have been included in the target group based on their expression and promoter structure.

Our analysis included information on 2,135 putative binding sites that were identified using a local alignment software tool MEME [Bailey and Elkan, 1994]. We then searched for presence of these binding sites in one thousand bases (1Kb) long promoter regions which were taken upstream from the translation start site (ATG) for ~6,700 yeast genes. The search identified ~302,000 matches (*i.e.* occurrences) of putative binding sites that were then used to infer rules by the rule-based clustering algorithm. The algorithm searched for rules describing groups with at least six target genes (N=5) and average group intra-correlation above 0.5 (*i.e.* the maximum allowed intra-distance was set to D=1.0–0.5=0.5). We limited the rule search beam to one thousand best rules for further refinements (L=1000). Distances between binding sites were rounded to increments of 40 bases; the maximum possible distance of 2Kb (given the promoter length, relative distances can be in range from -1Kb to +1Kb) was thus reduced to 50 (=2000b/40b) different values. This largely reduced the number of possible subintervals that needed to be considered when inferring rules.

The search resulted in 41 rules describing and dividing 114 target genes (out of total 224 target genes) into 37 subgroups (see Figure 8). No rule could be found for the remaining 110 target genes. Most discovered gene groups are composed of five genes with high pair-wise intra-group correlation (all are above 0.927). Many genes are shared (overlap) between the 37 discovered groups resulting in six major groups visible in Figures 8 and 9. Seven genes outside the target group were also identified by the method (marked in red in Figure 8). For example, the smallest eight gene group in the top-left corner in Figure 8 includes two outside genes (INP53 and YIL168W - also named *SDL1*). Gene ontology analysis shows that INP53 is involved together with two target genes (ATP3 and VHS1) in the biological process *phosphate metabolism*. Gene SDL1 is annotated to function together with the group's target gene LYS14 in the biological process *amino acid metabolism* and other similar parent GO terms (results not shown). These examples confirm the method's ability to identify functionally related genes that were not initially included in the target group.

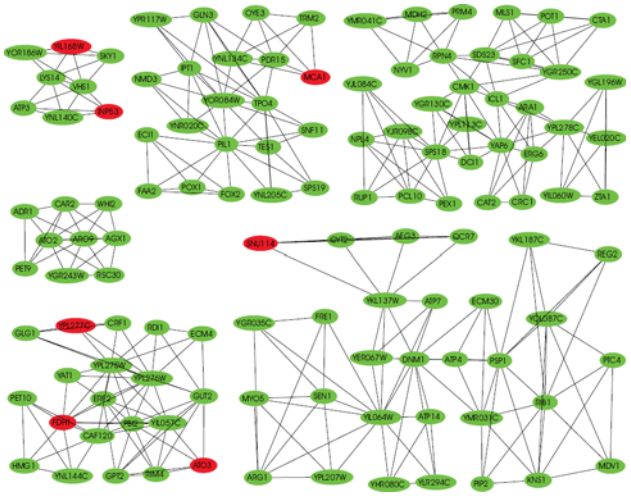


Figure 8. Gene network inferred on the peroxisome data set where 37 different subgroups were discovered. Genes are clustered in six major groups. Target gene nodes are colored green. Nodes with genes originally not included in the target group are colored red.

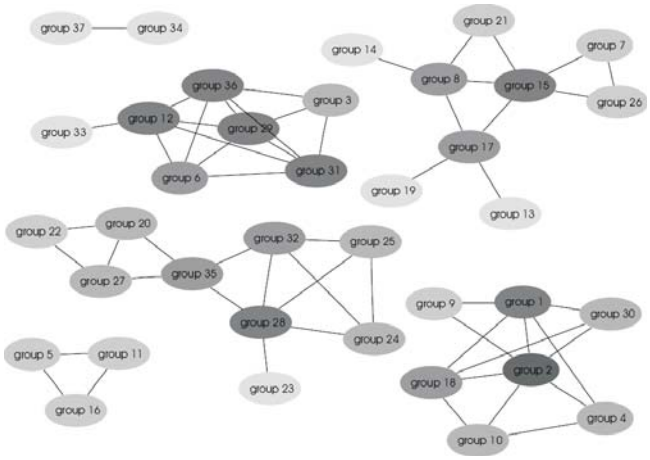


Figure 9. Group graph of discovered 37 subgroups that cluster 114 target and 7 outside genes into six major gene groups.

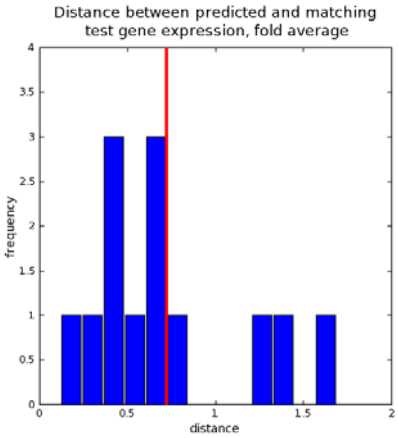


Figure 10. Distribution of distance in predicted and actual gene expression for matching rules. Red vertical line marks average distance 0.72 (average intra-correlation is 0.28).

Majority of discovered rules include conditions that are composed of three terms, each term describing a putative binding site's orientation and distance relative to ATG or binding sites included in the rule. An exhaustive search for all possible rules composed of three binding sites with defined orientation (three possible values: positive, negative, no preference) and distance (50 different values) would require checking a relatively huge number of rules:

$$\binom{2135 \cdot 3}{3} \cdot 50^3 \approx 5.47 \cdot 10^{15}$$

Our method checked $2.11 \cdot 10^9$ of the most promising rules, or less than 0.00004% of the entire three-part rule space. The search took 40 minutes on a Pentium 4, 3.4 GHz workstation.

3.1 Cross-validation of rule-based clustering

To asses the predictive ability of the inferred rules we applied five-fold cross-validation using the same data and parameter values as described in the section above. The data was randomly divided into five folds. The split was stratified, *i.e.* each fold contained one fifth of the 224 target genes and one fifth of the remaining ~6500 non-target genes. In each step of the cross-validation procedure the training part of the data was used to infer rules and identify clusters of genes. Discovered rules were then tested on genes from the test set. If a rule matched the promoter region of a test gene then the average distance (in expression) between the test gene and all training genes covered by the rule was calculated. These distances are plotted in the histogram in Figure 10. The achieved average distance is 0.72 (the average correlation is 1.0-0.72 = 0.28) which is a good indication of a good predictive quality of inferred rules.

4 Conclusion

Experimental results show the ability of the proposed rule-based clustering method to efficiently identify groups of similarly expressed genes with similar structure of regulatory region. In contrast with other contemporary methods that mainly use information on presence of binding sites in promoter region, the principal novelty of our approach is the use of a rich language to describe the structure of regulatory elements and a heuristic approach to find associated rules. To summarize the findings of our analysis we have also implemented three different visualizations that can help in understanding and biological interpretation. We believe that the main application of our method is the search for additional evidence that genes from theoretically or experimentally defined group actually share some common regulatory mechanisms or regulators. The biologist can then gain insight by looking at the presented evidence and can better decide which inferred hypotheses to test in the lab.

We experimentally confirmed the ability of the proposed method to infer rules that describe a complex regulatory structure and can reliably predict gene expression from regulatory content. We are currently implementing a web-based application allowing biologists to easily analyze their own data using the proposed rule-based clustering method.

Acknowledgments

This work was supported in part by Program and Project grants from the Slovenian Research Agency and by a grant from the National Institute of Child Health and Human Development, P01 HD39691.

References

[Wasserman and Sandelin, 2004] Wasserman WW, Sandelin A. *Applied bioinformatics for identification of regulatory elements*. Nature Reviews Genetics, 5:276-87, 2004.

[Bajic *et al.*, 2004] Bajic VB, Tan SL, Suzuki Y, Sugano S. *Promoter prediction analysis on the whole human genome*. Nature Biotechnology, 22:1467-73, 2004.

[Wingender *et al.*, 1996] Wingender E, Dietze P, Karas H, Knuppel R. *TRANSFAC: a database on transcription factors and their DNA binding sites*. Nucleic Acids Research, 24(1):238-41, 1996.

[Bailey and Elkan, 1994] Bailey TL, Elkan C. *Fitting a mixture model by expectation maximization to discover motifs in biopolymers.* Proc Int Conf Intell Syst Mol Biol, 2: 28-36, 1994.

[Tompa *et al.*, 2005] Tompa M, Li N, Bailey TL *et al*. *Assessing computational tools for the discovery of transcription factor binding sites*. Nature Biotechnology, 23:137-144, 2005.

[Chiang *et al.*, 2001] Chiang DY, Brown PO, Eisen MB. *Visualizing associations between genome sequences and gene expression data using genome-mean expression profiles*. Bioinformatics, 17(s1):S49-S55, 2001.

[Pilpel et al., 2001] Pilpel Y, Sundarsanam P, Church GM. *Identifying regulatory networks by combinatorial analysis of promoter elements.* Nature Genetics, 29(2):153-9, 2001.

[Beer and Tavazoie, 2004] Beer MA, Tavazoie S. Predicting gene expression from sequence. Cell, 117:185-198, 2004.

[Blockeel *et al.*, 1998] Blockeel H, De Raedt L, Ramon J. *Top-down induction of clustering trees*. Machine Learning, Proceedings of the 15th International Conference, Morgan Kaufmann, 1998.

[Clark and Niblett, 1989] Clark P, Niblett T. *The CN2 induction algorithm*. Machine Learning, 3(4):261-83, 1989.

[Smith *et al.*, 2002] Smith JJ, Marelli M, Christmas RH, Vizeacoumar FJ, Dilworth DJ, Ideker T, Galitski T, Dimitrov K, Rachubinski RA, Aitchison JD. *Transcriptome profiling to identify genes involved in peroxisome assembly and function*. The Journal of Cell Biology, 158(2):259-71, 2002.

Population substructure determination by means of Bayesian model averaging for Clustering

Guzmán Santafé[1], **Jose A. Lozano**[1], **Pedro Larrañaga**[1] **and Eleazar Eskin**[2]
[1]Dept. of Computer Science and A.I. (University of the Basque Country, Spain)
[2]Dept. of Computer Science and Engineering (University of California, San Diego)
guzman@si.ehu.es, ja.lozano@ehu.es, pedro.larranaga@ehu.es, eeskin@cs.ucsd.edu

Abstract

Bayesian statistical methods based mainly on Markov chain models have recently proposed in the literature and provide powerful tools to analyze population substructures on the basis of molecular markers such as SNPs. The SNPs sequences are used to classify individuals into their population of origin. These SNPs sequences may contain markers which are not relevant for this clustering process and they may blur the clustering assignation. In this paper we present the use of a powerful Bayesian model averaging algorithm for clustering which includes in the learning process a kind of implicit Bayesian variable selection. Therefore the algorithm is able to deal with irrelevant variables and then it is appropriated to analyze the population substructures. On the other hand, we also develop a two-step algorithm based on mutual information that can be used to obtain, from the clustering model, the set of SNPs which are considered relevant for clustering purposes. Hence, the proposed Bayesian approach not only offers to retrieve the population substructure, but also to obtain the set of relevant SNPs to retrieve this population substructure. The algorithm has been applied to both synthetic and real datasets, and the obtained results outperform the commonly used *Structure* software.

1 Introduction

Most genetic variation among different people can be characterized by single nucleotide polymorphisms (SNPs), which are single point mutations in the nucleotide sequence that have occurred during human history and are inherited among generations. There are lots of these nucleotide mutations that exist in a few or only one individual. However, it has been estimated that there are about 7 million common SNPs (with minor allele frequency of at least 5%) among the different human populations [Botstein and Risch, 2003].

Most studies of human variation begin by sampling individuals from a certain population. This population is usually defined in terms of subjective aspects such as religion, culture or geographical location, but these aspects do not necessarily reflect underlying genetic relationships. Individuals in the same population share certain genetic information that can be used to identify them. The problem of recovering the underlying group structure from a set of individuals is know as population substructure problem and it is well-studied in population genetics [Rosenberg *et al.*, 2002]. Apart from the evolutionary perspective, the estimated group structure can provide a useful insight into many applications such as correcting for population stratification in association studies [Sillanpää *et al.*, 2001]. In addition clinical trials can suffer from false positive of reduction of power due to population substructure [Pritchard *et al.*, 2000b].

From a purely machine learning approach, population substructure problem can be seen as a clustering problem where each SNP in the DNA sequence represents a predictive variable and the population substructure, the cluster variable that remains hidden. However, this clustering problem has certain special characteristics such as the occurrence of independent random mutation across the sequence and the presence of irrelevant variables (not all the SNPs available in the sequence may be relevant to retrieve the underlying group structure). This situation can make the problem not very amenable for distance-based clustering methods. Recently, Bayesian statistical methods based on Markov chain models have shown to provide powerful tools for the analysis of genetic population structure and to assign individuals or chromosomal segments into clusters using multilocus molecular markers [Pritchard *et al.*, 2000a; Corander *et al.*, 2004]. Other recently proposed techniques also include the use of mutual information-based metrics to obtain the best population partition [O'Rourke *et al.*, 2005].

On the other hand, the special characteristics of the problem make the application of other particular clustering techniques very interesting. Specifically, the Bayesian model averaging of naive Bayes proposed in [Santafé *et al.*, 2006] is a method which is suitable to be used in the population substructure problem. This is a Bayesian clustering method based on naive Bayes model, which is a kind of Bayesian network successfully used in many other biological problems [Barash and Friedman, 2002].

The naive Bayes model assumes that all the predictive variable are conditionally independent given the cluster variable. However, the Bayesian model averaging of naive Bayes [Santafé *et al.*, 2006] accounts for model uncertainty by taking into consideration that each predictive variable

may or may not be relevant for clustering purposes. Since in population substructure problem mutations in the sequence are considered independent and not all SNPs may be relevant to reveal the underlying group structure, the naive Bayes model learned with this Bayesian model averaging algorithm is an appropriated method to tackle this problem. Moreover, the model averaging process incorporates into the learned model a kind of implicit variable selection that can be used to decide which SNPs are more relevant to cluster individuals into their populations of origin. Therefore, in the same process in which we obtain the clustering structure of the data, we are able to obtain the set of relevant SNPs for this clustering process. In contrast to other SNPs selection methods such as the ones based on supervised classification models, the proposed method is a clustering approach, which is not guided by a predefined group structure. Clustering methods are supposed to be preferred in population substructure problem because not always the population assignation is known or, in the case where it is known, there may be unknown subpopulations groups inside the know groups or some individual, from the point of view of their genetic sequence, may be closer to other population than to the population that they have been assigned a priori.

The selection of a informative subset of SNPs that contains enough information to differentiate between populations under study may be very useful not only to correct population stratification in association studies or in admixture-mapping studies [Patterson *et al.*, 2004] but also to reduce the economical cost of sequencing samples for this studies.

In this paper we propose the use of the Bayesian model averaging algorithm (EMA algorithm) introduced in [Santafé *et al.*, 2006] to tackle population substructure problem. Furthermore, we develop a two-step mutual information test that allows the use of the model obtained by the EMA algorithm to select those SNPs which are relevant to retrieve the underlying structure from the dataset.

The rest of the paper is organized as follows. Section 2 overviews the theoretical aspects of the EMA algorithm in the context of the population substructure problem. This section shows how to learn the clustering model from data as well as how to use this model to select the most relevant SNPs or variables for clustering purposes. Section 3 tests the behavior of the EMA algorithm in a toy example problem and in a real problem with SNPs data. Finally, Section 4 presents some conclusions yielded from the paper as well as future work.

2 Methodology

The target that we are aiming for is to obtain a clustering model that provides a posterior distribution among the dataset and thus allows us to cluster the instances into different partitions or populations. For this purpose we use the EMA algorithm [Santafé *et al.*, 2006] which is a Bayesian model averaging approach to learn a naive Bayes model for clustering. The method obtains, in an efficient way, a naive Bayes model as a result of a Bayesian model averaging over all the possible selective naive Bayes (see Figure 1). The method itself is an adaptation of the well-known EM algorithm [Dempster *et al.*, 1977] that allows us to extend efficient model averaging techniques for complete data [Dash and Cooper, 2004] to clustering problems. The EMA algorithm is a greedy iterative algorithm that is comprised of two steps: expectation (E step) and model averaging (MA step). This last step accounts for model uncertainty by an approximation of the averaging over all the selective naive Bayes models.

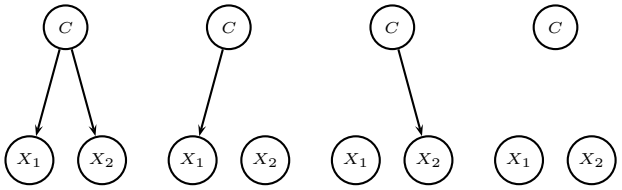


Figure 1: Selective naive Bayes structures with two predictive variables: each predictive variable can be dependent on or independent of C. That is, each predictive variable can be considered relevant or irrelevant for clustering purposes.

Let the predictive variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ denote the molecular marker loci, specifically SNPs in our case. The cluster variable, C, represents the population grouping of the given h haploid sequences $D = \{\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(h)}\}$ with $\mathrm{x}^{(l)}$ the SNPs alleles of the l-th individual $\boldsymbol{x}^{(l)} = \{x_1^{(l)}, \ldots, x_n^{(l)}\}$ and $l = 1, \ldots, h$. Note that the algorithm assumes that the number of clusters is know, although it can be varied across independent runs of the algorithm in order to select the best number of population groups.

For each locus X_i, following the idea of model averaging over selective naive Bayes models, we consider two kind of parameters: if the the locus is relevant for clustering purposes, we denote θ_{ijk} as the unknown relative frequency of the allele k at the i-th locus in population j. On the other hand, if the information of the i-th locus is irrelevant to decide the population to which a sequence belongs, the allele frequency does not depend on the population to which each sequence is assigned and therefore θ_{i-k} denotes the relative frequency of allele k at locus X_i. This last relative allele frequency can be calculated directly from the dataset D. Additionally, θ_{C-j} represents the unknown relative frequency of population j in the dataset D.

As was said before, the EMA is an iterative algorithm. At each iteration t, a naive Bayes model for clustering is calculated by means of a Bayesian model averaging over selective naive Bayes where each naive Bayes structure is, a priori, equiprobable. That is, the $\theta_{ijk}^{(t)}$ parameter of the model at the t-th iteration is calculated taking into account not only the parameter θ_{ijk}, but also θ_{i-k}.

In order to perform the model averaging calculations efficiently, it is necessary to assume that the allele frequencies θ_{ijk} and θ_{i-k} for any locus X_i and any population j and allele k are distributed following a Dirichlet distribution with parameters α_{ijk} and α_{i-k} respectively, and that θ_{ijk}, for any population j and allele k, is independent of other $\theta_{i'jk}$ with $i \neq i'$. Similarly, θ_{i-k} is independent of any other $\theta_{i'-k}$. This is known as parameter independence assumption. Moreover, it is assumed that there is no missing values for the predictive variables $\boldsymbol{X}$ in the dataset.

The whole set of parameters for the model learned at iteration t, $\theta_{ijk}^{(t)}$ and $\theta_{C-j}^{(t)}$ for any i, j and k, is denoted as $\boldsymbol{\Theta}^{(t)}$. The EMA algorithm successively performs the E and

MA steps until the difference between parameter set of the models calculated in two consecutive iterations is less than a given parameter, ϵ. The first parameter set, $\mathbf{\Theta}^{(0)}$, is usually taken at random.

For a better understanding of the EMA algorithm, we attempt to give some intuition about the calculations performed at E and MA steps, at each iteration t, in the following two sections.

2.1 E Step (Expectation)

Intuitively, we can see this step as a probabilistic assignation of each individual to each population on the basis of the current model $\mathbf{\Theta}^{(t)}$. Actually, this step computes, given the current model $\mathbf{\Theta}^{(t)}$, the expected number of individuals from a population j that present allele k at the i-th locus when the locus is considered relevant for clustering purposes, $E(N_{ijk}|\mathbf{\Theta}^{(t)})$, and the expected number of individuals classified into population j, $E(N_{C-j}|\mathbf{\Theta}^{(t)})$. The number of individuals of the dataset that present allele k at locus X_i, N_{i-k}, is also necessary for model averaging calculations. This value does not depend on population assignments, therefore, it is constant throughout the algorithm.

After the E step at each iteration t, we have some information about the population membership of each individual. To distinguish between the dataset D and the dataset after the E step where this information has been already computed, we refer to the dataset after the E step at iteration t as $D^{(t)}$.

2.2 MA Step (Model Averaging)

In this step, the EMA algorithm calculates a new set of parametes, $\mathbf{\Theta}^{(t+1)}$, for a naive Bayes model by averaging over all the selective naive Bayes models. These calculations are given by the following equation:

$$p(c_j, \boldsymbol{x}|D^{(t)}) = \sum_S p(c_j, \boldsymbol{x}|D^{(t)}, S)p(S|D^{(t)}) \quad (1)$$

$$= \sum_S \int p(c_j, \boldsymbol{x}|S, \mathbf{\Theta})\ p(\mathbf{\Theta}|S, D^{(t)})d\mathbf{\Theta}\ p(D^{(t)}|S)P(S)$$

where S denotes a specific selective naive Bayes model that sets which loci are considered relevant for clustering purpose and which are not. We abuse the notation by using c_j and $\boldsymbol{x}$ to denote the fact that the cluster variable C takes the j-th value and the molecular marker loci $\boldsymbol{X}$ take a specific SNP alleles $\boldsymbol{x}$. For a more simple notation, when we write $\boldsymbol{x}$, we assume that each SNP X_i takes its k-th allele value.

The general idea of an efficient model averaging over selective naive Bayes is that Equation 1 can be approximated in terms that only depend on each locus (variable X_i) or on each locus and the population membership (X_i and C).

On the one hand, the integral in Equation 1 can be approximated by the *maximum a posteriori* (MAP) parameter configuration.

$$p(\, c_j\, , \boldsymbol{x}|D^{(t)}, S) \approx \frac{\alpha_{C-j} + E(N_{C-j}|\mathbf{\Theta}^{(t)})}{\alpha_C + h}\cdot$$

$$\prod_{i=1}^{n} \frac{\alpha_{ijk} + E(N_{ijk}|\mathbf{\Theta}^{(t)})}{\alpha_{ij} + E(N_{ij}|\mathbf{\Theta}^{(t)})} = \tilde{\theta}^S_{C-j} \prod_{i=1}^{n} \tilde{\theta}^S_{ijk} \quad (2)$$

where $\tilde{\theta}^S_{ijk}$ and $\tilde{\theta}^S_{C-j}$ denote the MAP parameter configuration for a selective naive Bayes structure S. Additionally, $\alpha_{ij} = \sum_k \alpha_{ijk}$ and $E(N_{ij}|\mathbf{\Theta}^{(t)}) = \sum_k E(N_{ijk}|\mathbf{\Theta}^{(t)})$, and similarly for values related to C, where α_{C-j} represents the Dirichlet parameter for θ_{C-j}. Note that S sets if a variable X_i is dependent on or independent of C. Therefore, if S sets that the locus X_i is independent of C, we should use $\tilde{\theta}^S_{i-k}$ and N_{i-k} instead of the $\tilde{\theta}^S_{ijk}$ and $E(N_{ijk}|\mathbf{\Theta}^{(t)})$ respectively in Equation 2, and substitute N_{i-} for $E(N_{ij}|\mathbf{\Theta}^{(t)})$ with $N_{i-} = \sum_k N_{i-k} = h$.

On the other hand, the marginal likelihood can also be written in terms that only depend on X_i or on X_i and C. This is given by the well-known close formula for $p(D|S)$ [Cooper and Herskovits, 1992] adapted to our specific problem. Note that, while $p(D|S)$ is resolvable in closed form when the value of C is known (the dataset is complete), in our case $D^{(t)}$ is not a complete dataset, therefore we are not able to calculate the sufficient statistics N_{ijk} and N_{C-j} but only approximations given the current model $\mathbf{\Theta}^{(t)}$. Hence, the adaptation of Cooper and Herskovits' formula gives an approximation to $p(D^{(t)}|S)$.

$$p(D^{(t)}|S) \approx \frac{\Gamma(\alpha_C)}{\Gamma(\alpha_C + h)} \prod_j \frac{\Gamma(\alpha_{C-j} + E(N_{C-j}|\mathbf{\Theta}^{(t)}))}{\Gamma(\alpha_{C-j})} \cdot$$

$$\prod_{i=1}^{n} \prod_j \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + E(N_{ij}|\mathbf{\Theta}^{(t)}))} \prod_k \frac{\Gamma(\alpha_{ijk} + E(N_{ijk}|\mathbf{\Theta}^{(t)}))}{\Gamma(\alpha_{ijk})} \quad (3)$$

where $\Gamma(\cdot)$ represents the gamma function. Of course, again it is S which sets if we should use α_{i-k}, N_{i-j} and N_{i-} instead of the originals in Equation 3. The approximation given by this equation in the model averaging process, have been compared to a Monte Carlo approximation, which is a more accurate and computationally expensive technique to approximate $p(D|S)$, obtaining similar results [Santafé *et al.*, 2006].

At this point, as a consequence of parameter independence assumption, we can state that if two different selective naive Bayes structures set the same relationship between locus X_i and C (in both structures X_i is relevant or irrelevant for clustering purposes) the calculations related to locus X_i in Equations 2 and 3 are the same for both structures. This is essential for an efficient model averaging calculations since it allows to eliminate the dependence on S in the model averaging calculations performed in Equation 1. That is, using the approximations given by Equations 2 and 3 in Equation 1, and grouping these calculations in terms that depend on each variable X_i, each one of these groups will contain only two kind of terms: the ones that consider X_i relevant for clustering, ρ_{ijk}, and the ones that consider X_i irrelevant for clustering, ρ_{i-k}. Therefore, the model averaging calculations from Equation 1 can be approximated as follows:

$$p(c_j, \boldsymbol{x}|D^{(t)}) \approx \rho_{C-j} \prod_{i=1}^{n} (\rho_{i-k} + \rho_{ijk}) \quad (4)$$

where ρ_{C-j} is the term which groups the calculations related only to the cluster variable. Thus, the new set of parameters, $\mathbf{\Theta}^{(t+1)}$, can be calculated from ρ_{C-j}, ρ_{i-k} and ρ_{ijk}. The expression of terms ρ_{C-j}, ρ_{i-k} and ρ_{ijk} is given in Equation 5.

$$\rho_{C-j} \propto \tilde{\theta}_{C-j} \ \frac{\Gamma(\alpha_C)}{\Gamma(\alpha_C + h)} \prod_j \frac{\Gamma(\alpha_{C-j} + E(N_{C-j}|\boldsymbol{\Theta}^{(t)}))}{\Gamma(\alpha_{C-j})} \quad (5)$$

$$\rho_{i-k} \propto \tilde{\theta}_{i-k} \ \prod_j \frac{\Gamma(\alpha_i)}{\Gamma(\alpha_i + h)} \prod_k \frac{\Gamma(\alpha_{i-k} + N_{i-k})}{\Gamma(\alpha_{i-k})}$$

$$\rho_{ijk} \propto \tilde{\theta}_{ijk} \ \prod_j \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + E(N_{ij}|\boldsymbol{\Theta}^{(t)}))} \prod_k \frac{\Gamma(\alpha_{ijk} + E(N_{ijk}|\boldsymbol{\Theta}^{(t)}))}{\Gamma(\alpha_{ijk})}$$

where $\alpha_i = \sum_k \alpha_{i-k}$. Therefore, the calculation of the parameters for the naive Bayes model at $t+1$ step are given by:

$$\theta_{ijk}^{(t+1)} = \rho_{i-k} + \rho_{ijk} \qquad \theta_{C-j}^{(t+1)} = \rho_{C-j}$$

2.3 Multi-start EMA

The EMA is a greedy algorithm that is susceptible to be trapped in a local optima. The results obtained by the algorithm depend on the random initialization of the parameters. Therefore, we propose the use of a multi-start algorithm where m different runs of the algorithm with different random initializations are performed. In [Santafé *et al.*, 2006] different criteria to obtain the final model from the multi-start process are proposed.

In our case, we use the best choice multi-start EMA, where the best model, in terms of likelihood, among the m models calculated by the multi-start process is selected to be the final model. This is not a pure Bayesian approach to the model averaging process but, in practice, it works, at least, as well as other more complicated criteria.

2.4 Selecting the most relevant SNPs for clustering

The model averaging process performed by the EMA algorithm can be seen as an implicit unsupervised variable selection that is incorporated in the final model. In fact, although the EMA algorithm obtains a naive Bayes model where all the predictive variables (maker loci) are independent given the value of C (population assignation), the parameters of the resultant naive Bayes model are calculated by a model averaging over selective naive Bayes and thus, these parametes should reflect the significance of each locus for clustering purposes.

In this section we propose a two-step test that can be used to obtain information about relevant locus that is implicitly contained in the final naive Bayes model calculated by the EMA algorithm. This test is based on mutual information. It is known [Cover and Thomas, 1991] that the statistic $2NI(X_i, C)$ asymptotically follows a Chi-square probability distribution with $(r_i - 1)(r_C - 1)$ degrees of freedom. In our case, r_i is the number of alleles that a locus X_i can present and r_C the total number of populations.

The mutual information of a locus X_i and the cluster variable, or the mutual information of two predictive variables $X_{i'}$ and $X_{i''}$ with $i' \neq i''$ can be calculated using the naive Bayes model obtained by the EMA algorithm. Thus, a Chi-square test can be performed to decide which marker loci are relevant for the clustering process. In the first step, we set a test threshold p_{rel} and use a Chi-square test to filter out those predictive variables which are considered not relevant for clustering purposes. This first step selects the relevant SNPs but the set of selected SNPs may contain redundant information. As a consequence, we develop a second step to filter out redundant SNPs by again using a Chi-square test with a test threshold p_{red}. In this second step we calculate the pairwise mutual information of the variables selected in the first step and use this value to decide whether or not two variables are redundant. Figure 2 describes the two-step algorithm to obtain the set of SNPs, $\boldsymbol{X}_{rel}$, which are relevant to obtain the underlying group structure.

$\boldsymbol{X}_{rel} = (X_1, \ldots, X_n)$

– STEP 1 –

for $i = 1$ **to** n
  if $2NI(X_i, C) < \chi^2_{(r_i-1)(r_C-1);1-p_{rel}}$
    remove X_i from $\boldsymbol{X}_{rel}$
  end if
end for

– STEP 2 –

for all $X_{i'}, X_{i''}$ **with** $X_{i'}, X_{i''} \in \boldsymbol{X}_{rel}$ **and** $i' \neq i''$
  if $2NI(X_{i'}, X_{i''}) < \chi^2_{(r_{i'}-1)(r_{i''}-1);1-p_{red}}$
    if $I(X_{i'}, C) < I(X_{i''}, C)$
      remove $X_{i'}$ from $\boldsymbol{X}_{rel}$
    else
      remove $X_{i''}$ from $\boldsymbol{X}_{rel}$
    end if
  end if
end for

Figure 2: Pseudo-code for SNPs selection.

The thresholds p_{rel} and p_{red} can be used to control the number of selected variables. On the one hand, the bigger p_{rel} is, the less variables are selected as relevant for clustering. On the other hand, as p_{red} decreases, the number of variables considered redundant increases and therefore, the final number of selected variables is smaller.

3 Results

In order to show how the characteristics of the EMA algorithm can fit properly to the population substructure problem, we firstly use the toy example introduced in [O'Rourke *et al.*, 2005]. The dataset for this toy example is composed of individuals represented by 50-character bitstrings. We generate three ancestral sequences with relative Hamming distances 2, 3 and 5. Each ancestral sequence is used to generate a set of 19 new clone individuals in which random mutations may happen at each position of the string with a mutation rate 0.05. Thus, we obtain 60 individuals grouped into three different populations with an expected Hamming distance of 2.5 between a mutant and its ancestral sequence. The individuals from the three different population are differentiated by 5 positions of the sequence and the rest of the positions may only differ from one individual to another because of random mutations. In fact, it is enough with only 2 of the 5 positions to differentiate between individuals from the three populations because there

is redundant information.

We use a multi-start EMA algorithm with $\epsilon = 0.01$ and $m = 1000$ runs in the multi-start process to cluster the 60 individuals into 3 clusters. Since the EMA algorithm is not deterministic, we perform 10 runs of the multi-start algorithm that, on average, classify 95% of the sequences into their original population with null standard deviation. Moreover, the naive Bayes model obtained by the multi-start EMA at each run is used to obtain the set of relevant positions to decide the clustering membership. The test described in Section 2.4 with parameters $p_{rel} = 0.01$ and $p_{red} = 0.01$ is used to obtain these relevant positions. All the 10 runs yield the selection of only two positions of the five positions that diffentiate between the three populations. Thus, the algorithm correctly selects the minimum number of positions to differentiate between populations.

Moreover, the *Structure* software [Pritchard *et al.*, 2000a] with default parameters and $10,000$ models for the burning period and also $10,000$ iterations to learn the model is able to classify, on average over ten runs, only 79.64% of the sequences into their population of origin with standard deviation 9.91%.

The proposed toy example shows that the EMA algorithm presents a good behavior for clustering sequences from different populations and where random mutation across the sequence positions may happen. In the following section we apply the EMA algorithm to a real dataset.

3.1 Human Population Substructure

For this experiment, we use the dataset of common SNPs reported in [Hinds *et al.*, 2005]. This dataset contains about 1.5 million SNPs uniformly distributed across the human genome and which are common to, at least, individuals from the three human populations under study: European, African and Asian. The data came from the genotype of 71 unrelated individuals: 24 European-American, 23 African-American and 24 Han-Chinese from the Los Angeles area.

Following the experimental description from [O'Rourke *et al.*, 2005], and in order to avoid linkage disequilibrium from proximity in the genome, we only use every thousandth SNP, leaving a total of $1,520$ marker loci. The EMA algorithm, proposed in this paper to tackle the population substructure problem, uses haploid data. Therefore, each individual gives rise to two haplotype sequences belonging to the same population. Hence, the dataset is made up of 142 sequences with $1,520$ SNPs each. On the other hand, this dataset with $1,520$ SNPs contains information about SNPs from all over the human genome. However, in other real problems, not all this information is always available. Sometimes only SNPs from one or several chromosomal segments are provided and then, the population substructure is more difficult to retrieve. In order to simulate these situations, we split the dataset into 19 datasets with only 80 SNPs. In the experiments we refer to the dataset that contains all the $1,520$ SNPs as $dsSNPs$. The smaller datasets with 80 SNPs are denoted as $ds1, \ldots, ds19$.

Table 1 shown the percentage of individuals, on average over 10 independent runs, correctly assigned to their population of origin using both multi-start EMA and *Structure* algorithms for each dataset. The parameters used for both algorithms are the same as those used in the toy example. A Man-Withney test at 0.01 level is performed to check if the difference between the two algorithms is statistically significant. Those values which are significantly better are written in bold in Table 1. We can see that the results obtained by the multi-start EMA algorithm outperform the structure software, and the differences are statistically significant for all the datasets.

	Structure		**EMA**	
Dataset	Mean	Std	Mean	Std
$ds1$	87.28	0.55	**95.07**	0.47
$ds2$	88.27	1.35	**93.24**	0.36
$ds3$	88.48	0.48	**93.38**	1.49
$ds4$	91.27	0.34	**95.77**	0.00
$ds5$	83.75	7.67	**93.94**	0.36
$ds6$	84.53	0.51	**86.13**	0.48
$ds7$	92.20	0.25	**96.48**	0.00
$ds8$	74.79	2.66	**82.75**	2.11
$ds9$	81.86	0.62	**86.06**	1.78
$ds10$	85.28	0.74	**89.08**	1.11
$ds11$	87.25	0.38	**90.14**	0.00
$ds12$	89.59	0.35	**91.55**	0.00
$ds13$	90.72	0.38	**94.37**	0.00
$ds14$	89.41	0.26	**95.99**	0.48
$ds15$	86.08	0.30	**90.07**	0.52
$ds16$	84.96	0.84	**93.45**	0.48
$ds17$	88.34	0.17	**93.52**	0.30
$ds18$	81.16	0.53	**86.55**	2.92
$ds19$	82.65	0.86	**86.62**	0.00
Mean over $ds1$-$ds19$	86.20	—	**91.27**	—
$dsSNPs$	96.78	0.09	**100**	0.00

Table 1: Percentage of correctly assigned individuals to their population of origin (mean and standard deviation over 10 runs) using both EMA and *Structure* algorithms.

The SSCC method proposed in [O'Rourke *et al.*, 2005] also obtain 100% of correct classications for $dsSNPs$ dataset. In the case of the dataset with only 80 SNPs, only the mean value over the 19 datasets is reported in the paper. The multi-start EMA algorithm obtains, on average over the 19 datasets, 91.27% of correct classified individuals. This figure is very close to the precision obtainded by the SSCC, 91.80%

Another important characteristic of the EMA algorithm is its ability to select the set of relevant SNPs for clustering purposes. Using the $dsSNPs$ dataset and setting $p_{rel} = p_{red} = 0.01$, the 10 runs of the multi-start EMA algorithm give rise to 10 sets of selected relevant SNPs. These datasets contain, on average, 277.1 relevant SNPs and 146 of them are common to the 10 sets. Note that each set of relevant SNPs is selected on the basis of a different naive Bayes model learned with the multi-start EMA algorithm. Therefore, although two sets of relevant SNPs share many SNPs, some of them may be different. The fact that two sets of relevant SNPs contain different SNPs does not necessarily mean that one set is better than the other because they may contain different subsets of non-redundant SNPs. In order to evaluate the sets of relevant SNPs obtained by the multi-start EMA algorithm, we run the *Structure* software with the same parameters on these datasets. The percentage of correctly classified individuals is, on average over the 10 different sets of relevant SNPs, 94.98% with standard deviation 4.66%. Note that the reduction in the number of SNPs in the dataset is very big while the percentage of correctly classified individuals is similar.

4 Discussion

In this paper we propose the use of the EMA algorithm, originally introduced in [Santafé *et al.*, 2006], in population substructure problems. This algorithm learns a naive Bayes model for clustering by means of a Bayesian model averaging over selective naive Bayes models. That implies that the Bayesian model averaging includes in the learned model a kind of implicit variable selection of relevant predictive variables. This special characteristic make the algorithm appropriated for the considered problem. Moreover, in the same clustering process we are able to retrieve the underlying population substructure and to obtain the set of relevant markers used to retrieve this population substructure. In order to show this, we evaluate the algorithm in a toy example but also in a real problem. In the experiments performed in the paper the multi-start EMA algorithm outperforms other common used algorithms such as *Structure*.

In the literature, we can find some algorithms based on *Structure* that obtain quite good results [Rosenberg *et al.*, 2002; Patterson *et al.*, 2004]. By contrast, the EMA algorithm provides a probabilistic tool that can be used to identify unknown genetically related subgroups of samples and identify the set of SNPs that is providing most of the relevant genetic differences among the identified groups.

The current version of the EMA algorithm can not deal with missing values in the predictive variables. This could be a problem for a general use of the EMA algorithm in the population substructure problem since it is very common that not all the marker loci can be sequenced for all the individuals from a study. However, this assumption can be easily relaxed by modifying the E step.

On the other hand, the parameters of the EMA algorithm may influence the obtained results. Specifically the selection of the thresholds p_{rel} and p_{red} affects the number of selected SNPs. The aim of SNP selection is to select as smaller set of SNPs as possible but this subset of SNPs should contain enough information to describe the obtained clustering structure. It would be interesting an empirical evaluation of the impact of these parameters on the performance of the SNPs selection. Other future work may include the use of several runs of the algorithm with different number of clusters in order to evaluate how it is able to detect subpopulations in the dataset.

Acknowledgments

The authors are grateful to the anonymous reviewers for helpful comments. This work was done when the second author was in the university of California (San Diego) under grant PR2006-0315. The work was supported by the SAIOTEK-Autoinmune (II) 2006 and Etortek research projects from the Basque Government, by the Spanish Ministerio de Educación y Ciencia under grant TIN 2005-03824, and by the Government of Navarre under a PhD grant.

References

[Barash and Friedman, 2002] Y. Barash and N. Friedman. Context-specific Bayesian clustering for gene expression data. *Journal of Computational Biology*, 9:169–191, 2002.

[Botstein and Risch, 2003] D. Botstein and N. Risch. Discovering genotypes underlying human phenotypes: Past successes for Mendelian disease, future approaches for complex diseae. *Nature Genetics*, 33:228–237, 2003.

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

[Corander *et al.*, 2004] J. Corander, P. Waldmann, P. Marttinen, and M. J. Sillanpää. BAPS 2: Enhanced possibities for the analysis of genetic population structure. *Bioinformatics*, 20(15):2363–2369, 2004.

[Cover and Thomas, 1991] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[Dash and Cooper, 2004] D. Dash and G. F. Cooper. Model averaging for prediction with discrete Bayesian networks. *Journal of Machine Learning Research*, 5:1177–1203, 2004.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[Hinds *et al.*, 2005] D. A. Hinds, L. L. Stuve, G. B. Nilsen, E. Halperin, E. Eskin, D. G. Ballinger, K. A. Frazer, and D. R. Cox. Whole-genome patterns of common DNA variation in three human populations. *Science*, 307:1072–1079, 2005.

[O'Rourke *et al.*, 2005] S. O'Rourke, G. Chechik, and E. Eskin. Separation of overlapping subpopulation by mutual information. In *Proceedings of the NIPS Workshop on Computational Biology and the Analysis of Heterogeneous Data*, 2005.

[Patterson *et al.*, 2004] N. Patterson, N. Hattangadi, B. Lane, k: E: Lohmuller, D. A. Hafler, J. R. Oksenberg, S. L. Hauser, M. W. Simith, S. J. O'Brien, D. Altshuler, J. Daly, and D. Reich. Methods for high-density admixture mapping of disease genes. *American Journal of Human Genetics*, 74:1001–1013, 2004.

[Pritchard *et al.*, 2000a] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

[Pritchard *et al.*, 2000b] J.K. Pritchard, M. Stephens, N.A. Rosenberg, and P. Donnelly. Association mapping in structured populations. *American Journal of Human Genetics*, 67:170–181, 2000.

[Rosenberg *et al.*, 2002] N. A. Rosenberg, J. K. Pritchard, J. L. Weber, H. M. Cann, K. K. Kidd, L. A. Zhivotovsky, and M. W. Feldman. Genetic structure of human population. *Science*, 298:2381–2385, 2002.

[Santafé *et al.*, 2006] G. Santafé, J. A. Lozano, and P. Larrañaga. Bayesian model averaging of naive Bayes for clustering. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 2006. Accepted for publication.

[Sillanpää *et al.*, 2001] M. J. Sillanpää, R. Kilpikari, S. Ripati, P. Onkamo, and P. Uimari. Bayesian association mapping for quantitative traits in a mixture of two populations. *Genetic Epidemiology*, 21:S692–S699, 2001.

Identification of Feedback Structures from Gene Expression Time Series

Fulvia Ferrazzi[1,2], Paola Sebastiani[3], Riccardo Bellazzi[1], Isaac S Kohane[2], Marco F Ramoni[2]

[1]Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, Italy
[2]Children's Hospital Informatics Program and Harvard Partners Center for Genetics and Genomics, Harvard Medical School, Boston, USA
[3]Department of Biostatistics, Boston University School of Public Health, Boston, USA

Abstract

A critical aspect of gene regulation is the presence of cyclic control structures: in this paper we present an approach to identify feedback loop mechanisms from gene expression time series. Our approach is based on dynamic Bayesian networks and we apply it to the analysis of gene expression data measured during cell cycle in a human cell line. Cell cycle control mechanisms are fundamental in the development and treatment of tumors and the identification of feedback control loops can highlight central genes in the regulation and promising pharmacogenomics targets.

1 Introduction

Inference of gene expression networks from DNA microarray temporal data is nowadays one of the most challenging problems in functional genomics. In particular, the study of temporal profiles is considered very promising for the discovery of functional relationships among genes, as it allows to observe regulatory mechanisms in action and can help to identify genes taking part in the same cellular processes. A critical modelling aspect of any regulatory mechanism is the presence of cyclic control structures: here we present an approach to identify feedback loop mechanisms from gene expression time series measured through DNA microarrays. Our approach is based on dynamic Bayesian networks (DBNs), a special class of Bayesian networks particularly suited to study dynamic gene expression data, that is time series of expression measurements [Murphy and Mian, 1999; Sebastiani *et al.*, 2005a]. DBNs allow us to overcome the inability of Bayesian networks to represent cycles among variables and thus make the discovery of feedback loops in gene regulatory networks feasible.

2 Background

Bayesian networks (BNs) are a formalism for the representation and the use of probabilistic knowledge widely employed in various fields such as Artificial Intelligence and Statistics. BNs are becoming an increasingly popular modeling framework for different types of genomic and proteomic data. They offer indeed a number of significant advantages over other methods: they are able to model stochasticity, to incorporate prior knowledge and to handle hidden variables and missing data in a principled way. Bayesian networks have been applied to the analysis of gene expression data [Friedman *et al.*, 2000; Segal *et al.*, 2003], protein-protein interactions [Jansen *et al.*, 2003] and genotype data [Sebastiani *et al.*, 2005b].

Dynamic Bayesian networks (DBNs) are a particular type of BNs which models the stochastic evolution of a group of random variables over time. A traditional Bayesian network is only able to offer a static view of the system under analysis, useful in the case one is interested in modeling its steady state. DBNs are instead able to model how genes regulate each other over time.

Murphy and Mian in 1999 were the first to propose the suitability of DBNs for modeling time series gene expression data: they reviewed different learning techniques but they did not apply them to a real biological dataset [Murphy and Mian, 1999]. When microarray data availability increased, works in which DBNs were used to analyze real gene expression data started to be published. Ong *et al.* used time series data measured in *E. Coli* to infer a DBN network with a discrete model of regulation and showed that this network was able to identify genes in a common regulatory pathway [Ong *et al.*, 2002]. Perrin *et al.* proposed to treat gene expression variables as continuous and used a dynamical system model with Gaussian noise [Perrin *et al.*, 2003], while Kim *et al.* presented an algorithm in which the variables were still treated as continuous but described with a nonparametric regression model [Kim *et al.*, 2003]. Husmeier performed simulation studies to assess how the network inference performance varied according to changes in a number of factors, such as the training set size or the inclusion of further, sequence-based information [Husmeier, 2003]. Yu *et al.* exploited simulated data as well to test a few proposed advances for dynamic Bayesian network inference algorithms, especially in the context of limited quantities of expression data [Yu *et al.*, 2004]. More recently Zou *et al.* proposed an approach aimed at increasing the accuracy and reducing the computational time by limiting potential regulators to those genes with either earlier or simultaneous expression changes (up- or down-regulation) in relation to their target genes [Zou and Conzen, 2005]. Bernard and Hartemink presented an interesting method for jointly learning dynamic models of transcriptional regulatory networks from gene expression data and transcription factor binding location data [Bernard and Hartemink, 2005]. In our paper we employ a DBN ap-

proach based on Linear Gaussian models to describe gene relationships. This approach, which is described in the following Section, is very useful for a first level, genome-wide analysis of high throughput dynamic data.

3 Method

A DBN is a directed acyclic graph that encodes a joint probability distribution over a set of random variables: the nodes in the graph represent these stochastic variables and directed arcs represent the dependencies among them, which are quantified by conditional probability distributions.

Supposing expression values for n genes in p consecutive time points are available, it is possible to indicate with $Y(t) = \{Y_1(t), Y_2(t), \ldots, Y_n(t)\}$ the set of random variables representing gene expression values at time t. We assume that the process under study (the dynamics of gene expression) is

- *Markovian*, i.e. $p(Y(t+1)|Y(0), \ldots, Y(t)) = p(Y(t+1)|Y(t))$ and
- *stationary*, i.e. the transition probability $p(Y(t+1)|Y(t))$ is independent of t.

Furthermore, we assume that a certain time lag is necessary for the expression of a gene to affect the expression of other genes, so that *no instantaneous relationship* between the expression levels of two genes is possible. Given these three assumptions, it is necessary to learn only the transition network between expression values at time t and at time $t+1$ [Friedman *et al.*, 1998].

The use of DBNs allows us to overcome the inability of Bayesian networks to represent cycles among variables and thus makes the discovery of feedback loops in gene networks feasible. Indeed, the necessary acyclic structure of the directed graph that encodes the dependencies between the network variables is no longer a limitation in the framework of DBNs. Considering for example two genes A and B and indicating with the subscripts t and $t+1$ their expression values in two consecutive time points, if two links $A_t \rightarrow B_{t+1}$ and $B_t \rightarrow A_{t+1}$ are found through the learning of a DBN, it is possible to say that there is a feedback loop connecting these two genes, either directly or through other genes.

As Bayesian networks, DBNs can be learned from the data. To this aim, a probability model and a search strategy must be chosen. We assume that the variables $Y_1, \ldots, Y_n$ are all continuous, and that the conditional distribution of each variable Y_i given its parents $Pa(y_i) = (Y_{i1}, \ldots, Y_{ip(i)})$ follows a *Gaussian distribution* with mean μ_i that is a linear function of the parent variables and conditional variance σ_i^2 [Sebastiani *et al.*, 2005a]. The dependency of each variable on its parents is thus represented by the linear regression equation:

$$\mu_i = \beta_{i0} + \sum_j \beta_{ij} y_{ij} \qquad (1)$$

that models the conditional mean of Y_i at time $t+1$ given the parent values y_{ij}, measured at time t.

In accordance with the Bayesian literature, we look for the network associated with the maximum posterior probability with respect to the data. As an exhaustive search over the space of all possible networks encoding the probabilistic dependencies among the n analyzed variables is not feasible, we exploit a finite horizon local search and we explore the dependency of each variable on all the variables at the previous time point [Cooper and Herskovitz, 1992; Sebastiani *et al.*, 2005a].

4 Data analysis and discussion

We decided to focus our investigation on the cell cycle, a process of critical importance where feedback control loops are expected to play a central regulatory role. The analysis of cell cycle regulation is indeed one of the most important areas of research in the medical and biological community, as an understanding of how cells divide and proliferate is crucial for the study of various diseases, most notably cancer. In this analysis, the discovery of feedback loop mechanisms could be extremely useful in highlighting genes which play a key role in cell cycle control and which thus constitute potential targets for an effective cancer therapy.

For this purpose, we use a database of gene expression data measured in a human epithelial cell line derived from a cervical carcinoma [Whitfield *et al.*, 2002]. This database, available from the web (http://genome-www.stanford.edu/Human-CellCycle/Hela/), includes data coming from 5 different experiments. In particular, in one of these, gene expression values were measured every hour, from 0 to 46 hs, using cDNA microarrays containing about 40000 probes. This dataset currently represents one of the most extensive collections of gene expression temporal data. Its abundant amount of samples (47 time points) constitutes a significant advantage for the inference of the network. In fact, a problem often encountered in the learning is that the number of variables (genes/probes) is normally very high, while the available measurements (gene expression values) are usually very few, thus often leading to a low degree of sensitivity and specificity in the inferred networks [Husmeier, 2003].

We are currently analyzing a set of 1000 probes identified as periodical in [Whitfield *et al.*, 2002]: the authors, employing a Fourier transform and correlation to known cell cycle yeast genes, calculated a periodicity score for each of the probes and selected about 1000 probes whose score exceeded a carefully chosen threshold. We analyzed their time profiles with the algorithm presented in the previous Section. The inferred DBN is very parsimonious, with a number of parents for every node that ranges from 0 to 2.

Starting from this DBN, we reconstructed a gene regulatory network, in which nodes referring to the same variable at consecutive time points (eg. A_t and A_{t+1}) were collapsed into a single node. This regulatory network is not a Bayesian network anymore and can instead contain loops. Figure 1 presents an example of how a gene regulatory network is reconstructed from the transition network.

The reconstructed gene regulatory network is shown in Figure 2. The analysis of this network led us to concentrate our attention on a group of 12 probes that appear to be involved in various feedback loops. Some of these probes map to well characterized cell cycle genes, while others highlight potential new key players in cell cycle regulation.

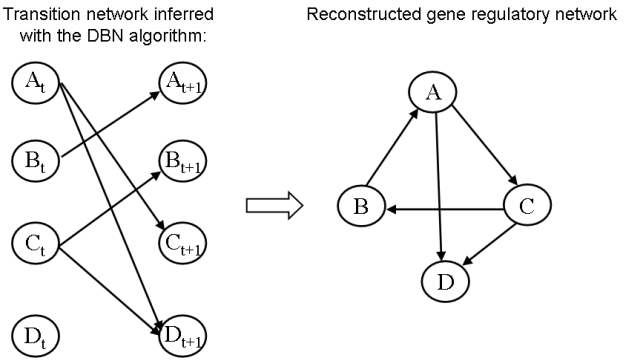


Figure 1: Example of translation of the transition network inferred by the DBN algorithm into a gene regulatory network.

We statistically validated the obtained model, both evaluating the *goodness of fit* of the network and assessing its *predictive accuracy*. The fitting appeared to be satisfactory, as it is visible in Figures 3 and 4. Figure 3 presents an example of the measured and fitted profiles relative to two genes. Figure 4 presents instead diagnostic plots relative to the network learnt. The symmetry of the standardized residuals, together with the closeness of the fitted and observed values and the lack of any significant patterns in the scatter plot of the fitted values vs the standardized residuals suggest that the model used is able to provide a good approximation of the analyzed temporal profiles.

The predictive accuracy was evaluated on an independent test set constituted by the data coming from another of the 5 experiments performed by Whitfield *et al.*: here a different cell synchronization method was employed and expression values were measured from 0 to 36 hours, every 2 hours. Results of the predictive analysis show that our model provides a good description also of the expression profiles measured in this dataset.

5 Conclusions

Linear Gaussian Networks are a powerful instrument to represent gene interactions. The results showed in this paper confirm the suitability of these networks for a first level analysis of high-throughput gene expression data. We have now undertaken an experimental validation of the functional implications of our model to assess the regulatory role of the genes involved in the feedback loops.

Moreover we are devoting additional efforts to understanding to what extent Linear Gaussian Networks are able to infer the complex dynamic interactions among cellular variables. For this reason, we have undertaken a study on simulated data aimed at quantitatively evaluating the performance of these networks, comparing the inferred connections with the "real" ones [Ferrazzi *et al.*, 2006]. Results are encouraging and generalizations of the linear regression model employed in this paper may provide additional flexibility in modeling the analyzed temporal profiles.

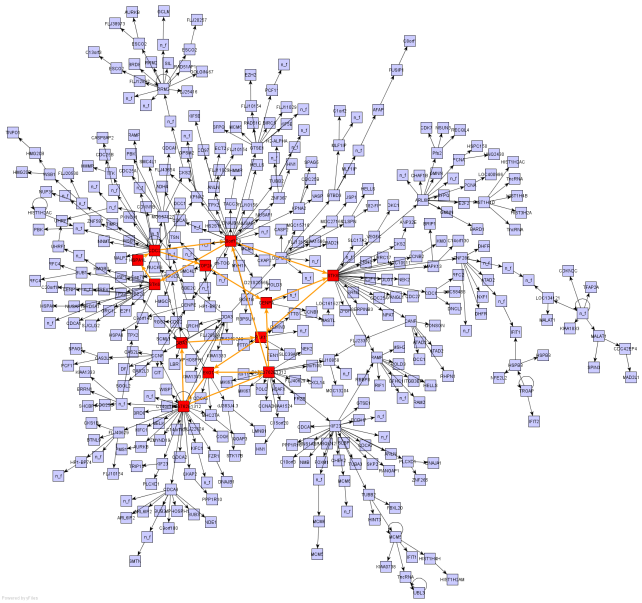


Figure 2: Gene regulatory network inferred using the data relative to the 1000 probes identified as cell cycle regulated in [Whitfield *et al.*, 2002]. The nodes in a darker color are involved in feedback loops.

References

[Bernard and Hartemink, 2005] A. Bernard and A. J. Hartemink. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Proceedings of the Pacific Symposium on Biocomputing*, pages 459–70, 2005.

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

[Ferrazzi *et al.*, 2006] F. Ferrazzi, P. Sebastiani, I. S. Kohane, M. F. Ramoni, and R. Bellazzi. Dynamic Bayesian networks in modelling cellular systems: a critical appraisal on simulated data. *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, 2006.

[Friedman *et al.*, 1998] N. Friedman, K. Murphy, and S. Russel. Learning the structure of dynamic probabilistic networks. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 139–147, 1998.

[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*, 2000.

[Husmeier, 2003] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19:2271–2282, 2003.

[Jansen *et al.*, 2003] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A Bayesian net-

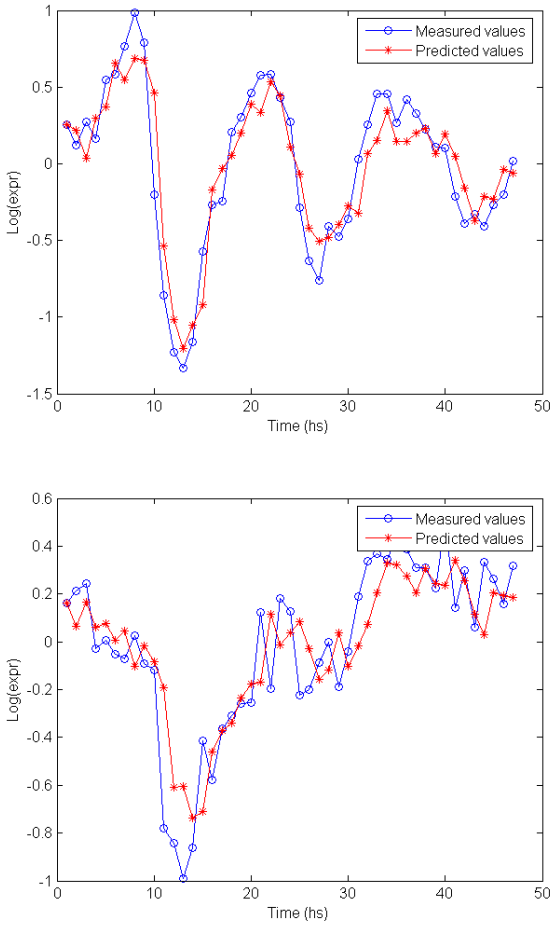
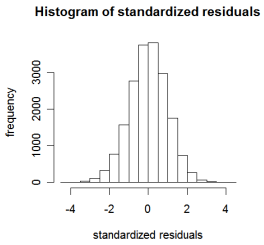
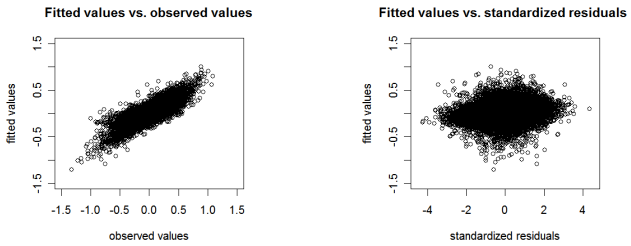


Figure 3: Measured and fitted profiles for two analyzed probes.



(a) Histogram of standardized residuals



(b) Fitted vs observed values (c) Fitted values vs standardized res.

Figure 4: Diagnostic plots for the network model learnt.

Genomic Signal Processing and Statistics. EURASIP Book Series on Signal Processing and Communications, 2005.

[Sebastiani *et al.*, 2005b] P. Sebastiani, M. F. Ramoni, V. Nolan, C. T. Baldwin, and M. H. Steinberg. Genetic dissection and prognostic modeling of overt stroke in sickle cell anemia. *Nature Genetics*, 37:435–40, 2005.

[Segal *et al.*, 2003] E. Segal, M. Shapira, A. Regev, 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*, 34:166–176, 2003.

[Whitfield *et al.*, 2002] M.L. Whitfield, G. Sherlock, A.J. Saldanha, J.I. Murray, C.A. Ball, K.E. Alexander, J.C. Matese, C.M. Perou, M.M. Hurt, P.O. Brown, and D. Botstein. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular Biology of the Cell*, 13(6):1977–2000, 2002.

[Yu *et al.*, 2004] J. Yu, V. A. Smith, P. P. Wang, A. J. Hartemink, and E. D. Jarvis. Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20:3594–603, 2004.

[Zou and Conzen, 2005] M. Zou and S. D. Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21:71–9, 2005.

works approach for predicting protein-protein interactions from genomic data. *Science*, 302:449–53, 2003.

[Kim *et al.*, 2003] S. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*, 4:228–235, 2003.

[Murphy and Mian, 1999] K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. *Technical Report, Berkeley, CA Computer Science Division, University of California*, 1999.

[Ong *et al.*, 2002] I. M. Ong, J. D. Glasner, and D. Page. Modelling regulatory pathways in E. coli from time series expression profiles. *Bioinformatics*, 18 (Suppl 1):S241–8, 2002.

[Perrin *et al.*, 2003] B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. D'Alche-Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19 (Suppl 2):II138–II148, 2003.

[Sebastiani *et al.*, 2005a] P. Sebastiani, M. Abad, and M.F. Ramoni. Bayesian networks for genomic analysis.

Visualisation of Associations Between Nucleotides in SNP Neighbourhoods

Kimmo Kulovesi[*,†], **Juho Muhonen**[*], **Ilkka Lappalainen**[‡], **Pentti T. Riikonen**[§], **Mauno Vihinen**[¶,†], **Hannu Toivonen**[*,‖] and **Tomi A. Pasanen**[*]

Abstract

A large number of single nucleotide polymorphisms have been mapped onto the human genome. Mutations are induced through endogenous and exogenous processes, and these procedures have been shown to be sequence-dependent. Association mining is a powerful tool for analyzing sequence neighbourhoods; however, visualisation is essential for pattern recognition because of the abundance of resulting association rules.

A software tool was developed to visualize position interdependencies within the sequence variation data. The software is capable of interactive reorganization of the association rules, enabling fast and easy exploration of the data using a standard web browser. The software and its complete source code is freely available at: http://www.cs.helsinki.fi/group/bioalgss/asvis/

1 Introduction

A single nucleotide polymorphism (SNP) is a site in DNA where at least two different nucleotides occur in a specific population, the less frequent nucleotide(s) occurring at a frequency of 1% or more. The nucleotide variation between individuals forms the genetic background responsible for biological and physical differences such as colour of hair, susceptibility to a disease or response to specific treatment. The International HapMap project aims to characterise the common human sequence variations [International HapMap Consortium, 2005]. The data is publicly available in the dbSNP database [Wheeler *et al.*, 2005].

New mutations arise by errors in endogenous processes involved in maintaining genomic stability, or are induced by various exogenous agents, such as UV radiation [Jiricny, 1998]. The efficiency and specificity of these processes is DNA sequence dependent [Cooper and Krawczak, 1993]. Data mining can be used to analyze the sequence neighbourhoods of neutral and disease-causing SNPs in order to better understand the genetic differences that underlie pathogenic conditions.

Association rules are suitable for this task. Discovery of association rules is popular in data mining and it has a wide variety of applications. In principle, individual association rules describing co-occurrences of sets of attributes in the input data are straightforward to interpret. However, the very large number of resulting association rules seriously hinders their analysis. General-purpose visualisation tools are ill-suited for association rules, and the few available association visu-alisation tools are not suitable for position-dependent SNP neighbourhoods. Furthermore, the existing tools for visualizing SNP neighbouring-nucleotide biases are not applicable to association rules (for example, [Zhang and Zhao, 2005]). Here we introduce a new, publicly available visualisation tool with interactive controls for the selection and arrangement of association rules obtained from SNP data. The tool has a novel position-dependent display of association rules. It can also be used with other similar data, such as protein sequences. The tool provides simple and legible graphical output.

2 System and Methods

For demonstration in this paper, sequence variation data was extracted from the dbSNP (build 124). Only true polymorphisms that could be located within gene coding regions of the human genome (build 35) were used. We consider a sequence neighbourhood of each mutation site that extends up to ten nucleotides on both sides. The mutation position is numbered zero, while positive and negative position numbers denote the distance of following and preceding nucleotides, respectively.

Our software consists of two programs, Firm and AsVis. Firm is used for application-independent discovery of association rules. As an example, the rule $\{0{:}'C \rightarrow T'\} \longleftarrow \{+1{:}'G'\}$ indicates that the substitution of cytosine (C) by thymine (T) is probable when the mutation site is immediately followed by guanine (G) (see Figure 1). To rank the association rules, Firm yields a number of measures for the strength and generality of each rule within the data. Support is the number of records that fully match the rule. Con-

[*]Department of Computer Science, FI-00014 University of Helsinki, Finland

[†]Institute of Medical Technology, FI-33014 University of Tampere, Finland

[‡]Department of Chemistry, Cambridge University, CB2 1EW Cambridge, UK

[§]Department of Information Technology, FI-20520 University of Turku, Finland

[¶]Research Unit, Tampere University Hospital, FI-33520 Tampere, Finland

[‖]Department of Computer Science, University of Freiburg, D-79110 Freiburg, Germany

Consequent	<=	Condition	Support (s)		Frequency (f)		Confidence (c)		Lift (l)		J-Measure (j)	
0:'>T'	<=	0:'C>' 1:'G' 2:'A'	s=	1140	f=	4.1	c=	87.9	l=	2.49	j=	0.012
0:'>T'	<=	0:'C>' 1:'G' -1:'C'	s=	1100	f=	4.0	c=	87.9	l=	2.49	j=	0.012
0:'>T'	<=	0:'C>' 1:'G' -6:'A'	s=	722	f=	2.6	c=	83.9	l=	2.38	j=	0.007
0:'C>'	<=	0:'>T' 1:'G' -1:'C'	s=	1100	f=	4.0	c=	83.9	l=	2.64	j=	0.012

-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	Confidence	Freq.	Support	Lift	J-Measure
										C >T	G	A									87.9	4.1	1140	2.49	0.012
									C	C >T	G										87.9	4.0	1100	2.49	0.012
				A						C >T	G										83.9	2.6	722	2.38	0.007
									C	C >T	G										83.9	4.0	1100	2.64	0.012

Figure 1: Example association rules (above) visualised by AsVis (below). No information is lost in the visualisation, but the positional nature of the rules is immediately apparent.

fidence of a rule is the conditional probability of the consequent (e.g. {0: 'C→T'}) given the condition ({+1:'G'}). Lift is simply the ratio of confidence over the relative frequency of the rule consequent, whereas J-measure [Smyth and Goodman, 1992] is an information-theoretic measure describing the amount of information that the condition gives about the consequent.

Association rules are suitable as such for exploratory data analysis, and a large amount of them can be discovered efficiently, without setting a strong focus on any particular attributes. Usually only support and confidence thresholds are used to limit the number of rules, and algorithms such as Apriori [Agrawal *et al.*, 1996] produce all association rules between any sets of attributes that exceed the thresholds. A rule can have any number of conditions and consequents, but with reasonable threshold values, our data did not give any strong associations for complex dependencies with more than one consequent.

To facilitate visual browsing and exploration of thousands of rules, an interactive web interface, AsVis, was developed. AsVis graphically renders the rules into a form that visually reflects the sequential nature of the data, taking advantage of the relatively small number of dimensions (positions). The association rules are listed in a table, where each row represents a single rule, and each column corresponds either to a position relative to the point of mutation or to a strength or generality measure for the rule (see Figure 1). Consequents and conditions are colour-coded in the positional columns.

The interface enables the user to explore the most interesting rules. Clicking on the column header of a measure sorts the rules by that measure, with the best scores at the top. Clicking on a positional column header limits the display to only those rules that have either a condition or a consequent in that position, while keeping the current sorting. This approach is somewhat similar to the TASA system [Klemettinen *et al.*, 1999].

This simple approach enables quick visual scanning of the rules for an overview of dependencies between positions. The measures for each rule are for closer inspection, providing the full scope of information discovered by the data mining process.

Acknowledgements

This research has been supported by Tekes (the National Technology Agency of Finland), The Medical Reseach Fund of Tampere University Hospital, and the Academy of Finland. We thank Adrian Nickson for valuable discussions.

References

[Agrawal *et al.*, 1996] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. pages 307–328, 1996.

[Cooper and Krawczak, 1993] David N. Cooper and Michael Krawczak. Human Gene Mutation. 1993.

[International HapMap Consortium, 2005] The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, 2005.

[Jiricny, 1998] J. Jiricny. Replication errors: cha(lle)nging the genome. 17(22):6427–6436, 1998.

[Klemettinen *et al.*, 1999] M. Klemettinen, Heikki Mannila, and Hannu Toivonen. Interactive exploration of interesting patterns in the telecommunication network alarm sequence analyzer tasa. *Information and Software Technology*, 41:557–567, 1999.

[Smyth and Goodman, 1992] P. Smyth and R. M. Goodman. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316, 1992.

[Wheeler *et al.*, 2005] David L. Wheeler, Tanya Barrett, Dennis A. Benson, Stephen H. Bryant, Kathi Canese, Deanna M. Church, Michael DiCuccio, Ron Edgar, Scott Federhen, Wolfgang Helmberg, David L. Kenton, Oleg Khovayko, David J. Lipman, Thomas L. Madden, Donna R. Maglott, James Ostell, Joan U. Pontius, Kim D. Pruitt, Gregory D. Schuler, Lynn M. Schriml, Edwin Sequeira, Steven T. Sherry, Karl Sirotkin, Grigory Starchenko, Tugba O. Suzek, Roman Tatusov, Tatiana A. Tatusova, Lukas Wagner, and Eugene Yaschenko. Database resources of the National Center for Biotechnology Information. *Nucl. Acids Res.*, 33:D39–45, 2005.

[Zhang and Zhao, 2005] Fengkai Zhang and Zhongming Zhao. SNPNB: analyzing neighboring-nucleotide biases on single nucleotide polymorphisms (SNPs). *Bioinformatics*, 21(10):2517–2519, 2005.

Paper session: *Markov Models*

Prognosis of High-Grade Carcinoid Tumor Patients using Dynamic Limited-Memory Influence Diagrams*

Marcel A.J. van Gerven[1], **Francisco J. Díez**[2], **Babs G. Taal**[3], **Peter J.F. Lucas**[1]
[1]Institute for Computing and Information Sciences, Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
E-mail: marcelge@cs.ru.nl
[2]Department of Artificial Intelligence, UNED, Madrid
[3]Netherlands Cancer Institute/Antoni van Leeuwenhoek hospital, Amsterdam

Abstract

Dynamic limited-memory influence diagrams (DLIMIDs) have been developed as a framework for decision-making under uncertainty over time. We show that DLIMIDs constructed from two-stage temporal LIMIDs can represent infinite-horizon decision processes. Given a treatment strategy supplied by the physician, DLIMIDs may be used as prognostic models. The theory is applied to determine the prognosis of patients that suffer from an aggressive type of neuroendocrine tumor.

1 Introduction

An important task in medicine is making an accurate prognosis for a particular patient given the patient's history. Accurate prognosis facilitates patient feedback and allows the physician to adjust the treatment strategy but is non-trivial in a world that is characterized by change and uncertainty. In our research, we have been engaged in the construction of a prognostic model for high-grade carcinoid tumors of the midgut, which are an aggressive type of neuroendocrine tumor [Modlin *et al.*, 2005]. The model has been constructed in collaboration with an expert physician of the Netherlands Cancer Institute (NKI).

The aim of this paper is to show how prognostic models may be constructed using an approach that is based on *limited-memory influence diagrams* (LIMIDs) [Lauritzen and Nilsson, 2001]. We extend the definition of LIMIDs to *dynamic* LIMIDs, which explicitly take time into account. We show that dynamic LIMIDs allow the handling of infinite-horizon and partially observable Markov decision processes (POMDPs) [Aström, 1965] whenever they are representable as a so-called *two-stage temporal* LIMID (2TLIMID). Infinite-horizon POMDPs cannot be dealt with using standard (limited-memory) influence diagrams, and contrary to POMDPs, the 2TLIMID representation makes explicit a factorization of the state-space that is defined by the variables in the domain[1]. This is advantageous, from a computational point of view, since it allows for more efficient inference algorithms, and also from a representational point of view, since it allows us to describe the model in terms of the relations that hold between domain variables (see e.g. [Peek, 1999]). Given the strategy of a decision maker, a 2TLIMID can be transformed into a two-stage temporal Bayes network [Dean and Kanazawa, 1989], and prognosis then proceeds by means of probabilistic inference using this Bayesian network.

In contrast to classical approaches to prognosis, such as Cox's proportional hazard model [Cox, 1972], we take a *model-based* approach that aims to represent as accurately as possible the causal relations that hold between domain variables. It has been argued that models which capture cause-effect relationships are more meaningful, accessible and reliable than models which capture empirical associations [Druzdzel, 1997]. Causal models are also richer in representational power than non-causal models, since they allow for reasoning under interventions [Pearl, 2000]. In the context of decision support in medicine, causal models have several advantages. They allow for capturing expert knowledge, which is a valuable commodity in itself, and are more easily modified when new knowledge becomes available (i.e. they are less *brittle* than models based on empirical associations). Furthermore, they facilitate the explanation of drawn conclusions, which may increase the acceptance of decision-support systems in medicine [Teach and Shortliffe, 1984; Lacave and Díez, 2002]. However, building causal models often proves to be non-trivial, as it is difficult to elicit the needed qualitative and quantitative knowledge.

We proceed as follows. Section 2 describes required preliminaries. Dynamic LIMIDs and 2TLIMIDs are introduced in Section 3. Section 4 presents a formalization of prognosis, where we use 2TLIMIDs to represent prognostic models. Section 5 describes the prognostic model for high-grade carcinoids as an illustration of the theory. Section 6 describes some results concerning prognostic model performance. The paper is concluded in Section 7.

2 Preliminaries

Bayesian networks [Pearl, 1988] provide for a compact factorization of a joint probability distribution of a set of random variables by exploiting the notion of *conditional independence*. One way to represent conditional independence is by means of an acyclic directed graph (ADG) G whose nodes $V(G)$ correspond to random variables $\mathbf{X}$ and

*This research was sponsored by the Dutch Institute Madrid and by the Dutch Science Foundation under grant number 612.066.201.

[1]Much recent POMDP research has been concerned with taking advantage of such factorizations [Boutilier *et al.*, 1996a].

the absence of arcs from the set of arcs $A(G)$ represents conditional independence. Due to this one-to-one correspondence we will use nodes $v \in V(G)$ and random variables $X \in \mathbf{X}$ interchangeably. A *Bayesian network* (BN) is then defined as a pair $\mathcal{B} = (G, P)$, such that the joint probability distribution P is factorized according to G:

$$P(\mathbf{X}) = \prod_{X \in \mathbf{X}} P(X \mid \pi_G(X))$$

where $\pi_G(X)$ denotes the *parents* of $X : \{X' \mid (X', X) \in A(G)\}$. We also say that X is the *child* of some $X' \in \pi(X)$ where we drop the subscript G when clear from context. In this paper, we say that a (random) variable X takes values x from a set Ω_X and use $\mathbf{x}$ to denote an element in $\Omega_{\mathbf{X}} = \times_{X \in \mathbf{X}} \Omega_X$ for a set $\mathbf{X}$ of (random) variables.

Limited-memory influence diagrams are models for decision-making under uncertainty [Lauritzen and Nilsson, 2001]. They generalize standard influence-diagrams (IDs) by relaxing the *no-forgetting* assumption [Howard and Matheson, 1984]. This assumption states that, given a total ordering of the decisions, the information known when making decision D is also available when making decision D', if D precedes D' in the ordering. A *limited-memory influence diagram* (LIMID) is defined as a tuple $\mathcal{L} = (\mathbf{C}, \mathbf{D}, \mathbf{U}, G, P)$. Here, $\mathbf{C}$ is a set of *chance variables* (graphically depicted by circles), which are random variables as in a Bayesian network that represent the stochastic component of the model. $\mathbf{D}$ is a set of *decision variables* (graphically depicted by squares), which take on a value from a set of choices Ω_D that represent the decisions that may be externally manipulated by a decision maker. $\mathbf{U}$ is a set of *utility functions* (graphically depicted by diamonds), which represent the utility of being in a certain state as defined by configurations of chance and decision variables. G is an ADG, where nodes $V(G)$ correspond to $\mathbf{C} \cup \mathbf{D} \cup \mathbf{U}$. Again, due to this correspondence, we will use nodes in $V(G)$ and corresponding elements in $\mathbf{C} \cup \mathbf{D} \cup \mathbf{U}$ interchangeably. P is a family of probability distributions that specifies for each configuration $\mathbf{d} \in \Omega_{\mathbf{D}}$ a distribution:

$$P(\mathbf{C} \colon \mathbf{d}) = \prod_{C \in \mathbf{C}} P(C \mid \pi(C))$$

that represents the distribution over $\mathbf{C}$ when the decision maker has set $\mathbf{D} = \mathbf{d}$ [Cowell *et al.*, 1999]. Hence, $\mathbf{C}$ is not conditioned on $\mathbf{D}$, but rather parameterized by $\mathbf{D}$.

The meaning of an arc $(X, Y) \in A(G)$ is determined by the type of Y. If $Y \in \mathbf{C}$ then the conditional probability distribution associated with Y is conditioned by X, as in a Bayesian network. If $Y \in \mathbf{D}$ then the state of X is available to the decision maker prior to deciding upon Y. If $Y \in \mathbf{U}$ then X takes part in the specification of the utility function Y such that $Y \colon \Omega_{\pi(Y)} \to \mathbb{R}$. Utility nodes cannot have children and the joint utility function $\mathcal{U}$ is assumed to be additively decomposable such that $\mathcal{U} = \sum_{U \in \mathbf{U}} U$.

In contrast to standard influence diagrams, the order in which decisions are made in a LIMID should only be compatible with the partial order induced by G, and making a decision D is based solely on its direct parents $\pi(D)$. A *stochastic policy* for decisions $D \in \mathbf{D}$ is defined as a distribution $P_D(D \mid \pi(D))$ that maps configurations of $\pi(D)$ to a distribution over alternatives for D. If P_D is degenerate (i.e. consisting of ones and zeros only) then we say that the policy is deterministic. Let $\mathbf{V}$ denote $\mathbf{C} \cup \mathbf{D}$. A *strategy* is a set of policies $\Delta = \{P_D \colon D \in \mathbf{D}\}$ which induces the following joint distribution over the variables in $\mathbf{V}$:

$$P_\Delta(\mathbf{V}) = P(\mathbf{C} \colon \mathbf{D}) \prod_{D \in \mathbf{D}} P_D(D \mid \pi(D)).$$

Using this distribution we can compute the expected utility of a strategy Δ as

$$E_\Delta(\mathcal{U}) = \sum_{\mathbf{v}} P_\Delta(\mathbf{v}) \mathcal{U}(\mathbf{v}).$$

The aim of any rational decision maker is then to maximize the expected utility by finding the optimal strategy $\arg\max_\Delta E_\Delta(\mathcal{U})$.

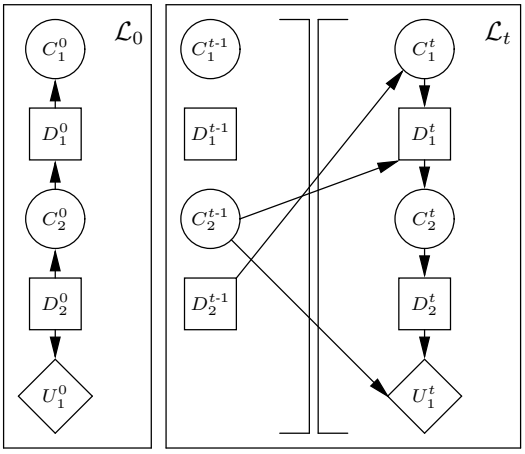


Figure 1: Structure of a 2TLIMID.

3 Dynamic LIMIDs

In this section we demonstrate how to use dynamic LIMIDs that are constructed by means of a structure that we term a *two-stage temporal LIMID* (2TLIMID). When dealing with time, we use $\mathbf{T} \subseteq \mathbb{N}$ to represent a set of time points, which we normally assume to be an interval $\{u \mid t \leq u \leq t', \{t, u, t'\} \subset \mathbb{N}\}$, also written as $t : t'$. We assume that chance variables, decision variables and utility functions are indexed by a superscript $t \in \mathbf{T}$, and use $\mathbf{C^T}$, $\mathbf{D^T}$ and $\mathbf{U^T}$ to denote all chance variables, decision variables and utility functions at times $t \in \mathbf{T}$, where we abbreviate $\mathbf{C^T} \cup \mathbf{D^T}$ with $\mathbf{V^T}$. If $\mathbf{T} = 0 : n$, where $n \in \{1, 2, \ldots\}$ is the *horizon*, then we suppress $\mathbf{T}$ altogether, and we suppress indices for individual chance variables, decision variables and utility functions when clear from context.

3.1 Constructing Dynamic LIMIDs

A *dynamic LIMID* (DLIMID) is simply defined as a LIMID $(\mathbf{C}, \mathbf{D}, \mathbf{U}, G, P)$ such that for all pairs of variables $X^t, Y^u \in \mathbf{V} \cup \mathbf{U}$ it holds that if $t < u$ then Y^u cannot precede X^t in the partial ordering that is induced by G. If the first-order Markov assumption holds that the future is conditionally independent of the past given the present, then we can define a DLIMID in terms of a two-stage temporal LIMID (Fig. 1).

Definition 3.1. *A* two-stage temporal LIMID *(*2TLIMID*) is a pair* $(\mathcal{L}_0, \mathcal{L}_t)$ *with* prior model $\mathcal{L}_0 = (\mathbf{C}^0, \mathbf{D}^0, \mathbf{U}^0, G^0, P^0)$ *and* transition model $\mathcal{L}_t = (\mathbf{C}^{t-1:t}, \mathbf{D}^{t-1:t}, \mathbf{U}^t, G, P)$ *such that chance and decision variables* V_i^{t-1} *in* $\mathbf{V}^{t-1}$ *have no parents.*

The prior model is used to represent the initial distribution $P^0(\mathbf{C}^0 : \mathbf{D}^0)$ and utility functions $U \in \mathbf{U}^0$. The transition model is not yet bound to any specific t, but if bound to some $t \in 1 : n$, then it is used to represent the conditional distribution $P(\mathbf{C}^t : \mathbf{D}^{t-1:t})$ and utility functions $U \in \mathbf{U}^t$ where both G and P do not depend on t. We define the *interface* of the transition model as the set $\mathbf{I}^t \subseteq \mathbf{V}^{t-1}$ such that $(V_i^{t-1}, V_j^t) \in A(G) \Leftrightarrow V_i^{t-1} \in \mathbf{I}^t$.

Given a horizon n, we may *unroll* a 2TLIMID for n *time-slices* in order to obtain a DLIMID such that we obtain the following joint distribution:

$$P(\mathbf{C}, \mathbf{D}) = P^0(\mathbf{C}^0 : \mathbf{D}^0) \prod_{t=1}^{n} P(\mathbf{C}^t : \mathbf{D}^{t-1:t}). \quad (1)$$

Let $\Delta^t = \{P_D(D \mid \pi_G(D)) \mid D \in \mathbf{D}^t\}$ be the strategy for time t and $\Delta = \Delta^0 \cup \cdots \cup \Delta^n$. Given a strategy Δ^0, $\mathcal{L}_0$ defines the following distribution over variables in $\mathbf{V}^0$:

$$P_{\Delta^0}(\mathbf{V}^0) = P^0(\mathbf{C}^0 : \mathbf{D}^0) \prod_{D \in \mathbf{D}^0} P_D(D \mid \pi_{G^0}(D)).$$

Likewise, given a strategy Δ^t with $t > 0$, $\mathcal{L}_t$ defines the following conditional distribution over variables in $\mathbf{V}^t$:

$$P_{\Delta^t}(\mathbf{V}^t \mid \mathbf{V}^{t\text{-}1}) = P(\mathbf{C}^t : \mathbf{D}^{t:t-1}) \prod_{D \in \mathbf{D}^t} P_D(D \mid \pi_G(D)).$$

Combining these equations, given a horizon n and strategy Δ, a 2TLIMID induces the following distribution over variables in $\mathbf{V}$:

$$P_\Delta(\mathbf{V}) = P_{\Delta^0}(\mathbf{V}^0) \prod_{t=1}^{n} P_{\Delta^t}(\mathbf{V}^t \mid \mathbf{I}^t). \quad (2)$$

Let $\mathcal{U}^0(\mathbf{V}^0) = \sum_{U \in \mathbf{U}^0} U(\pi_{G^0}(U))$ stand for the joint utility for $t = 0$ and let $\mathcal{U}^t(\mathbf{V}^{t-1:t}) = \sum_{U \in \mathbf{U}^t} U(\pi_G(U))$ denote the joint utility for time-slice $t > 0$. We redefine the joint utility function for a dynamic LIMID as

$$\mathcal{U}(\mathbf{V}) = \mathcal{U}^0(\mathbf{V}^0) + \sum_{t=1}^{n} \gamma^t \mathcal{U}^t(\mathbf{V}^{t-1:t})$$

where γ, with $0 \leq \gamma < 1$, is a *discount factor*, representing the notion that early rewards are worth more than the same rewards earned later in time.

3.2 Representing Observed History

It is clear from Eq. 1 that DLIMIDs constructed from a 2TLIMID take into account *at most* all chance and decision variables in two subsequent time-slices, since $\pi(D_i^0) \subseteq \mathbf{V}^0$ and $\pi(D_i^t) \subseteq \mathbf{V}^{t-1:t}$. Observations made earlier in time are not taken into account and as a result, states that are qualitatively different can appear the same to the decision maker, leading to suboptimal policies. This phenomenon is known as *perceptual aliasing* [Whitehead and Ballard, 1991]. In this paper we use *memory variables* to take into account (part of) the observed history $\mathbf{v}'$ with $\mathbf{V}' \subseteq \mathbf{V}^{0:c}$ and current time c, as depicted in Fig. 2.

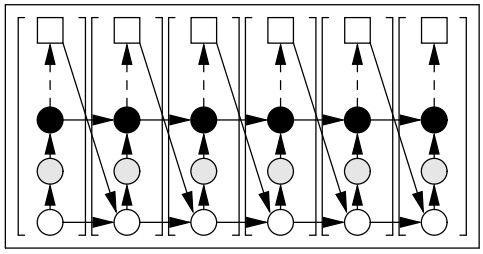


Figure 2: Dealing with perceptual aliasing by introducing memory variables (black circles). Memory variables are used instead of associated observed variables (shaded circles) as the informational predecessor for a decision variable (squares).

Note that if we represent the full observed history, inference becomes intractable for long histories since the states of a memory variable $M \in \mathbf{C}$ associated with a variable $V \in \mathbf{V}$ are given by Ω_M^n, where $\Omega_M^{j+1} = \Omega_M^j \cup (\Omega_M^j \times \Omega_V)$ and $\Omega_M^0 = \Omega_V$. However, by restricting the length of the observed history and/or by using *aggregation* techniques [Boutilier *et al.*, 1996a] that group states which are indistinguishable from the point of view of the decision maker, we can both use the limited-memory assumption of LIMIDs and deal with perceptual aliasing[2]. Examples of variables that fulfill the role of memory variable are BMDHIST and TREATHIST in Fig. 3, which maintain information regarding complications and previous treatments respectively. An additional advantage of the use of memory variables is the fact that we retain the first-order Markov assumption. Due to this property DLIMIDs can take benefit from efficient algorithms for probabilistic inference.

3.3 Inference using 2TLIMIDs

To perform inference with a LIMID $\mathcal{L} = (\mathbf{C}, \mathbf{D}, \mathbf{U}, G, P)$ given a strategy Δ, we convert $\mathcal{L}$ into a Bayesian network $\mathcal{B} = (G', P')$ that is subsequently used for inference purposes. As has been remarked, a strategy Δ induces a distribution over variables $\mathbf{V}$ (viz. Eq. 2). Hence, given Δ, we may convert decision variables D into random variables X_D with parents $\pi_G(D)$ such that

$$P'(X_D \mid \pi_{G'}(X_D)) = P_D(D \mid \pi_G(D)).$$

Additionally, it is possible to convert utility functions U into random variables X_U. Let $\pi_{G'}(X_U) = \pi_G(U)$ where $\Omega_{X_U} = \{0, 1\}$. We associate $P'(X_U \mid \pi(X_U))$ with X_U by means of a transformation

$$P'(X_U = 1 \mid \mathbf{x}') = \frac{U(\mathbf{x}') - \min_{\mathbf{x}} U(\mathbf{x})}{\max_{\mathbf{x}} U(\mathbf{x}) - \min_{\mathbf{x}} U(\mathbf{x})}$$

with $\mathbf{x}, \mathbf{x}' \in \Omega_{\pi(U)}$, as defined in [Cooper, 1988]. We use $B(\mathcal{L}, \Delta)$ to denote this transformation. If the strategy is stationary for each time-slice $t \in \{1, \ldots, n\}$ then we can apply the transformation to a 2TLIMID $(\mathcal{L}_0, \mathcal{L}_t)$, to obtain a so-called *two-stage temporal Bayes network*

[2] In the context of POMDPs, methods that rely on the use of a finite history are common [Aberdeen, 2003].

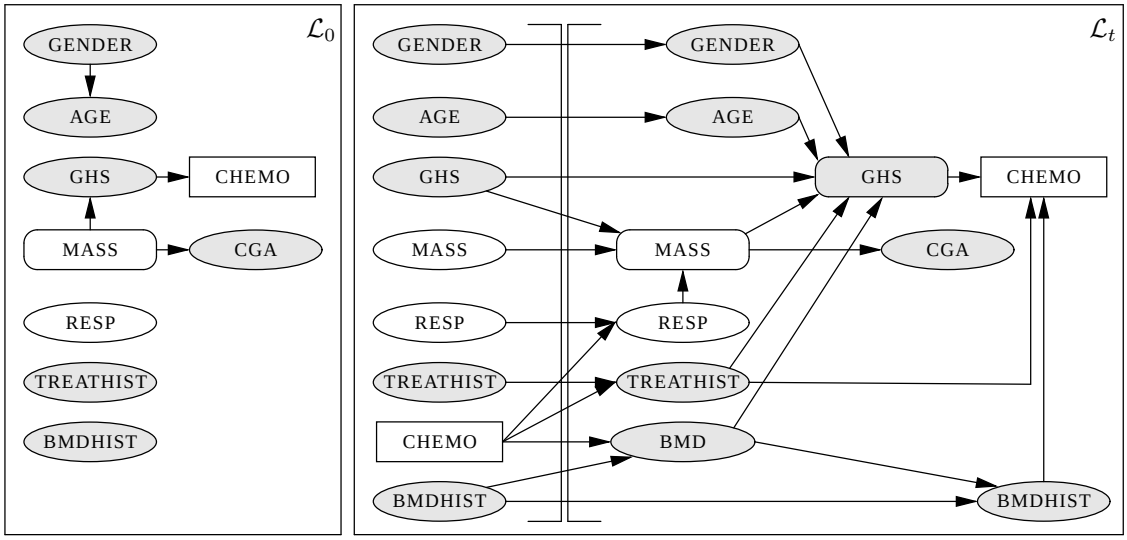


Figure 3: The prognostic model, where shaded nodes are observed and rounded rectangles denote internal structure.

(TBN) $(\mathcal{B}_0, \mathcal{B}_t)$ that is often used to construct a *dynamic Bayesian network* or DBN [Dean and Kanazawa, 1989; Boutilier *et al.*, 1996a; Peek, 1999]. For online inference, efficient algorithms exist that exploit the structure of a 2TBN. In our work, we have used the *interface algorithm* [Murphy, 2002], which allows for online filtering, where the space and time taken to compute $P(\mathbf{X}^t \mid \mathbf{X}^{t-1})$ is independent of the number of time-slices.

4 Prognosis with 2TLIMIDs

Informally, we interpret prognosis as *the prediction of the future status of the patient given the patient history, conditional on a treatment strategy*. This is a non-trivial task since the physician often has incomplete information upon which to base treatment and treatment itself can have uncertain effects. Let $\mathbf{C}$ and $\mathbf{D}$ be sets of chance and decision variables respectively. Let $\mathbf{o}^{0:c}$ with $\mathbf{O}^t \subseteq \mathbf{C}^t$, $t \in 0:c$, represent the observed evidence until the *current time* c and let n denote the horizon. We use the *query variable* $Q \subseteq \mathbf{C} \cup \mathbf{D}$ to denote the variable of interest, and define prognosis given a 2TLIMID as follows:

Definition 4.1. *A* prognosis *for a query variable Q and a horizon n is a conditional probability distribution $P_\Delta(Q^{c:n} \mid \mathbf{o}^{0:c})$ over $Q^{c:n}$.*

In order to compute $P_\Delta(Q^{c:n} \mid \mathbf{o}^{0:c})$, we assume that the prognostic model is defined by $\big((\mathcal{L}_0, \mathcal{L}_t), (\Delta^0, \Delta^t)\big)$, where $(\mathcal{L}_0, \mathcal{L}_t)$ is a 2TLIMID and (Δ^0, Δ^t) is a pair of strategies. Prognosis then proceeds as follows:

1. Define $\big((\mathcal{L}_0, \mathcal{L}_t), (\Delta^0, \Delta^t)\big)$.
2. Create $(\mathcal{B}_0, \mathcal{B}_t) = \big(B(\mathcal{L}_0, \Delta^0), B(\mathcal{L}_t, \Delta^t)\big)$.
3. Recursively compute $P_\Delta(Q^{c:n} \mid \mathbf{o}^{0:c})$ using $(\mathcal{B}_0, \mathcal{B}_t)$.

Although the processes we consider in medicine are finite since they are bounded by patient's life-span, we describe them as infinite-horizon processes where the process has some probability of terminating at each time-slice. In computing the prognosis however we assume that the horizon n is finite. In the next section we develop the actual model for prognosis of high-grade carcinoid tumor patients using the theory developed so far.

5 The High-Grade Carcinoid Model

A carcinoid tumor is a type of neuroendocrine tumor that is predominantly found in the midgut and is normally characterized by the production of excessive amounts of biochemically active substances, such as serotonin [Modlin *et al.*, 2005]. In a small minority of cases, tumors are of high-grade histology which, although biochemically much less active than low-grade carcinoids, show much more rapid tumor progression. Therefore, aggressive chemotherapy in the form of an etoposide and cisplatin-containing scheme is the only treatment option [Moertel *et al.*, 1991]. In this section we develop the prognostic model for high-grade carcinoid tumors, consisting of a 2TLIMID $(\mathcal{L}_0, \mathcal{L}_t)$ and a strategy (Δ^0, Δ^t), supplied by the physician. Patients are admitted to the hospital at the initial time $t = 0$. Each time-slice represents the patient status at three-month intervals since patients return for follow-up every three months. Since the aim is not to improve upon the provided strategy, we omit utility nodes from the discussion.

The qualitative structure of the 2TLIMID that resulted from our modeling efforts is depicted in Fig. 3. The patient's *general health status* (GHS) is of central importance. In oncology, one way to represent the general health status is by means of the *performance status* [Oken *et al.*, 1982]. We define $\Omega_{\text{GHS}} = \{0, \ldots, 5\}$ where $\text{GHS} = 0$ stands for normal health status, $\text{GHS} = 1$ stands for mild complaints, $\text{GHS} = 2$ stands for impaired age-appropriate activity, $\text{GHS} = 3$ stands for confinement to bed for more than 50% of the time, $\text{GHS} = 4$ stands for intensive care and $\text{GHS} = 5$ stands for patient death. The general health status depends on patient properties such as AGE, GENDER and current general health status. Furthermore, GHS is influenced by the tumor mass (MASS) and the treatment policy

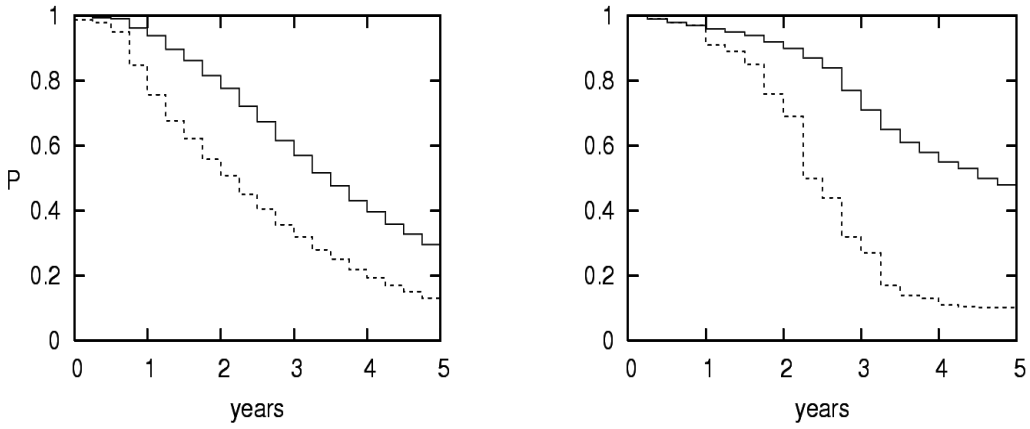


Figure 4: Kaplan-Meier curve, showing the cumulative probability of survival for patients A (dashed line) and B (solid line) over a five year period, as predicted by the model (left), and the physician (right).

that is adopted. Chemotherapy (CHEMO), with $\Omega_{\text{CHEMO}} = \{\textit{none}, \textit{reduced}, \textit{standard}\}$, is the only available treatment, where a reduced dose is at 75% of the standard dose. Chemotherapy can have both positive and negative effects on general health status; positive due to reductions in tumor mass, and negative due to severe bone-marrow depression (BMD) and damage associated with prolonged chemotherapy. We use BMDHIST, with $\Omega_{\text{BMDHIST}} = \{\textit{no-bmd}, \textit{bmd}\}$, as a memory variable to represent whether or not the patient has experienced BMD in the past. Severe BMD is assumed to be fully observable since patients are always tested for it. We use TREATHIST, with $\Omega_{\text{TREATHIST}} = \{0, 1, 2, 3\}$, as a memory variable to represent the patient's relevant treatment history, such that $\text{TREATHIST} = i$ represents continued chemotherapy over the past $3 \cdot i$ months. Reductions in tumor mass due to chemotherapy are often described in terms of tumor response (RESP).

The amount of tumor mass can be estimated by measuring the plasma *chromogranin A* level (CGA) since it is strongly correlated with tumor burden [Nobels *et al.*, 1998]. Since CGA levels are always measured we need not include the decision variable whether or not to determine CGA levels (i.e., the associated policy is *blind*). Severe bone-marrow depression may cause patient death due to associated sepsis and/or internal bleeding [Moertel *et al.*, 1991]. AGE and GENDER are risk factors that may lead to patient death due to causes other than the disease. MASS and GHS in Fig. 3 are compact representations of a Bayesian network fragment. This representation has the advantage of preventing unnecessary clutter in the graphical representation of a Bayesian network and provides a way to represent *context-specific independence* [Boutilier *et al.*, 1996b]. Due to space restrictions, we will not discuss the internal structure of these fragments.

To complete the model, we have to choose a treatment strategy and assess the probabilities that parameterize the model. We mention only the chosen treatment strategy. In $\mathcal{L}_0$, $\pi(\text{CHEMO}^0) = \{\text{GHS}^0\}$, whereas in $\mathcal{L}_t$, $\pi(\text{CHEMO}^t) = \{\text{TREATHIST}^t, \text{BMD}^t, \text{GHS}^t\}$. The policy for chemotherapy in Δ^0 is to apply standard chemotherapy only if the general health status is good enough ($\text{GHS}^0 \leq 3$); otherwise no chemotherapy is applied. The policy used in Δ^t is as follows:

$$(\text{TREATHIST}^t = 0 \wedge \text{GHS}^t \leq 3 \wedge \text{BMDHIST}^t = x) \vee$$
$$(\text{TREATHIST}^t = 1 \wedge \text{GHS}^t < 3 \wedge \text{BMDHIST}^t = x)$$
$$\rightarrow \text{CHEMO}^t = y$$

where $x = \textit{no-bmd} \Leftrightarrow y = \textit{standard}$ and $x = \textit{bmd} \Leftrightarrow y = \textit{reduced}$. In all other cases, we do not give chemotherapy.

6 Experimental Results

In this section we use the prognostic model to answer the following query:

> *What is the probability of patient survival over the next five years?*

We assume that the current time $c = 0$ and compare the prognosis for the following two patients. Patient A is a 75 year old male of poor general health status ($\text{GHS}^0 = 2$) and an initially extreme CGA level. Patient B is a 50 year old female of average general health status ($\text{GHS}^0 = 0$) and an initially elevated CGA level.

In order to compute the probability of patient survival (Q) over the next five years, we assume that $Q \in \mathbf{C}$ with $\Omega_Q = \{\textit{alive}, \textit{dead}\}$, where GHS is a parent of Q, such that $P(Q^t = alive \mid \text{GHS}^t = x)$ is one if $x \neq 5$ and zero otherwise for $0 \leq t \leq n$. We have compared the prognosis made by the model with the prognosis made by the physician, as is shown in Fig. 5.

The physician felt that model predictions where somewhat too positive for patient A, whereas they where somewhat too negative for patient B. Of course, it is difficult to decide how the model would perform in clinical practice, since the physician's opinion is not necessarily the gold standard with which to compare performance. Furthermore, according to the physician, the predictions made by the model do make sense from a qualitative point of view in that it reflects a much worse prognosis for patient A than for patient B. The evaluation and possible calibration of the model in a clinical setting deserves further attention.

7 Conclusion

We have defined DLIMIDs constructed from 2TLIMIDs as a framework for decision-making under uncertainty and used them as the basis for a prognostic model for high-grade carcinoid patients. Although the repetitive structure of a 2TLIMID has been used implicitly in [Lauritzen and Nilsson, 2001], the explicit use of a 2TLIMID and its transformation to a 2TBN allows for the representation of infinite-horizon POMDPs. This benefit comes at the expense of using policies that may suffer from perceptual aliasing. This is resolved by means of memory variables which represent the observed history that is considered relevant by the physician. This approach is particularly useful whenever the policy depends on a small subset of the observed history, as is for instance dictated by a treatment protocol. In general, we would also like to use 2TLIMIDs in order to improve strategies for infinite-horizon partially-observable Markov decision processes, which is a research topic we are currently pursuing. The advocated model-based approach allows for a computationally efficient prognostic model that facilitates interpretation by the physician, while the experimental results demonstrate the feasibility of our approach to prognosis in medicine.

References

[Aberdeen, 2003] D. Aberdeen. A (revised) survey of approximate methods for solving partially observable markov decision processes. Technical report, National ICT Australia, Canberra, Austalia, 2003.

[Aström, 1965] K.J. Aström. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

[Boutilier *et al.*, 1996a] C. Boutilier, T. Dean, and S. Hanks. Planning under uncertainty: structural assumptions and computational leverage. In M. Ghallab and A. Milani, editors, *New Directions in AI Planning*, pages 157–171. IOS Press, Amsterdam, 1996.

[Boutilier *et al.*, 1996b] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 115–123, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[Cooper, 1988] G.F. Cooper. A method for using belief networks as influence diagrams. In *Proceedings of the 4th Workshop on Uncertainty in AI*, pages 55–63, University of Minnesota, Minneapolis, 1988.

[Cowell *et al.*, 1999] R. Cowell, A. P. Dawid, S. L. Lauritzen, and D. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[Cox, 1972] D.R. Cox. Regression models and life tables (with discussion). *Journal of the Royal Statistical Society, Series B*, 34:197–220, 1972.

[Dean and Kanazawa, 1989] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[Druzdzel, 1997] M.J. Druzdzel. Five useful properties of probabilistic knowledge representations from the point of view of intelligent systems. *Fundamenta Informaticae*, 30(3–4):241–254, 1997.

[Howard and Matheson, 1984] R.A. Howard and J.E. Matheson. Influence diagrams. In R.A. Howard and J.E. Matheson, editors, *Readings in the Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA, 1984.

[Lacave and Díez, 2002] C. Lacave and F.J. Díez. A review of explanation methods for Bayesian networks. *Knowledge Engineering Review*, 17(2):107–127, 2002.

[Lauritzen and Nilsson, 2001] S.L. Lauritzen and D. Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251, 2001.

[Modlin *et al.*, 2005] I.M. Modlin, M. Kidd, I. Latich, M.N. Zikusoka, and M.D. Shapiro. Current status of gastrointestinal carcinoids. *Gastroenterology*, 128:1717–1751, 2005.

[Moertel *et al.*, 1991] C.G. Moertel, L.K. Kvols, M.J. O'Connell, and J. Rubin. Treatment of neuroendocrine carcinomas with combined etoposide and cisplatin. Evidence of major therapeutic activity in the anaplastic variants of these neoplasms. *Cancer*, 68(2):227–232, 1991.

[Murphy, 2002] K.P. Murphy. *Dynamic Bayesian Networks*. PhD thesis, UC Berkely, 2002.

[Nobels *et al.*, 1998] F.R.E. Nobels, D.J. Kwekkeboom, R. Bouillon, and S.W.J. Lamberts. Chromogranin A: Its clinical values as marker of endocrine tumours. *European Journal of Clinical Investigation*, 28:431–440, 1998.

[Oken *et al.*, 1982] M.M. Oken, R.H. Creech, D.C. Tormey, J. Horton, T.E. Davis, E.T. McFadden, and P.P. Carbone. Toxicity and response criteria of the eastern cooperative oncology group. *American Journal of Clinical Oncology*, 5:649–655, 1982.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 2 edition, 1988.

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

[Peek, 1999] N.B. Peek. Explicit temporal models for decision-theoretic planning of clinical management. *Artificial Intelligence in Medicine*, 15:135–154, 1999.

[Teach and Shortliffe, 1984] R.L. Teach and E.H. Shortliffe. An analysis of physicians' attitudes. In B.G. Buchanan and E.H. Shortliffe, editors, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, Mass, 1984.

[Whitehead and Ballard, 1991] S.D. Whitehead and D.H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83, 1991.

A Dynamic Model for Therapy Selection in ICU Patients with VAP

Theodore Charitos[1], **Stefan Visscher**[2],
Linda C. van der Gaag[1], **Peter Lucas**[3], and **Karin Schurink**[2]

[1] Dept. of Inform. and Comp. Sciences, Utrecht University,The Netherlands
[2] Dept. of Internal Medic. and Infect. Diseases, University Medical Center Utrecht, The Netherlands
[3] Inst. for Comp. and Inform. Sciences, Radboud University, Nijmegen, The Netherlands

Abstract

Treating ventilator-associated pneumonia in mechanically ventilated patients in intensive care units is seen as a clinical challenge. In this paper, we develop a dynamic-decision model that explicitly captures the development of the disease over time. To represent the dependencies between the variables involved in a compact way we use a dynamic Bayesian network and combine it with the framework of partially observable Markov decision processes to choose optimal antimicrobial therapy for respiratory tract infections. We discuss implementation issues and modelling advantages of our model and demonstrate its use for a number of real patients.

1 Introduction

Many patients admitted to an intensive care unit (ICU) need respiratory support by a mechanical ventilator, which promotes the development of ventilator-associated pneumonia (VAP) in these patients. Effective and fast treatment of VAP is seen as an issue of major significance. The difficulty in diagnosing VAP is in the lack of an accurate, non-invasive (that is, patient-friendly) gold standard; VAP is therefore diagnosed by taking a number of different clinical features into account [9; 14].

A prominent role in the development of VAP is played by two stochastic processes: *colonisation* of the laryngotracheobronchial tree by pathogens and the onset and development of *pneumonia*. A dynamic Bayesian network, called dVAP was developed that explicitly captures the temporal relationships between the variables involved [5]. This network takes into account the patient's characteristics from earlier days when performing diagnosis. The numerical part of the network was constructed from estimations by infectious-disease experts and from the literature. In a later stage these probabilities were updated through machine learning using collected patient data, which resulted in a better diagnostic performance of the model.

The treatment of VAP is seen as a significant problem by ICU doctors. Firstly, many of the patients suffering from VAP are severely ill. Secondly, the presence of multi-resistant bacteria in clinical wards, in particular the ICU, makes prescription of antibiotics with a spectrum as narrow as possible essential; the description of broad-spectrum antibiotics promotes the development of antimicrobial resistance, and should therefore be avoided when possible. In this paper, we address optimal therapy selection using the dVAP model. For this purpose, we focus on the framework of partially observable Markov decision processes (POMDPs) [1; 7; 12; 15] for sequential decision making.

Although the standard POMDP framework in essence allows us to capture the main elements of choosing a therapy of VAP, it cannot be used directly, mainly because: (1) the number of parameters required can be huge, and (2) exact methods for solving the problem are computationally very demanding and only small problems can be solved exactly. In view of these considerations, we extend the dVAP network and construct a dynamic-decision model that incorporates the uncertainty included in the treatment procedure. We then use the Perseus algorithm for its evaluation [16]. Perseus is a point-based approximate value-iteration algorithm for POMDPs that achieves competitive performance both in terms of solution and speed comparing to alternative (and more complex) algorithms in the literature [3]. Perseus can moreover be easily implemented in practice [13]. Perseus, however, is designed for problems without any structure among the variables representing the state of the process. We enhance the applicability of Perseus for our structured domain to take advantage of the factorisations and independencies among the variables included in the dVAP model.

We tested the resulting model on a group of patients drawn from the files of the ICU of the University Medical Center Utrecht in the Netherlands. The solutions obtained indicate that our dynamic-decision model provides a useful framework for solving and analysing complex decision problems. Our results in fact advocate further application of Perseus in structured domains of other medical therapy problems.

The remainder of this paper is organised as follows. In Section 2, we describe the dVAP network for the diagnosis of VAP. In Section 3 we describe the basics of the POMDP framework and of the Perseus algorithm; in Section 4 we discuss modelling and computational issues related to applying Perseus to decision making for patients with VAP. Section 5 presents and discusses the results from an evaluation study. Finally, the paper ends with our conclusions in Section 6.

2 Diagnosing VAP

We begin by discussing the pathophysiology of VAP and then describe the dVAP model that captures the development of VAP.

2.1 Pathophysiology of VAP

Ventilator-associated pneumonia is, when looking at a daily level, a low-prevalence disease occurring in mechanically-ventilated patients in critical care and involves infection of the lower respiratory tract [2]. In contrast to infections of more frequently involved organs (such as the urinary tract), for which mortality is low, ranging from 1 to 4%, the mortality rate for VAP ranges from 24 to 50% and can reach 76% for some high-risk pathogens. Variables that change due to the development of VAP, among others, are an increased *body temperature*, an abnormal amount of coloured *sputum*, *signs* on the chest X-ray, the duration of *mechanical ventilation*, and an abnormal number of *leukocytes*.

The relationship between the *colonisation* by pathogens and the development of *pneumonia* is captured as follows. Periodically, a sample of the patient's sputum is cultured at the laboratory. When the culture shows a number of colonies of a particular bacterium that is above a particular threshold, the patient is said to be colonised by this bacterium. The seven groups of microorganisms that occur most frequently in critically ill patients and cause colonisation, are modelled in the *diagnostic part* of the network. Information about which bacterium or bacteria are currently present in a patient and the current signs and symptoms constitute the basis for choosing optimal antimicrobial treatment.

2.2 A dynamic model for diagnosis

A *dynamic Bayesian network* (DBN) is a graphical model that encodes a joint probability distribution on a set of stochastic variables, explicitly capturing the temporal relationships between them. More formally, let $\mathcal{V}_n = (V_n^1, \ldots, V_n^m), m \geq 1$, denote the set of variables at time n. Then, a dynamic Bayesian network is a tuple (B_1, B_2), where B_1 is a Bayesian network that represents the prior distribution for the variables in the first time slice $\mathcal{V}_1$, and B_2 defines the transitional relationships between the variables for two consecutive time slices, so that for every $n \geq 2$

$$p(\mathcal{V}_n \mid \mathcal{V}_{n-1}) = \prod_{i=1}^{m} p(V_n^i \mid \pi(V_n^i))$$

where $\pi(V_n^i)$ denotes the set of parents of V_n^i, for $i = 1, \ldots, m$.

DBNs are usually assumed to be time invariant, which means that the topology and the parameters of the network per time slice and across time slices do not change. Moreover, the Markov property for transitional dependence is assumed, which means that $\pi(V_n^i)$ can include variables either from the same time slice n or from the previous slice $n-1$, but not from earlier time slices [10]. Then, by unrolling B_2 for N time slices, a joint probability distribution $p(\mathcal{V}_1, \ldots, \mathcal{V}_N)$ is defined for which the following decomposition property holds:

$$p(\mathcal{V}_1, \ldots, \mathcal{V}_N) = \prod_{n=1}^{N} \prod_{i=1}^{m} p(V_n^i \mid \pi(V_n^i))$$

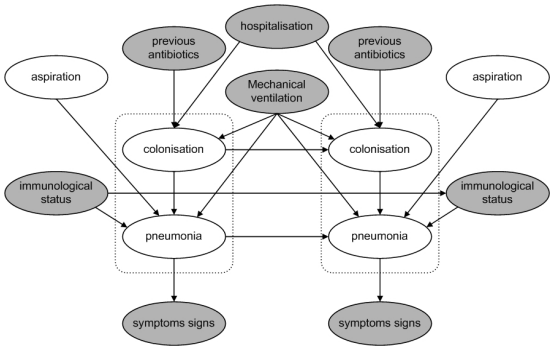


Figure 1: The dVAP network for the diagnosis of VAP; clear variables are hidden, shaded variables are observable. The dashed boxes indicate the hidden processes.

Monitoring in a DBN is the task of computing the probability distribution for a set of variables of interest $\mathcal{X}_n \subset \mathcal{V}_n$ at time n given the observations that are available up to and including time n.

2.3 Modelling and computational issues

An overview of the structure of the dynamic network constructed for the diagnosis of VAP [5] is depicted in Figure 1. The dVAP network includes two interacting dynamic hidden processes, modelled by the compound variables *colonisation* (7 variables) and *pneumonia* (8 variables). The process of colonisation is influenced by three input variables: *hospitalisation*, *mechanical ventilation* and *previous antibiotics*, and one hidden variable *aspiration* that in essence controls its dynamics. We note that the variables *hospitalisation* and *mechanical ventilation* are observed for a period that is longer than the transition interval of the model. The variables thus are modelled as affecting adjacent time slices. The variable *previous antibiotics* represents the effect of previous medication to the patient on the process of colonisation. The symptoms and signs of pneumonia are depicted in Figure 2. These variables are included in the *diagnostic part* of the network.

The practicability of the dVAP network depends to a large extent on the computational burden of inference with the network. For diagnosing patients with VAP, monitoring is performed at each time. For this purpose, the *interface algorithm* can be applied [10]. This algorithm is an extension of the *junction-tree algorithm* for inference with Bayesian networks in general [6] and efficiently exploits the forward interface of a dynamic network. Recall that the forward interface is the set of variables at time slice n that affect some variables at time slice $n+1$ directly. Note that the interface algorithm is linear in the total number of time slices and for large time scopes, the computation time can prove to be prohibitive in practice.

Recent results show that, in case consecutive similar observations are obtained, the probability distribution of the hidden process converges to a limit distribution within a given level of accuracy [4]. After some number of time slices, therefore, there is no need for further inference as long as similar observations are obtained. The phenomenon of consecutive similar observations was evident for several patients in the ICU files. For example, for these patients

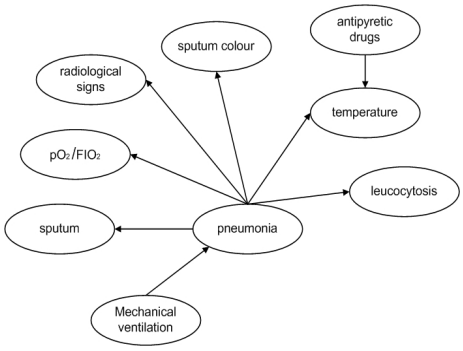


Figure 2: Symptoms and signs of pneumonia.

it was found that the same combination of values was observed for all or almost all of the observable variables for a number of consecutive days. On a set of ICU patients, test results indicated that representing time explicitly and taking into consideration the history of the patient increased diagnostic performance [5].

3 Therapy planning

In this section we describe our approach to solving the dynamic-decision model for patients with VAP. We begin with the theoretical background of POMDPs.

3.1 Basics of POMDPs

Partially Observable Markov Decision Processes (POMDPs) constitute a common framework for decision making about complex dynamic processes where the state of the process cannot be fully observed [1; 7; 15; 16]. A POMDP more specifically describes a stochastic process of which the states are hidden and for which decisions can only be based on observations seen and past actions performed.

Formally, a POMDP is a 6-tuple (S, Θ, A, P, O, R) where S is a finite set of states of the hidden process; Θ is a finite set of observations (findings, results of diagnostic tests); A is a finite set of actions; $P : S \times A \times S \rightarrow [0, 1]$ is a set of *Markovian transition models*, one for each action α, such that $p_\alpha(s' \mid s)$ represents the probability of going from state s to s' with action α; $O : S \times A \times \Theta \rightarrow [0, 1]$ is a set of *observation models*, one for each action α, such that $p_\alpha(o \mid s')$ represents the probability of making observation o after taking action α and transitioning to state s'; and R is a reward function $R : S \times A \times S \times \Theta \rightarrow \mathbb{R}$, such that $R(s, \alpha)$ represents the expected reward received in state s after taking action α.

Given a POMDP, the goal is to construct a *control policy* that maximizes an *objective (value) function*. The objective function combines rewards over multiple time slices, and typically is the expectation of the cumulative sum of rewards r_n at each time n over a *finite-horizon* of N slices, that is $E(\sum_{n=1}^{N} r_n)$, or over a *discounted infinite-horizon*, that is $E(\sum_{n=1}^{\infty} \gamma^n r_n)$, where $0 < \gamma < 1$ is a discount rate. In this paper we focus on the discounted infinite-horizon model as in previous applications of POMDPs in medicine [7].

A belief state b assigns a probability $b(s)$ to every possible state $s \in S$. There thus are an infinite number of possible belief states over S. An optimal policy for b has a *value function* that satisfies the Bellman optimality equation

$$V^*(b) = \max_{\alpha \in A} \left[r(b, \alpha) + \gamma \sum_{o \in \Theta} p(o \mid b, \alpha) V^*(\tau(b, \alpha, o)) \right] \quad (1)$$

where

- $r(b, \alpha) = \sum_{s \in S} b(s) R(s, \alpha)$;
- $p(o \mid b, \alpha) = \sum_{s' \in S} p(o \mid s', \alpha) \sum_{s \in S} p(s' \mid s, \alpha) b(s)$;
- $\tau(b, \alpha, o) \propto p(o \mid s, \alpha) \sum_{s' \in S} p(s \mid s', \alpha) b(s')$;

in which $r(b, \alpha)$ represents the expected reward for a belief state b and current action α, $p(o \mid b, \alpha)$ represents the probability of making observation o one time slice ahead under current action α for a belief state b, and $\tau(b, \alpha, o)$ is the update of the belief state given a previous belief state b and action α, and a current observation o. The optimal policy $\mu^* : b \rightarrow A$ now selects the value-maximizing action

$$\mu^*(b) = \arg\max_{\alpha \in A} \left[r(b, \alpha) + \gamma \sum_{o \in \Theta} p(o \mid b, \alpha) V^*(\tau(b, \alpha, o)) \right]$$

In order to compute the value function $V^*(b)$ in equation (1) we can use the *value iteration algorithm* [15], which guarantees that the sequence of value function approximations V_i defined as

$$V_i(b) = \max_{\alpha \in A} \left[r(b, a) + \gamma \sum_{o \in \Theta} p(o \mid b, \alpha) V_{i-1}(\tau(b, \alpha, o)) \right] \quad (2)$$

converges to the optimal solution. An important property of this approximation sequence is that the value functions $V_i(b)$ in equation (2) are piecewise linear and convex, which allows for computing the update in finite time for the complete belief space [1]. However, the computational cost of doing so is high for all but trivial problems, and thus several methods have been proposed in the literature that try to approximate the optimal value function V^* [8].

3.2 The Perseus algorithm

Perseus is an efficient point based approximate value iteration algorithm for POMDPs [16]. The main idea is to use a set of reachable belief states B that are sampled from the belief simplex to perform value function updates, ensuring that in each iteration the new value function is an upper bound to the previous value function, as estimated on the sampled set of belief states. The intuition behind this approach is that in most practical problems the belief simplex is sparse, in the sense that only a limited number of belief states can ever be reached by letting the hidden process interact with its environment. The algorithm performs value function updates, making sure that in each step the new value function estimate $V_{i+1}(b)$ is an upper bound for $V_i(b)$ for all $b \in B$. The major advantage of Perseus is that in each iteration i it uses only a (random) subset of states in B until the value $V_i(b)$ of every $b \in B$ has improved or remained the same. This property makes the algorithm efficient even in problem domains with large state spaces compared to other approximate methods [8]. We note, however, that the Perseus algorithm has been designed for POMDPs

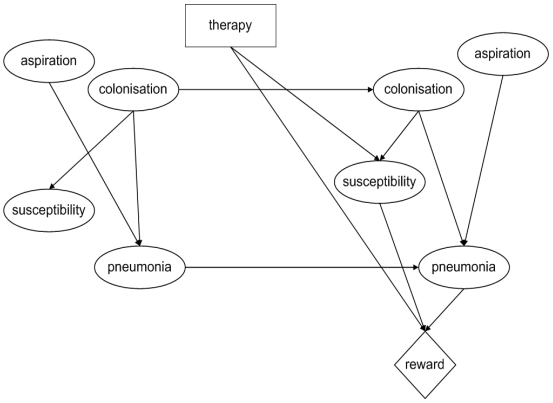


Figure 3: The dynamic-decision network for therapy selection of VAP. Action choice is represented as a rectangle and the reward function as a diamond. The observable variables are excluded for clarity.

with flat belief state, that is, for states without any type of internal structure. To incorporate Perseus into our approach to solving the dynamic-decision model for VAP, we have to enhance its applicability to more structured domains. We discuss such modifications in the next section.

4 Decision making for VAP

The aim of our dynamic-decision model is to aid clinicians in dealing with patients with suspected VAP. Optimal antimicrobial therapy for VAP is selected by balancing the expected efficacy of treatment, which is related to the number of pathogens causing the infection, against the spectrum of antimicrobial treatment. Each of the seven modelled groups of pathogens are susceptible to particular antibiotics. Some of these pathogens are easy to cover. For these pathogens, a narrow or even very narrow antimicrobial spectrum is sufficient. Some pathogens, however, are more difficult to eradicate. Here, we need broader spectrum antibiotics. The problem of prescribing unnecessarily broad-spectrum antibiotics is the occurrence of antibiotic resistance, which means that pathogens are no longer susceptible to a particular antibiotic. Antibiotic resistance is a well-known problem in health-care [2]. Our dynamic-decision model now incorporates the idea of prescribing antibiotic spectra as narrow as possible. The narrower the spectrum, the higher the preference. How well the pathogen is covered by an antibiotic times the preference of the broadness of its spectrum gives the final utility of prescribing this antibiotic [9]. The prescribed treatment thus is a trade-off between maximising coverage and narrowing broadness of spectrum.

To incorporate for decision making in the dVAP model, we add a decision-theoretic part that represents the effect of selected therapy on the probability distribution of VAP. Figure 3 depicts the resulting model. The dynamic-decision model includes the hidden compound variable *susceptibility* (8 variables) that represents the susceptibility of the suspected pathogens to particular antibiotics. A causal independence model, known as the noisy-AND gate [9], is used to model the conjunctive effect of antibiotics on the susceptibility of pathogens. The model thus includes 24 binary hidden variables with 2^{24} possible configurations. The *therapy* variable includes 26 different antibiotics or combinations of antibiotics and the value "none" indicating that the clinician does not prescribe any antibiotic to the patient. These antibiotics have been further classified into four different groups from very narrow to very broad, according to their spectrum. The reward function is thus based on these four spectrum groups and has been assessed by a domain expert [14]. Insight into the potential efficacy of treatment can be obtained by entering symptoms and signs of a patient. In total, the model contains 13 observable variables with 1382400 possible configurations.

At first sight, it seems impossible for Perseus or for any other algorithm to solve our model since both $|S|$ and $|\Theta|$ are extremely large. However, an important feature of our model is that its state and observation sets are not flat, but structured in a factored way. More specifically, the states and the observations of the model are not represented enumeratively but via hidden and observation variables respectively. We further note that although the hidden state of our model consists of 24 variables, only the variables pneumonia and susceptibility are important for decision making. Now, to make efficient use of the Perseus algorithm, we compute the joint probability distribution of just these two variables, which can be done in a similar manner to monitoring in the dVAP model. Our implementation of Perseus in addition takes into account that some variables in the forward interface are observable. Since, for example, *immunological status* is always observed and *colonisation* can be observed for some days, the belief state is modelled as a hybrid state with an observed and a hidden component [7]. Finally, we observe from Figure 2 that we have to consider just six observable variables that are probabilistically affected by pneumonia.

To decrease the computational burden of applying Perseus, we further do not take all observable variables into account when computing the summation in equation (2). That is, upon applying Perseus we sample belief states reflecting realistic data settings only. For example, VAP by definition may be initiated after a patient has been ventilated for more than two days. The state of the mechanical ventilation variable can thus be selected in every iteration of equation (2), from among just the states in which the duration of the ventilation is greater than two days.

As a result of the above modifications, the set Θ includes to 768 possible combinations, and thus is smaller in size than the original set by a factor 1800. The aforementioned considerations were used initially to create a set of reachable belief states B and then to apply Perseus with $\gamma = 0.95$. In our experiments on a 2.4 GHz Intel(R) Pentium computer, creating B took 1.5 seconds per belief state, while computing an optimal policy took approximately one minute using a total of 10000 sampled belief states.

5 Evaluation

We examined the performance of our dynamic-decision model on 5 patients diagnosed with VAP randomly selected from a prospectively collected database of ICU patients. Using the dVAP network we monitored these patients and computed their belief state per day for a total of 10 days.

day	p(VAP)	+ colon.path.	antibiotic
2	0.2295	- -	none meropenem (b)
3	0.0049	Enterobacteria2 -	cotrimoxazol (n) none
4	0.0052	- -	cotrimoxazol (n) none
5	0.0848	- -	cotrimoxazol (n) none
6	0.3401	- -	none cotrimoxazol (n)
7	0.0363	- -	none erythromycin (vn)
8	0.0017	P.aeruginosa Enterobacteria2	cotrimoxazol (n) none
9	0.0012	P.aeruginosa Enterobacteria2	cotrimoxazol (n) none
10	0.0046	- -	cotrimoxazol (n) none

(a) patient A

day	p(VAP)	+ colon.path.	antibiotic
2	0.028	- -	none cotrimoxazol (n)
3	0.0344	- -	none cotrimoxazol (n)
4	0.1905	Enterobacteria2 S.aureus	cotrimoxazol (n) erythromycin (vn)
5	0.5929	- -	cotrimoxazol (n) erythromycin (vn)
6	0.5445	- -	cotrimoxazol (n) erythromycin (vn)
7	0.9823	- -	cotrimoxazol (n) erythromycin (vn)
8	0.9791	Acinetobacter -	ceftazidim (i) aztreonam (i)
9	0.9459	Acinetobacter S.aureus, S.pneumoniae	cotrimoxazol (n) meropenem (b)
10	0.9918	- -	cotrimoxazol (n) meropenem (b)

(b) patient B

Table 1: The best two recommendations (and their spectrum in parenthesis) at each time slice for two patients. Abbreviations for antibiotic spectrum: vn=very narrow; n=narrow; i=intermediate; b=broad.

Contrary to an earlier evaluation of the diagnostic performance of the dVAP network [5], we took into account the sparse colonisation data that existed in the datasets of some patients. In contrast to the data for the observable variables that were readily available, the colonisation data were provided by the laboratory from sputum cultures and took on average 48 hours to become available. Also, these data concerned only a (small) subset of colonisation pathogens and were observed for a few days (maximum 3). To process the colonisation data, we assumed that whenever there was a positive culture for a specific pathogen on a specific day, then the values of the other non-observed pathogens were set to negative. We are aware that this assumption should be used with care. More specifically, the transition matrices estimated by the expert [5] suggested that, under particular conditions, if a pathogen is positive (negative) on one day then it cannot be negative (positive) on the next day. For one patient for example, we noticed upon processing the available data, that on day 8 we assumed the presence of S.aureus and S.pneumoniae to be negative while these two pathogens were actually observed to be positive on day 9. To resolve this issue, we made no assumption about these two pathogens on day 8 and left their value as unobserved.

We compared the recommended decisions from the model with an expert opinion as to the most appropriate antibiotics to cover the likely pathogens. The results were not entirely satisfactory in the opinion of the expert. For one patient, for whom no colonisation data were available, we found that the decisions recommended by the model were acceptable; for two patients the model recommended too broad a spectrum antibiotics, while for the other two patients the recommended antibiotics did not cover the observed pathogens. A possible explanation of this suboptimal performance of the model is that its decisions are strongly affected by the probability of VAP at each time and less by the colonising pathogens; that is, the prescription of antibiotics is heavily dominated by $p(\text{VAP})$, while less weight is given to the presence of colonisation data. For example, if on a specific day a patient has a very small $p(\text{VAP})$ but a positive colonising pathogen, then the model will abstain from prescribing antibiotics and will not use a narrow spectrum antibiotics as would be expected. Another reason is that the influence of the colonisation data on the recommended decision diminished with time according to the specification of the model. More precisely, since colonisation data are sparsely observed, a colonisation pathogen found to be positive on one day will have minor effect on the decision taken two days later because of the Markov assumption underlying the dVAP network.

In view of the above considerations, we enhanced our decision model to incorporate the influence of the colonisation data on the recommended decision in a more appropriate fashion. For each colonisation (group of) pathogen(s) found to be positive on a given day, we force the model to prescribe antibiotics to cover this pathogen. In this way our model considers a conglomeration of different decision plans that are influenced by the presence of positive pathogens in the patient's dataset. To cope with the sparsity of the colonisation data, we use the enhanced model for the following two days as well. As a result, the clinician is presented with a therapy plan that aims to cover positive observed pathogens for at least three days. For the remaining days for which no colonisation data were available, the original decision model was used. The evaluation now showed now that the new recommendations better comply with the expert's recommendations.

We discuss the results for two patients in order to convey how our dynamic-decision model might be employed clinically, and to point out some of its limitations. For patient A, the dVAP network assigns a small probability to VAP

for almost all the days. As a consequence, the decision not to prescribe any antibiotic is always recommended by the model. However, positive cultures of pathogens are observed for the days 3, 8 and 9. For these days (and for the next two days) the antibiotic cotrimoxazol (narrow spectrum) is recommended first. This recommendation reflects the ability of the model to prescribe an antibiotic even if the probability of VAP is very small. We note, however, that on day 2, the model suggests the antibiotic meropenem. This recommendation is far too broad for this patient, and raises the question whether alternative utility models might alleviate this problem. For patient B, the dVAP network assigns quite early (day 5) a relatively high probability to VAP which even further increases in the following days. In addition, positive cultures of pathogens are observed for the days 4, 8 and 9. Our dynamic-decision model takes into account both the high probability of VAP and the positive cultures to recommend appropriate antibiotics that belong to a narrow spectrum whenever possible. This is evident in the recommendations for days 4 to 7, while for days 8 to 10 ther recommendation belongs to the intermediate or broad spectrum. The predictions made and the therapy suggested by the model for both patients are shown in Table 1.

6 Conclusions

We have described the development of a dynamic-decision model that is able to assist clinicians in the clinical management of ventilator-associated pneumonia. For the purpose of computing appropriate decisions from the model, we applied the framework of partially observable Markov decision processes for modelling the action-outcome uncertainty and partial observability. The application and potential of the POMDP framework to medical planning has been discussed in [12] and successfully explored in [7]; in the latter work, a hierarchical Bayesian network was used to represent the disease dynamics and to decrease the computational burden involved. Since exact computation in a POMDP is intractable, we discussed the application of the Perseus algorithm to our problem, in which the belief state of the hidden process is structured. The solutions obtained for a small set of patients from an initial evaluation of our model showed that POMDPs could provide a useful framework for solving complex decision problems. We feel that the promising results justify further refinement and extension of our current model as well as application of our framework to other complex structured decision problems [11].

Acknowledgements

This research was (partly) supported by the Netherlands Organization for Scientific Research (NWO). The first author would like to thank Matthijs Spaan and Nikos Vlassis for their help in implementing Perseus.

References

[1] J. Astrom (1965). Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174-205.

[2] M.J.M. Bonten (2004). Prevention of infection in the intensive care unit. *Current Opinion in Critical Care*, 10(5):364-368.

[3] R. Brafman (2006). Personal communication.

[4] T. Charitos, P. de Waal, and L.C. van der Gaag (2005). Speeding up inference in Markovian models. *Proceedings of the 18th International FLAIRS conference*, pp. 785-790.

[5] T. Charitos, L.C. van der Gaag, S. Visscher, K. Schurink, and P. Lucas (2005). A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in ICU patients. Working notes of the 10th IDAMAP Workshop, pp. 32-37.

[6] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter (1999). *Probabilistic Networks and Expert Systems*. Springer.

[7] M. Hauskrecht and H.S.F. Fraser (2000). Planning treatment of ischemic heart disease with partially observable Markov decision processes. *Artificial Intelligence in Medicine*, 18(3):221-244.

[8] M. Hauskrecht (2000). Value-Function Approximations for Partially Observable Markov Decision Processes. *Journal of Artificial Intelligence Research*, 13: 33-94.

[9] P.J.F. Lucas, N.C de Bruijn, C.A.M Schurink, and A. Hoepelman (2000). A probabilistic and decision theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine*, 19(3):251-279.

[10] K. Murphy (2002). *Dynamic Bayesian networks: Representation, Inference and Learning*. Ph.D. thesis, University of California Berkeley.

[11] P. Poupart (2005). *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Ph.D. thesis, University of Toronto.

[12] N.B. Peek (1999). Explicit temporal models for decision-theoretic planning of clinical management. *Artificial Intelligence in Medicine*, 15(2): 135-154.

[13] Perseus: URL: http://www.science.uva.nl/ mtjspaan/pomdp.

[14] C.A.M. Schurink (2003). *Ventilator Associated Pneumonia: a Diagnostic Challenge*. Ph.D. thesis, Utrecht University.

[15] E.J. Sondik (1978). The optimal control of partially observable Markov decision processes over the infinite horizon: Discounted costs. *Operations Research*, 26:282-304.

[16] M.T.J. Spaan and N. Vlassis (2005). Perseus: Randomized Point-based Value Iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195-220.

Describing scenarios for disease episodes and estimating their probability: a new approach with an application in Intensive Care

Linda Peelen[a], **Niels Peek**[a], **Robert J Bosman**[b]
[a] Department of Medical Informatics, Academic Medical Center - University of Amsterdam
PO Box 22700 1100 DE Amsterdam The Netherlands. Email: {l.m.peelen, n.b.peek} @ amc.uva.nl
[b] Department of Intensive Care, Onze Lieve Vrouwe Gasthuis, Amsterdam, The Netherlands

Abstract

In medical reasoning, disease scenarios are often expressed in abstract terms, e.g., 'multiple organ failure during ICU stay'. In estimating the probability of this type of scenarios from clinical data two problems arise. First, the data are expressed in terms of clinical observations, i.e., low-level data attached to specific time points, whereas scenarios are expressed in terms of high-level concepts, related to time intervals. Second, the amount of data is often too small to directly estimate the probabilities from the data.

This paper provides solutions for these problems. First, the paper introduces a symbolic language to describe multivariate, discrete data measured at a low frequency, and to define abstract scenarios based on these descriptions. Second, the paper proposes a model-based approach to arrive at reliable estimates for the probabilities, using a Markov model and Monte Carlo simulations.The approach is illustrated with an example from the area of Intensive Care.

1 Introduction

Time is an important concept in medicine. The dynamics of a disease, i.e., changes that occur over time in the condition of a patient, give an indication of the progression of the disease. If the condition of a patient worsens, a negative outcome of the disease becomes more likely. Physicians therefore closely monitor changes that occur over time, as these changes may reveal a necessity of changes in treatment or additional examinations of the patient.

The condition of a patient is often described by a combination of variables. Together, the values of these variables at a given time point describe the *state* of a patient's condition, and subsequent states form the *scenario* of the disease.

Because of the tight relationship between changes in the condition of a patient, treatment, and outcome, physicians are interested in the scenarios that are likely to occur, in the patient population in general or in a specific patient group. In medical reasoning the focus is often not on the low-level data observed at specific time points, but on high-level concepts that are related to events during a time interval. E.g., a physician at the Intensive Care Unit (ICU) may not be directly interested in patients who experienced liver failure on the second day of their ICU stay and renal failure on the third, but moreover in patients who suffered from liver failure early in their ICU stay and developed subsequent renal failure.

Providing physicians with estimates of the probability of this type of 'abstract scenarios' poses two problems. First, the low-level data is to be abstracted to a more general level. Most of the temporal abstraction methods that have been developed through the years focus on univariate, high-frequency measurement of continuous data. If the state of the patient is jointly described by multiple, discrete variables, measured at a low frequency, a different type of temporal abstraction is required. Second, when directly estimating the probability of a complex scenario, which consists of a combination of high-level concepts, from data, a large amount of data is necessary, which is often not available in real practice.

This paper provides a solution for these problems. We introduce a symbolic language (based on set-theory) which allows for the construction of high-level concepts by abstracting from multiple, discrete variables, which describe the changes in the condition of a patient. Using this language the knowledge of clinical experts (expressed at a high-level) can be related to clinical observations expressed in terms of low-level data and vice versa. To arrive at reliable estimates of the probability of a specific scenario we propose a relatively simple model-based approach, by constructing a Markov model and subsequently performing Monte Carlo simulations.

Both the language and the approach are illustrated using an example from Intensive Care medicine, an area of medicine in which insight into changes in the state of a patient plays a crucial role and can literally be life-saving.

The paper is organized as follows: Section 2 explains the notation to describe data of this particular type. Section 3 describes the approach to arrive at reliable estimates. In Section 4 we present the case-study that was undertaken using this method in the area of Intensive Care. We discuss our approach and relate it to other work in Section 5.

2 Describing scenarios

In this section we introduce a notation to describe the condition of a patient, and changes therein, both in concrete and abstract terms. The notation is applicable in situations in which the condition of a patient is described by multiple

discrete variables, measured at a low frequency at equidistant time points.

State and time Let $X = \{x_1, \ldots, x_m\}$ be a set of discrete random variables, each with a finite value domain. For convenience, we will assume that $x_i \in \{0, 1\}$ for each $1 \leq i \leq m$, but the theory is easily adapted to more general discrete variables. Formally speaking, the variables in the set X are used to jointly describe the state of a stochastic dynamic system; in our application, this "system" is a patient, and the variables $x_i \in X$ describe different aspects of his or her health condition. To describe temporal progression, we also assume that a discrete and ordered set T of time points is given. It defines the time points where the state of the system is observed; for convenience we will assume that $T = \{1, \ldots, N\}$, $N \in \mathbb{N}^+$.

We note that as the length of the care process varies from patient to patient, T will also vary per patient. The value of N however is equal for all patients. This implies that for a patient who has been observed at time $t > N$, only the values observed at times $t, \ldots, N$ are considered. If N is chosen sufficiently large though, most patients will not reach time $t = N$. In the domain of intensive care for example, if we choose $N = 65$ (days), less than 1 out of 500 patients stays longer, in reality.

To capture the fact that most patients depart from the care process before time point $t = N$ is reached, we require that the set X contains a designated variable, called `exit`, to indicate that the patient leaves the process. There is at most one time point $t \in T$ where $\texttt{exit} = 1$; after that point, all variables from the set X become meaningless. In our application, $\texttt{exit} = 1$ at the day of ICU discharge.

Scenarios To describe state changes of the system over time, we further extend our notation. A *fact* is an expression of the form $x_i(t) = c$, and it denotes that variable $x_i \in X$ takes value $c \in \{0, 1\}$ at time point $t \in T$. A *scenario* is a set of facts that are mutually consistent. That is, two facts in a given scenario cannot refer to the same variable $x_i \in X$ at the same point $t \in T$, otherwise these two facts would contradict each other. We use Φ to denote the set of all possible scenarios. Scenarios can be very general, for instance when they contain just a single fact, or very specific, when they involve a large number of facts. We say that a given scenario $\varphi \in \Phi$ *permits* scenario $\psi \in \Phi$, and write $\varphi \rightsquigarrow \psi$, when $\varphi \cup \psi$ is also a possible scenario (i.e., when the elements of φ and ψ do not contradict each other). Thus, the empty scenario, $\varnothing$, is the most general scenario and permits all scenarios.

As noted before, patients may depart from the care process before the final time point $t \in N$ is reached, and this is indicated by the variable `exit`. Thus far, the definition of our language Φ allows for improper assignments to this variable. We therefore impose restrictions on Φ.

First, we define *open* and *closed* scenarios. A scenario that contains a fact $\{\texttt{exit}(t) = 1\}$ is said to be closed; all other scenarios are called open. Now, let $\mathrm{maxobs}(\varphi)$ be the largest time point t covered by scenario φ, other than through the exit variable. So, scenario φ contains information on the patient's clinical state up to time t, but not thereafter. We distinguish *proper* scenarios from *improper* ones, by defining φ to be a proper scenario when (i) there exists at most one time point t such that φ contains $\{\texttt{exit}(t) = 1\}$, and (ii) if so, then $\mathrm{maxobs}(\varphi) \leq t$. In words, we require that the exit time is unique and exceeds all observation times. We use $\Phi^* \subseteq \Phi$ to denote the set of all proper scenarios. Note that all open scenarios are proper by definition.

We finally distinguish scenarios that describe all state information up to the patient's exit time from those who do not. Let $\varphi \in \Phi^*$ be a proper scenario. Now φ is *complete* when either (i) φ is closed at time point t and contains a fact for each variable $x_i \in X$ at each time point $t' \leq t$, or (ii) φ is open, and contains a fact for each variable $x_i \in X$ at each time point $t'' \in T$. In the second case, the patient left the care process after time $t = N$, but we have complete state information at each of the moments in T.

Abstract scenarios Whereas complete scenarios refer to entire disease histories of individual patients, partial scenarios are more general and may permit many completions. They are therefore associated with groups of patients instead of with individuals. Yet, the language Φ^* still forces us to be highly specific on the facts that constitute partial scenarios: we must specify the exact values and time points that are involved. In a data analysis, however, we are rather interested in identifying groups of patients that share more general characteristics; for instance, our attention could be focused on all ICU patients that experienced multiple organ failure at some time during their stay. This type of circumstance cannot be expressed in our language.

We therefore now introduce the notion of an *abstract scenario*. An abstract scenario σ is a set of (concrete) proper scenarios, each of which is considered a possible realization of the disease process. The elements of σ may be both partial and complete scenarios from Φ^*, and they may, but need not, be mutually exclusive.

As an example, consider the abstract scenario

$$\sigma \quad = \quad \{\{x_i(1) = c\}, \ldots, \{x_i(N) = c\}\}. \qquad (1)$$

The elements of σ are concrete, single-fact scenarios where variable x_i has value c, but each of them refers to a different point in time. We may therefore summarize this scenario as "there was a point in time where $x_i = c$ occurred".

The notion of permittance is easily generalized to abstract scenarios. We say that an abstract scenario σ *permits* concrete scenario ψ (written $\sigma \rightsquigarrow \psi$) when there exists at least one $\varphi \in \sigma$ that permits ψ.

Time and state abstractors To construct abstract scenarios, we will make use of two additional types of variables, *time abstractors* and *state abstractors*. A time abstractor is a variable $\mathbf{t}$ that ranges over the set T of all time points, and can be arbitrarily instantiated with elements from that set. For instance, $\{x_i(\mathbf{t}) = c\}$ covers all concrete scenarios of the form $\{x_i(t) = c\}$, $t \in T$, and is therefore a shorthand nation for the abstract scenario σ of Eq. 1.

A state abstractor is a variable that summarizes particular aspects of the patient's condition, captured by multiple state variables from the set X, and independent of time.

We will often use a time abstractor to express that the state abstraction holds at all time points. For instance, when $x_1, \ldots, x_m$ represent distinct injuries, we may introduce a state abstractor $\mathbf{y}$ that indicates that two or more injuries occur simultaneously. That is, we then define that $\mathbf{y}(\mathbf{t}) = 1$ if and only if the abstract scenario

$$\{\{x_i(\mathbf{t}) = 1,\ x_j(\mathbf{t}) = 1\} \mid 1 \leq i, j \leq m,\ i \neq j\} \quad (2)$$

holds. Again this is merely a shorthand notation for complex expressions. So, we can now write

$$\{\mathbf{y}(\mathbf{t}) = 1,\ x_i(\mathbf{t}) = 0\} \quad (3)$$

to define the abstract scenario where, at some point in time, multiple injuries occur simultaneously, but not involving the one described by x_i.

We finally extend our language with basic arithmetic and comparison operators for time abstractors. In abstract scenarios we allow for the inclusion of additional facts that express relations between time abstractors, using one of the operators '$=$', '$>$', '$\leq$', '$<$', and '$\leq$'. For instance,

$$\{\mathbf{y}_1(\mathbf{t}_1) = 1,\ \mathbf{y}_2(\mathbf{t}_1) = 0,\ \mathbf{y}_2(\mathbf{t}_2) = 1,\ \mathbf{t}_2 > \mathbf{t}_1\} \quad (4)$$

expresses that both abstract states $\mathbf{y}_1$ and $\mathbf{y}_2$ occur, and that the former precedes the latter.

3 Estimating the probability of scenarios

Let $\mathcal{D} = \{\psi_z | z = 1, \ldots, n\}$, be a random sample of complete, proper scenarios, and let σ be an abstract scenario. The set $\mathcal{D}$ could describe, for instance, a retrospective sample of completed disease histories from patients with a given diagnosis, and σ could describe a particular type of malign disease progression for such patients. Then we can estimate the marginal probability of σ in $\mathcal{D}$ as

$$P(\sigma \mid \mathcal{D}) = \frac{1}{n} \sum_{z=1}^{n} I(\sigma \rightsquigarrow \psi_z), \quad (5)$$

where I is the identity function. That is, we estimate $P(\sigma \mid \mathcal{D})$ by counting the number of complete scenarios in $\mathcal{D}$ that are permitted by σ. Similarly, we can estimate the conditional probability that σ occurs, given that a second scenario σ' applies:

$$P(\sigma \mid \sigma', \mathcal{D}) = \frac{\sum_{z=1}^{n} I(\sigma \rightsquigarrow \psi_z)\ I(\sigma' \rightsquigarrow \psi_z)}{\sum_{z=1}^{n} I(\sigma' \rightsquigarrow \psi_z)} \quad (6)$$

For instance, we could be interested in the frequency with which σ occurs among those who die from the disease, and compare it to the frequency where this happens among the survivors.

However, this approach of estimating probabilities directly from the data is not feasible when the second scenario, σ', is rare. In that case, there may be no scenarios in $\mathcal{D}$ that are both permitted by σ and σ'; and even if they exist, the numbers will be too small to make reliable estimates. In summary, a nonparametric estimation approach is feasible for simple, general scenarios that occur often, but not for more complex scenarios with a lower prevalence.

To alleviate this problem, we propose a *model-based* approach for estimating marginal and conditional probabilities of abstract scenarios. The approach consists of two steps: (1) describe the underlying stochastic dynamic system as a Markov model, and (2) derive the probabilities of interest by drawing inferences on this model. Both steps are now described in more detail.

Markov model Let $S_X = \{0, 1\}^m$ denote the set of all possible *states* of the system described by X, i.e. the set of all 2^m possible value assignments to $x_1, \ldots, x_m$. A *transition probability function* for X is a function $f : S_X \times S_X \rightarrow [0, 1]$ where, for each state $s \in S_X$, we have that $\sum_{s' \in S_X} f(s, s') = 1$.

A *Markov model* is now defined as a pair $M = (T, f)$, where f is a transition probability function for X. It assumes that, at each time point $t \in T$, $t > 1$, the system's state is conditionally independent of all earlier states, given the state at time point $t - 1$. The function f describes the conditional probability distribution of state changes at subsequent time points, and is estimated from the dataset $\mathcal{D}$; in principle, it has 2^{2m} parameters (one for each pair of states). As the function itself is independent of time, the underlying Markov process is assumed to be stationary.

In a multivariate Markov model, such as described here, the joint transition probability function f can be described by separate functions $f_1, \ldots f_m$ for each of state variables. Reductions in the number of parameters that need to be estimated can subsequently be obtained by (i) assuming conditional independence among the state variables, and (ii) using a parametric form for these functions. In Section 4 we illustrate this approach by using logistic regression equations for $f_1, \ldots, f_m$.

Inference Once the Markov model has been constructed, we can use it to infer probabilities of interest such as those in Eqs. 5 and 6. Roughly speaking, there are two options for doing this. The first one is to describe the multivariate Markov model as a dynamic Bayesian network, and to use methods for exact probabilistic inference on such networks (e.g., the unrolled junction tree algorithm [Kjærulff, 1995]). However, this would require all state and time abstractors to be modelled explicitly within the network. The complexity of the model and the associated computations would quickly increase when more abstractors are defined.

The second option, which is chosen here, is to use Monte Carlo simulations of the model to randomly generate scenarios, and estimate the probabilities of interest from the resulting simulated data [Robert and Casella, 2004]. When the model M fits well to the original dataset $\mathcal{D}$, the estimated probabilities will approximate the true probabilities that generated $\mathcal{D}$. The procedure to generate scenarios is summarized as follows:

1. sample values for $x_1(1), \ldots, x_m(1)$, and set $t = 1$
2. determine the probability distribution over $x_1(t + 1)$, $\ldots, x_m(t + 1)$ using f
3. sample values for $x_1(t + 1), \ldots, x_m(t + 1)$ based on this distribution, and increment t by one
4. repeat steps 2 and 3 until $\texttt{exit}(t) = 1$ or $t = N$.

We note that steps 2 and 3 can be repeated for a large number of times. At each iteration we test whether the system occupies an exit state, and if the maximum time point $t = N$ is reached. The procedure thus generates scenarios that are both proper and complete. If we choose N to be large, most of of the scenarios will also be closed (i.e., $\texttt{exit}(t) = 1$ before $t > N$).

Two options exist to choose the starting values of a simulated scenario. They can be randomly sampled from scenarios in the original dataset $\mathcal{D}$. This option is used to answer questions of the form 'Which scenarios do frequently occur within the given patient population?'. A second possibility is to use a predefined set of starting values for each simulation. In this case the question that is answered amounts to the form: 'Given that a patient arrives in this particular condition, which scenarios are likely to occur?'.

The entire procedure is repeated a large number of times, n, resulting in a dataset $\mathcal{D}^*$ which contains a large number of generated scenarios. For each scenario in $\mathcal{D}^*$ the relevant time and state abstractors are computed. Based on this data the probabilities described in Eqs. 5 and 6 are calculated.

4 Case-study in intensive care

We have applied this approach in the area of intensive care to investigate changes in organ failure. In this section we first briefly introduce the role of organ failure in the ICU. Then we describe how we applied the approach in this specific situation and the results of these experiments.

4.1 Organ failure at the intensive care unit

A major goal of treatment at the ICU is to stabilize the functioning of organ systems and if necessary to temporarily take over organ function using machinery and medication. Changes in the functioning of specific organ systems are an important indicator of progress of the disease. The development of failure in more than one organ system (*multiple organ failure*, MOF) requires specific attention, as it is a major cause for mortality and morbidity in ICU patients [Bone *et al.* , 1992].

In the ICU organ failure is described on a day-to-day basis using the Sequential Organ Failure Assessment (SOFA) scoring system [Vincent *et al.*, 1998], which consists of six scores that indicate the function of six major organ systems: circulation, respiration, kidney function, central nervous system, the liver and coagulation. For each organ system the degree of organ failure is quantified by an integer value between 0 and 4 (with 0 indicating normal organ function and 4 referring to complete failure). These scores are based on the values of one or two (mostly physiological) variables related to the particular organ system. For example, the score for the hepatic system is determined by the level of bilirubin in the blood. The SOFA scores are measured daily based on the preceding 24 hours of ICU stay.

Although the importance of MOF is well recognized and the SOFA score is routinely collected in more and more ICUs, physicians do not explicitly know which sequences or combinations of organ failure occur more often than others. In this study we use the SOFA score as a basis to discover scenarios of organ failure. We focus on scenarios that describe the relation between MOF and outcome.

4.2 Data

This section gives a brief description of the dataset and indicates which states, abstractor variables and scenarios were used.

Characteristics of the dataset The experiments described in this paper are based on data collected in the ICU of the Onze Lieve Vrouwe Gasthuis (OLVG), an 18 bed mixed medical-surgical ICU in a teaching hospital in Amsterdam. We used data from patients admitted between January 1^{st}, 2002 and December 31^{st}, 2004. We excluded patients admitted after cardiac surgery. For patients who were readmitted to the ICU within the same hospital stay we only used data on their first ICU admission. The dataset contained information on 1508 patients of which 248 (16.4%) died in the ICU. The median length of stay was 2 days, 587 patients were discharged or died after 1 day of ICU stay. In total 6845 records on SOFA scores were available.

States Each of these records describes the state of a particular patient at a given day. The clinical condition of a patient is described by six variables, $\texttt{coag}, \texttt{hepa}, \texttt{circ}, \texttt{neuro}, \texttt{renal}, \texttt{resp}$, describing the functioning of the coagulation, hepatic, circulatory, neurological, renal and respiratory system respectively. In the original data the SOFA scores were measured on a scale ranging from 0 to 4, we dichotomized these values based on the distribution of the scores in the dataset.

Next to these variables which describe the *clinical* aspect of the state of a patient, we use three additional variables that serve as the status indicator: the `exit` variable indicates whether the patient has departed from the care process. To explicitly distinguish between patients who died at the ICU and those who were discharged from the ICU alive, we introduce the variables `ICU death` and `ICU discharge` respectively. If $\texttt{exit} = 0$, these variables equal zero; in case $\texttt{exit} = 1$ exactly one of these variables equals 1.

The nine variables together describe the state of the patient. In total 66 different states are distinguished: 2^6 different states when $\texttt{exit} = 0$ and 2 additional states when $\texttt{exit} = 1$ (either $\texttt{ICU death} = 1$ or $\texttt{ICU discharge} = 1$).

Abstractors and scenarios Based on these variables we define a number of time and state abstractors and scenarios related to MOF and outcome. The state abstractor $\texttt{MOF2}(\mathbf{t})$ expresses whether the patient experienced organ failure in at least 2 organ systems at any time point during ICU stay. In analogy with Eq. 2, $\texttt{MOF2}(\mathbf{t}) = 1$ if the following scenario holds:

$$\{\{x_i(\mathbf{t}) = 1,\ x_j(\mathbf{t}) = 1\}|, i \neq j\}, \tag{7}$$

where $x_i, x_j \in \{\texttt{coag}, \texttt{hepa}, \texttt{circ}, \texttt{neuro}, \texttt{renal}, \texttt{resp}\}$. In a similar fashion we define $\texttt{MOF3}(\mathbf{t})$, $\texttt{MOF4}(\mathbf{t})$, $\texttt{MOF5}(\mathbf{t})$, $\texttt{MOF6}(\mathbf{t})$ which evaluate to 1 if the patient experienced organ failure in at least three, four, five, or in all organ systems respectively. To express the exact amount of organ failure, we combine these abstractors as follows:

$$\texttt{MOFexact2}(\mathbf{t}) = 1 \Leftrightarrow \{\texttt{MOF2}(\mathbf{t}) = 1, \texttt{MOF3}(\mathbf{t}) = 0\}. \tag{8}$$

To arrive at scenarios of interest we can combine scenarios based on these abstractors with concrete scenarios. For example,

$$\{\ \texttt{MOFexact2}(\mathbf{t}) = 1,\ \texttt{hepa}(1) = 1\ \} \tag{9}$$

expresses the scenario in which the patient was admitted with liver failure and experienced failure in exactly two organ systems at a given day during the ICU stay.

To investigate the relation between MOF and outcome, we define the following scenarios:

$$\text{died}: \quad \{\texttt{ICU death}(\mathbf{t}) = 1\}, \text{ and} \tag{10}$$

$$\text{MOF2exit}: \quad \{\texttt{MOF2}(\mathbf{t_1}), \texttt{exit}(\mathbf{t_2}) | \mathbf{t_2} = \mathbf{t_1} + 1\} \tag{11}$$

which indicate respectively the scenario in which the patient died at the ICU and the scenario in which the patient suffered from MOF at the day before leaving the ICU.

Finally, the abstract scenario MOFmaxexit refers to the scenario in which the maximum number of failing organ systems was reached at the day before ICU discharge or ICU death. For brevity we do not provide the definition of MOFmaxexit here, we will however use this scenario in our experiments.

4.3 Estimating probabilities

This section describes the transition probability function we used and the resulting probabilities associated with particular scenarios based on these abstractions.

Transition probability function The number of possible states clearly shows the need for a parametric transition probability function. We chose to describe the transition probabilities using regression equations. The regression equations express the probability that the value of the dependent variable at time point t equals 1, given the values of the covariates at time point $t-1$. As covariates we only use the six variables that relate to the clinical condition of the patient. The status indicator variables do not form part of the covariates, as this would result in an improper scenario once the the status indicator variable evaluates to 1. The variables `ICU death` and `ICU discharge` do however serve as dependent variables. The combination of the regression equations for these two variables makes a separate equation using `exit` as dependent variable redundant. So in total eight regression equations are developed.

We note that we assume the variables at time t to be independent of each other, given the values for the covariates at time $t-1$. Therefore we estimate the parameters in these equations independently for all dependent variables using normal regression procedures. As the variables in our data are all binary, we used logistic regression analysis. A more extensive description of the procedure we applied and the resulting coefficients are given in [Peelen *et al.*, 2006]. The resulting Markov model is depicted in Figure 1.

Probability of scenarios Based on this Markov model we generated 10,000 scenarios using Monte Carlo simulation. The values for $x_1(1), \ldots, x_m(1)$ were sampled based on the distribution in the original data. For both $\mathcal{D}^*$ (generated data) and $\mathcal{D}$ (OLVG data) the aforementioned abstractors were calculated. Together with the time-indexed variables these were used to estimate the probability of a number of scenarios. To enable a comparison, these estimates were based on $\mathcal{D}$ and on $\mathcal{D}^*$.

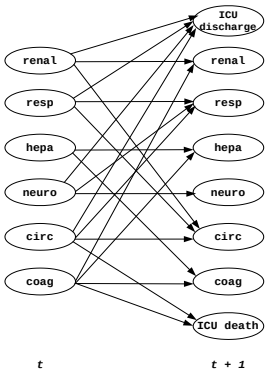


Figure 1: The resulting Markov model

Results Table 1 shows the results for four of these scenarios. Comparing estimates based on $\mathcal{D}$ and $\mathcal{D}^*$, we note that for simple scenarios both datasets yield similar results, which indicates that our model is able to correctly represent the basic underlying processes in organ failure. Once the scenarios get more complicated however, a difference between the estimates occurs. The estimates based on $\mathcal{D}$ are unreliable as they are based on few patients only (26 in the fourth scenario). For the more complex scenarios we therefore use the results based on $\mathcal{D}^*$.

5 Discussion

This paper describes an approach to estimate the probability of particular scenarios for a disease. The approach has been illustrated with a case-study in the area of intensive care. Temporal abstraction (TA) plays an important role in our work. TA is often applied in the situation in which a large time series describes the changes in a single variable over time. Based on such a time series, TA results in a description of subsequent *states* of a variable, and *trends*, i.e. changes in these states over time [Bellazzi *et al.*, 2000]. These abstractions can be purely based on statistics of the data (e.g. by using the distribution of the data), or medical knowledge can serve as input in the TA process (e.g., in defining state boundaries) [Verduijn *et al.*, 2005].

In contrast to most of the work on TA, our approach applies to situations in which the condition of the patient is described by multiple, discrete variables, sampled at a low frequency. In these situations *knowledge-based* temporal abstraction is more appropriate. In our approach medical knowledge is explicitly incorporated in the specification of the time and state abstractors, and in the variables that are included in the Markov model. Our time and state abstractors relate to the concepts of horizontal and vertical inference, respectively, as introduced by [Shahar and Musen, 1996].

Another aspect at which our approach differs from most of the work on TA lies within the *use* of the abstractions. In other work TAs often provide a 'summary' of the time series and serve as input for further modeling or reasoning. In our approach TAs are defined apart from the data, and we subsequently use the data to estimate the frequency at which these TAs (or combinations thereof) occur.

Scenario	$\mathcal{D}$ $(n = 1508)$		$\mathcal{D}^*$ $(n = 10000)$	
$P(\mathsf{died})$	0.164	(248/1508)	0.173	(1727/10000)
$P(\{\mathtt{hepa}(1) = 1\}\|\mathsf{died})$	0.214	(53/248)	0.213	(368/1727)
$P(\{\mathtt{hepa}(1) = 1, \mathtt{renal}(\mathbf{t}) = 1\}\|\mathsf{died})$	0.105	(26/248)	0.063	(108/1727)
$P(\{\mathsf{MOFmaxexit}\|\{\mathtt{hepa}(1) = 1, \mathtt{renal}(\mathbf{t}) = 1\}, \mathsf{died})$	0.462	(12/26)	0.676	(73/108)

Table 1: Probability of scenarios estimated based on original and simulated data

Markov models have been used in health care for various purposes since the 1970s [Beck and Pauker, 1983]. They have been implemented using Causal Probabilistic Networks, e.g. by [Riva and Bellazzi, 1996; Andreassen *et al.*, 1999]. [Charitos *et al.*, 2005; Kayaalp *et al.*, 2001] used dynamic Bayesian networks to model changes in the condition of patients admitted to the ICU. In the latter study, changes in SOFA scores were used, among other variables, to predict ICU survival. They reduced the parameter space by replacing a series of scores by a binary variable indicating whether a particular pattern was present. Conceptually this relates to our abstract scenarios. However, our abstract scenarios do not restrict to one particular variable.

We are aware that our approach has its limitations. An important determinant of the success of the approach is the Markov model. The first-order Markov assumption may be too strict for many clinical applications. The assumption of stationarity can be questioned for some disease areas and might disturb the model's fit to the data. It therefore seems useful to somewhat relax these assumptions. Finally, the parametric form chosen for $f_1, \ldots, f_m$ (in our application logistic regression) should be appropriate. Therefore evaluation of the transition probability function is an important aspect in the application of this approach.

We have chosen to estimate the probability of scenarios using Monte Carlo simulations, instead of using methods for explicit probabilistic inference. This saves us from the necessity to model all abstractors explicitly. Furthermore, adding abstractors would rapidly increase the complexity of a dynamic Bayesian network, whereas the complexity of performing Monte Carlo simulations increases in a linear fashion. We do however realize that in the case of rarely occurring scenarios, the number of simulations that is to be generated also requires a large number of computations.

In our experiments on ICU data we assumed a very simple Markov model. In future work we will improve the model by adding factors that are known to influence the transition probabilities, such as admission type (medical or surgical) and pre-existing chronic organ dysfunction (e.g., cirrhosis), and by adding "memory variables" that copy or summarize information from earlier states. Another important direction of future work lies within the development of appropriate evaluation measures. Finally, until now we have estimated the probabilities of scenarios that were defined in advance. In future investigations we will apply data mining techniques on $\mathcal{D}^*$ to discover frequently occurring scenarios.

Acknowledgments

Niels Peek receives a grant from NWO (Netherlands Organization for Scientific Research) under no. 634.000.020.

References

[Andreassen *et al.*, 1999] S Andreassen, C Riekehr, B Kristensen, et al. Using probabilistic and decision-theoretic methods in treatment and prognosis modeling. *Artif Intell Med*, 15:121–134, 1999.

[Beck and Pauker, 1983] JR Beck and SG Pauker. The Markov process in medical prognosis. *Medical Decision Making*, 3(4):419–458, 1983.

[Bellazzi *et al.*, 2000] R Bellazzi, C Larizza, P Magni, et al. Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. *Artif Intell Med*, 20:37–57, 2000.

[Bone *et al.* , 1992] R.C. Bone and other members of the ACCP/SCCM Consensus Conference Committee. Concensus conference: Definitions for sepsis and organ failure. *Crit Care Med*, 20(6):864–874, 1992.

[Charitos *et al.*, 2005] T Charitos, LC Van der Gaag, S Visscher, et al. A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in icu patients. In *Working notes of IDAMAP*, 2005.

[Kayaalp *et al.*, 2001] MS Kayaalp, GF Cooper, and G Clermont. Predicting with variables constructed from temporal sequences. In *Proc 8th Int WS on Artificial Intelligence and Statistics*, pages 220–225, 2001.

[Kjærulff, 1995] U Kjærulff. dHugin: a computational system for dynamic time-sliced Bayesian networks. *International Journal of Forecasting*, 11:89–111, 1995.

[Peelen *et al.*, 2006] L Peelen, N Peek, NF De Keizer, et al. Discovering scenarios for organ failure in the intensive care unit. *Proc. Medical Informatics Europe (MIE)*, 2006. *To appear*.

[Riva and Bellazzi, 1996] A Riva and R Bellazzi. Learning temporal probabilistic causal models from longitudinal data. *Artif Intell Med*, 8:217–234, 1996.

[Robert and Casella, 2004] CP Robert and G Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[Shahar and Musen, 1996] Y Shahar and MA Musen. Knowledge-based temporal abstraction in clinical domains. *Artif Intell Med*, 8:267–298, 1996.

[Verduijn *et al.*, 2005] M Verduijn, A Dagliati, L Sacchi, et al. Comparison of two temporal abstraction procedures: a case study in prediction from monitoring data. In *Proc AMIA annual symposium*, 2005.

[Vincent *et al.*, 1998] J-L Vincent, A De Mendonca, F Cantraine, et al. Use of the SOFA score to assess the incidence of organ dysfunction/failure in intensive care units. *Crit Care Med*, 26(11):1793–1800, 1998.

Panel discussion: *An infrastructure for collaboration in time series analysis*

TSNet – A Distributed Architecture for Time Series Analysis

Jim Hunter
Department of Computing Science
University of Aberdeen
King's College, Aberdeen, UK
jhunter@csd.abdn.ac.uk

Abstract

Abstraction of complex time-series data is a necessary precursor to other higher-level activities; this is particularly true in the Intensive Care Unit. If we are to obtain a consensus as to the best ways to achieve these abstractions, different research groups need to be able to experiment with data acquired from a variety of sources, and to apply algorithms developed elsewhere. This paper sets out an infrastructure which has been developed to allow collaborative working of this kind.

1 Introduction

Imaging apart, the Intensive Care Unit (ICU) is arguably the clinical environment that generates the largest volume of data. The greatest contribution to this comes from the measurement of up to ten physiological variables often as frequently as every second, 24 hours a day (i.e. almost a million measurements per day). In addition there will be a number of data items which are entered sporadically – laboratory results, blood gases, medication, equipment settings, patient observations, etc.

These data can be processed for a number of purposes including:

- the application of clinical guidelines and other decision support activities;
- generation of textual summaries;
- clinical audit;
- data mining.

All of these activities share the same requirement as far as the raw data are concerned – namely that the volume of the data be reduced in some way, i.e. that abstractions be generated. This process of abstraction may involve time series from only one numerical variable or from several; it may also involve the data which are entered sporadically. It will almost certainly involve the removal of low level artifact arising from patient movement, ambient noise and clinical intervention – e.g. [Cao et al., 1999]; higher level abstractions may involve segmentation e.g. [Keogh *et al.*, 2001], trend detection, Markov modeling e.g. [Williams *et al.*, 2006] and other sophisticated pattern matching techniques.

At present, most of the analysis of time series data from the ICU is carried out by individual research groups, who apply techniques developed in their own laboratories to data generated by their own clinical collaborators. There is little sharing of data nor comparison of the effectiveness of different techniques when applied to the same data.

In part, I believe this to be due to the lack of a suitable infrastructure to enable this sharing of data and algorithms to take place. This paper presents such an infrastructure which allows sharing across the internet. Our vision is of a researcher in group A, being able to access data acquired by group B, and comparing a signal processing algorithm she has developed (say in MatLab) with algorithms developed by groups C (written in Java) and D (written in Delphi). She might even use a display technique written by group E.

I believe such collaboration to be necessary because:

- the abstraction of complex ICU time series data is difficult; we do not know in advance which approach(es) will bear fruit;
- people tend not to appreciate the advantages or disadvantages of the algorithms developed by others until they have tried them themselves;
- it is dangerous to claim generality for a specific approach until it has been tested on data from different sources.

The basic architecture we propose (called TSNet) is, unsurprisingly, based on the standard client/server model. Before we describe it in more detail, we need to define some terms.

2 Definitions

2.1 Channels

A *channel* consists of a named data stream. The data may be the raw data, or may be the result of some form of processing. Channels have two main sub-classes – equi-sampled and interval. The data values in *equi-sampled* channels are (as the name implies) acquired at a regular, constant frequency. The data 'values' can be a variety of different types:

- numerical (floating point);
- Boolean;
- enumerated (in the Pascal sense – i.e. sequential integers starting at 0);

- a vector of floating point values – typically a frequency spectrum.

An *interval channel* consists of a set of temporal intervals, each defined by:

- start and end date/times;
- an attribute name;
- a value (of any type).

Intervals can be of zero length duration, known as *events*. Interval attributes can be organised into a tree structure, known as a *descriptor tree*. Irregularly sampled numerical channels can also be handled by an interval channel of events.

2.2 Filters

Within TSNet, any module which has zero or more channels as inputs and zero or more channels as outputs is known as a *filter*. A very simple example would be the family of moving window filters (mean, standard deviation, median, slope, etc.) which take in one equi-sampled numerical channel and output a channel of the same class. Special sub-classes of filter are:

- data sources, which have no inputs but whose output channels provide the raw data from a particular source; of course there are inputs in the forms of files or databases, but the function of the raw data source is to hide the details of the format of that data from the rest of the system;
- data sinks, which have one or more channels as inputs but no output channels; again, of course, there *are* outputs - one type of data sink is the inverse of the raw data source, in that the input channels are written to a permanent medium.

The introduction of sinks and sources means that every channel is the output of some filter and acts as the input to one or more filters.

Filters can carry out more complex operations such as segmentation, where the filter takes in an equi-sampled numerical channel and outputs an interval channel - the intervals representing the segments. Filters can even be complex rule-based pattern recognisers. One example is a filter that recognises the presence of a transcutaneous probe change [Hunter and McIntosh, 1999]. Another example is the Asbru Guideline Execution Engine which takes in a number of processed channels (e.g. with artefacts removed) and outputs an interval channel containing the recommendations which result from the application of a particular guideline written in the Asbru language [Fuchsberger *et al*., 2005].

One barrier to collaboration between groups is that they may well use different programming languages and are reluctant to devote time to translating their algorithms into languages that other groups can easily use. By defining standards for channels and for the interfaces to filters, TSNet enables filters to be written in a variety of languages including Java, Delphi, MatLab and CLIPS.

2.3 Plots

The most common type of data sink is the plot, where input channels are converted to a visual representation; although plots are sub-classes of filters, they are sufficiently specialised that we will refer to them explicitly. The most common type of plot is that of the values of a variable against time, but different type of plot are appropriate for representing interval channels, descriptor trees, spectra, etc.

2.4 Displays

A *display* consists of a number of plots, laid out according to the wishes of the user.

2.5 Channel/Filter Networks

An output channel of one filter can be an input to another thus enabling the construction of complex channel/filter networks. Filters specify the classes of channel that they input and output and any network must respect these type constraints. A display will specify the plots it requires, a plot specifies its channels, a channel specifies the filter it is derived from (which in turn will specify other channels and filters) – see Figure 1, with display D, plots P1 and P2, and filters F1 to F5; channels are shown by arrows.

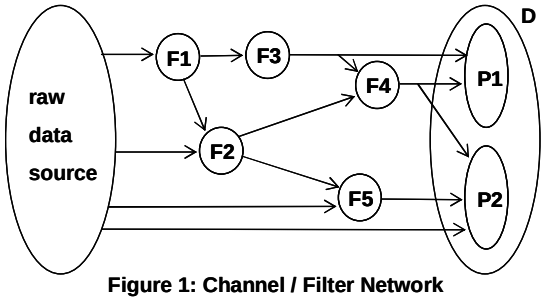


Figure 1: Channel / Filter Network

2.6 Complex filters

We can define a connected sub-graph of channel and filter classes as constituting a filter class in its own right; such a filter is said to be *complex*, in that it can be decomposed into further 'hidden' filters and channels. Figure 2 shows a complex filter has with two input channels, two output channels, three hidden filters and three hidden channels.

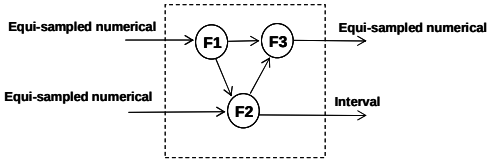


Figure 2: Complex Filter

2.7 Contexts

TSNet is designed to be flexible and incremental. To this end, *contexts* are defined and organised hierarchically. Typically a context corresponds to a specific data source and/or project. Contexts provide an environment for the definition of channels, filters (especially raw data sources) plots and displays. All contexts inherit from the ROOT context. As is usual with inheritance hierarchies, the advantage is that filters, channels etc. are inherited down the hierarchy. Filters which are used extensively (e.g. involving moving windows or segmentation) can be defined in the ROOT context, whereas more specialised filters can be defined at an appropriate level. An example of a deep hierarchy is found in the Neonate project [Hunter *et al*., 2003a and b, Law *et al*., 2005]:

ROOT
  ICU
    Neonatal ICU
      Badger (the basic data collection system)
        Neonate (including observational channels)
          GraphVsText (a specific experiment)

2.8 Catalogues

A catalogue is a list of time periods (called data periods) which the researcher wishes to study. Contexts can have as many catalogues defined for them as is desired. Normally the catalogue is displayed to the user for her to select a data period for display, but it is possible to arrange for the same processing to be applied to all of the elements of a catalogue.

2.9 TSNetClasses

The TSNet architecture makes explicit the classes which are available and which can be referred to – these are implementation independent. The base class is TSNetComponent:

TSNetComponet
  Catalogue
  Channel
    Equisampled
      Numerical
        EquisampledNumerical
      EquisampledBoolean
      EquisampledEnumerated
    Interval
  Context
  Display
  Filter
    Plot

2.9 Parameters

All TSNetComponents can have associated *parameters* with the following class structure:

String
    Memo
    DescriptorTree
    FileName
    URL
    DirectoryName
    Selection
Integer
    Colour
Float
Boolean

For example, moving window filters need to know the width of the window, and the amount by which the window is advanced.

3 TSNet Clients

TSNet clients enable the user (i) to manage the catalogues; (ii) to configure channels, filter classes, plots and displays; (iii) to display the data from a selected data period. Clients may implement filters internally or they may invoke external filters as TSNet services.

At present only one serious client, the Time Series Workbench (TSW) has been implemented (in Delphi); however a prototype is also being written in Java to demonstrate that clients can be written in other languages. The TSW offers the following functionality.

3.1 Catalogues

Within the TSW, the contents of the catalogue may be determined by:

- the structure of the raw data; if the data has been collected sporadically over an extended period, then the catalogue will represent those intervals over which data is available;
- some other temporal structure in the data; for example, in the Neonate project we were interested in those time periods where the baby was being manually ventilated; these intervals were available as the result of observations and could be made available as a catalogue;
- intervals which someone (usually a clinical expert) has 'marked up' by hand while looking at other data; a good example would be where the expert has marked up the presence of artefact.

3.2 Plots

A plot is a visual representation of one or more channels. The simplest plot is a graph of value against time (referred to as a *temporal* plot) but there are many other possibilities:

- the descriptor tree associated with a channel;

- interval plots can list the attributes and values of the intervals forming an interval channel;
- time slice plots show the values present in all channels at a particular point in time.

Figure 3 shows a temporal plot, a descriptor tree and an interval plot.

3.3 Configuration

The TSW allows interactive configuration of channels, plots and displays:

- *Channel* configuration includes the introduction of new raw data channels and channels derived from other filters, the deletion, copying and renaming of channels, the specification of the parameters of the channel and of the filter from which the channel is derived.
- In configuring a *plot*, the user specifies which channels are to be displayed. Interval channels may be displayed in a number of ways – as a solid bar, as lines indicating the start and end points of the interval, as shading underneath numerical data, etc); the TSW allows these characteristics to be set interactively. As with channels, plots can be created, deleted, copied and renamed.
- Within the TSW, a *display* consists of a number (currently three) of separate areas; see Figure 3. Each area can contain as many plots on selectable tabs as are required. Configuring a display means allocating specific plots to specific areas on the display.

3.4 Execution

Execution consists of displaying the data specified by an individual catalogue entry. In order to improve efficiency, only those channels which are required for plotting are computed; this subset of all possible channels is derived by working backwards from the plots through the channel/filter network.

A vital parameter for any filter class is *Execution at*; this defines whether the filter is implemented internally within the client or externally. It makes sense to implement some filters (especially simple ones which belong to the root context) within the client for the sake of efficiency.

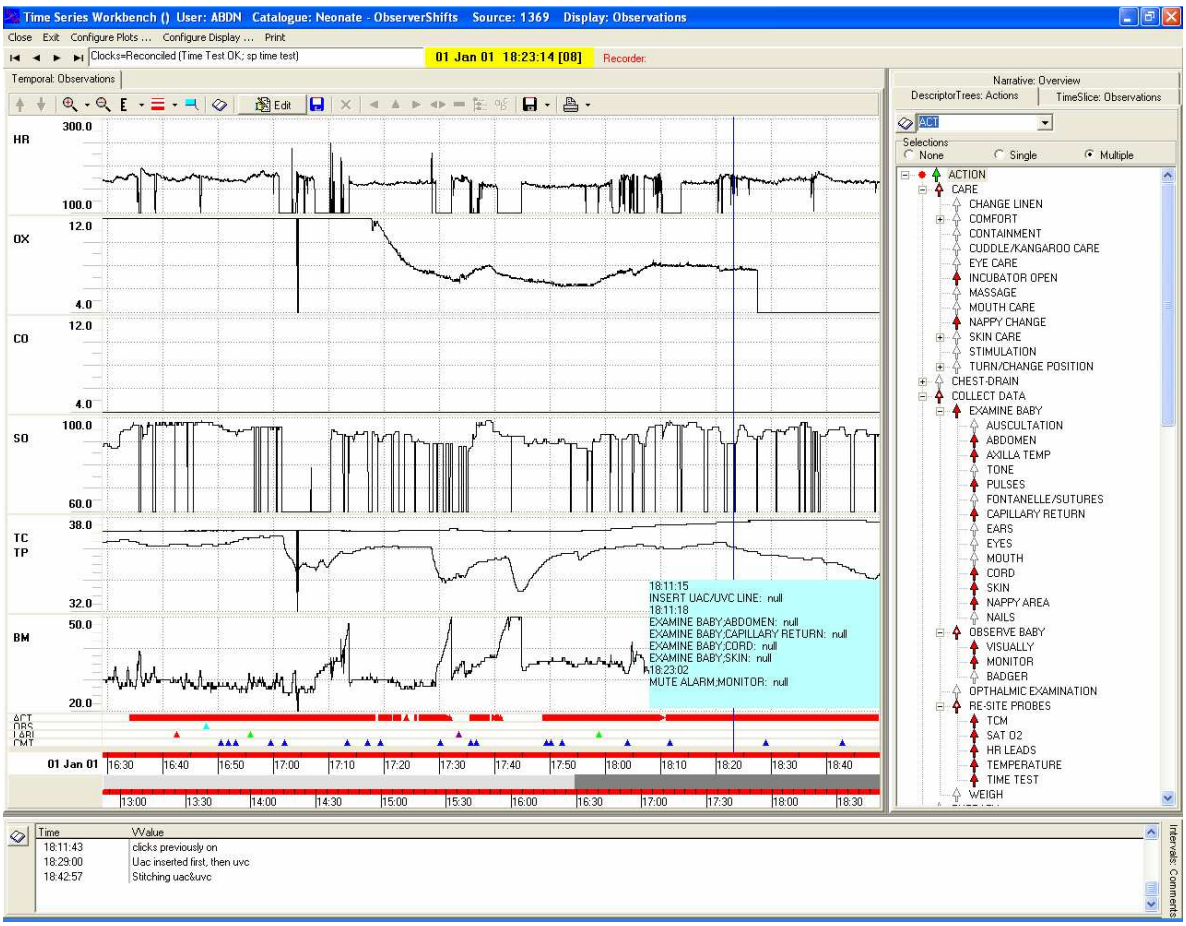


Figure 3 TSW Client

4 TSNet Servers

Servers make external filters available. Currently TSNet uses web service technology supported by Apache Tomcat. The primary interface on the server side is a filter manager which locates the filter concerned and arranges for the channel data to be presented to and recovered from it. Special case has been taken to optimize the structure of the SOAP messages to ensure rapid transfer of large volumes of data.

5 Collaboration

As we have said, the main aim of TSNet is to allow collaboration between different sites (research groups). This means that channels, filter classes (including complex filters), plots and displays can be named and described by one group or individual in such a way that they can be used by others. This requires that mechanisms be found for managing name spaces and for exchanging definitions.

5.1 Name spaces

Groups and individuals are organized in a hierarchy of *originators*. At the root of this hierarchy is a super-user called CORE. This user corresponds to the TSNet administrator, and is responsible for those entities which are judged to be useful to all users. Such entities will belong to root context. Under CORE will be a number of sites and within each site a number of individuals.

An entity is fully named by the following tuple:

- the context to which it belongs;
- its originator;
- its class;
- a name which is unique within the space defined by context/originator/class.

5.2 TSNetClass Definitions.

The built-in TSNet classes were described in section 2.9; all of these classes belong to the ROOT context, are owned by the CORE originator and can not be altered. However subclasses of filter can be defined by other originators; for each subclass, the originator needs to provide:

- the name of the class;
- an abbreviation;
- a list of the input channel classes;
- a list of the output channel classes;
- a list of the parameters which are applicable to this filter class (in addition to those defined for the generic filter) and default values;
- a textual description - this is the only place where the semantics of the operation of the filter are provided; it is up to users to decide whether the filter provides functionality which is of use to them.

Whenever a new TSNet class is defined, it is tagged with the name of the originator.

All of the TSNet classes available to a given client are described in a database associated with that client.

5.3 Instances of TSNetComponent

Instances of channels, filters, plots, displays and studies as configured by the user are held in the same database as the TSNet classes. Each is tagged with the name of the originator and the context to which it belongs.

5.4 Sharing

The general architecture of TSNet is shown in Figure 4. Sharing of classes and instance takes place as follows.

An individual user will have her own TSNet database. Initially this will contain those classes and instances which are built-in to her client plus those that she has developed herself. Let us suppose that she has developed a new filter class. Once the class has been debugged, she copies the definition of the class to her site (server side) database and installs the class as a run time module on the site's web server.

Each site participating in the collaboration will maintain a TSNet database containing those classes and instances which originate from that site. These site databases are visible (with password protection) to all other sites and users who are members of the collaboration.

Either by regular browsing of the TSNet sites, or by email notification, other users become aware of the existence of the filter and copy the definition into their own personal databases. Note that they are not importing the filter code, but only sufficient information for their client to refer to the filter and to configure it into channel/filter networks.

The TSNet client should ensure that imported entities are read-only so that definitions of named entities are

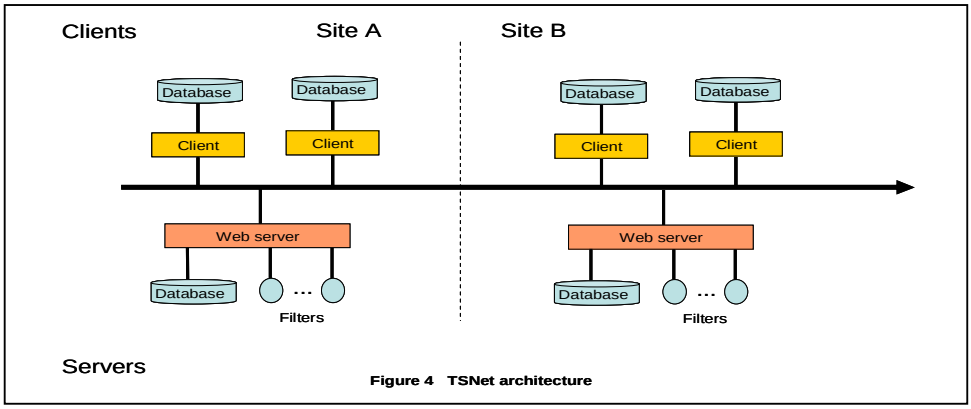


Figure 4 TSNet architecture

unique within the system. It is always open to users to modify copies of imported definitions, but the user who makes the copy will become the originator of the copy.

When this other user comes to execute the filter on a particular data period, his client recognizes that the filter is external, and has enough information to locate the relevant web-server. The client then interacts with that server, sending the data for the input channel(s) and recovering the output channel(s).

Such an arrangement has the advantage that the originator of the filter can make it available to the community without giving up ownership. We envisage that other levels of access could be made available, such as allowing the copying of the compiled code, or eventually of its source.

It is possible for a site to expose the description of a complex filter (i.e. one composed of other filters) while retaining the knowledge about its internal workings. Thus an external user of such a complex filter exports the input channels to the site which 'owns' the filter. That site then takes responsibility for managing the passage of the data through the elements of the complex filter, even when this may involved exporting the data to a third site which owns filters which contribute to the operation of the complex filter.

Confidentiality is important when it comes to exchanging medical data. TSNet assumes that all input data sets have been fully anonymised before being made available.

5 Discussion

TSNet has much in common with the MEDIATOR architecture [Wiederhold and Genesereth, 1997] and specialisations thereof designed to handle time oriented data [Nguyen et al., 1997; Boaz and Shahar, 2005]. What differentiates TSNet from these systems is its emphasis on the processing of large volumes of rapidly sampled data. The consequences of this emphasis may be implementational rather than conceptual, but they are none the less significant.

TSNet represents a serious attempt to enable collaborating research groups to work together in developing ways of analysing complex time series data. Issues of standardized schemas and vocabularies need to be addressed; however given that TSNet has been designed to foster collaborative research between a relatively small number of groups, this issue is perhaps less important than if it were attempting to offer a real-time service to a wider community.

The system has been fully implemented and a demonstration will be provided – over the internet if suitable access facilities are available. It is hoped that it can be evaluated by an interested user group in the near future.

Acknowledgements

The author is grateful to all of those students who have contributed to the development of TSNet over the years, but in particular to Christian Fuchsberger, Steven McHardy, Mark Homer, Lazaros Mavridis, Paul McCue and Ying Gao. The NEONATE project was supported by the UK PACCIT Programme run jointly by ESRC and EPSRC.

References

[Boaz and Shahar, 2005] D Boaz and Y Shahar, "A Framework for Distributed Mediation of Temporal-Abstraction Queries to Clinical Databases", *Artificial Intelligence in Medicine*, **34**(1), 2005.

[Cao *et al*., 1999] CG Cao, IS Kohane, N McIntosh and K Wang, "Artifact Detection in the PO_2 and PCO_2 Time Series Monitoring Data from Preterm Infants", *Journal of Clinical Monitoring and Computing*, **15**, pp 369-378, 1999.

[Fuchsberger *et al*., 2005] C Fuchsberger, JRW Hunter, P McCue, "Testing Asbru Guidelines and Protocols for Neonatal Intensive Care", *AIME-05: Proceedings of the Tenth European Conference on Artificial Intelligence in Medicine*, Springer Verlag, pp 101-110, 2005.

[Hunter and McIntosh, 1999] JRW Hunter and N McIntosh, "Knowledge-Based Event Detection in Complex Time Series Data", *AIMDM'99: Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, Horn W et al. (Eds.), Springer Verlag, pp 271-280, 1999.

[Hunter *et al*., 2003a] JRW Hunter, Y Freer, G Ewing, R Logie, P McCue and N McIntosh, "NEONATE: Decision Support in the Neonatal Intensive Care Unit – A Preliminary Report", *AIME-03: Proceedings of the Ninth European Conference on Artificial Intelligence in Medicine*, Springer Verlag, pp 41-45, 2003.

[Hunter *et al*., 2003b] JRW Hunter, L Ferguson, Y Freer, G Ewing, R Logie, P McCue and N McIntosh, "The NEONATE Database", *Workshop on Intelligent Data Analysis in Medicine and Pharmacology and Knowledge-Based Information Management in Anaesthesia and Intensive Care, AIME-03*, pp 21-24, 2003.

[Keogh *et al*., 2001] E Keogh, S Chu S, D Hart D, M Pazzani, "An Online Algorithm for Segmenting Time Series", *Proceedings of IEEE International Conference on Data Mining*, pp 289-296, 2001.

[Law *et al*., 2005] A Law, Y Freer, JRW Hunter, RH Logie, N McIntosh, J Quinn, "A Comparison of Graphical and Textual Presentations of Time Series Data to Support Medical Decision Making in the Neonatal Intensive Care Unit", *Journal of Clinical Monitoring and Computing*, 19, pp 183-19, 2005.

[Nguyen et al., 1999] J Nguyen et al., "Integration of Temporal Reasoning and Temporal-Data Maintenance into a Reusable Database Mediator to Answer Abstract Time-Oriented Queries: The Tzolkin System", *J Int. Inf. Sys.*, 13, pp 121-145, 1999.

[Wiederhold and Genesereth, 1997] G Wiederhold and M Genesereth, "The Conceptual Basis of Mediation Services", *IEEE Expert*, **12**(5), pp 38-47, 1997.

[Williams *et al*., 2006] CKI Williams, J Quinn, N McIntosh, "Factorial Switching Kalman Filters for Condition Monitoring in Neonatal Intensive Care", to appear in *Advances in Neural Information Processing Systems* 18, 2006.

Paper session: *Information Retrieval, Data Mining*

Investigating the Value of Diagnosis Codes in the Primary Care Patient Record

Thomas Brox Røst, Øystein Nytrø, Anders Grimsmo
Department of Computer and Information Science and The Norwegian EHR Research Center
Norwegian University of Science and Technology
Trondheim, Norway
{brox, nytroe}@idi.ntnu.no, anders.grimsmo@medisin.ntnu.no

Abstract

We have used supervised document classification to investigate whether or not the ICPC diagnosis code used in the primary care patient record complements the associated encounter note. Our hypothesis is that diagnosis codes are set independently from the notes and that the code accordingly provides additional information that is not reflected in the note. To investigate this hypothesis we built a set of document classifiers using data from a primary care practice and attempted to predict the correct diagnosis code for each encounter note. At best we achieved a correct prediction rate of 51.6 %. Given our results we discuss possible reasons behind misclassifications and find some indicators that the diagnosis code may add information that can not be inferred from the encounter note.

1 Introduction

In this study we attempt to classify primary care clinical encounter notes into their corresponding diagnosis groups. We do so by learning document classifiers from a manually coded dataset collected from a Norwegian primary care center. While being able to infer diagnoses from written text may be a worthy goal in itself, e.g. for detecting incorrect diagnoses and improving electronic patient record systems, our main purpose is to explore if the diagnosis code complements the written encounter note; that is, whether or not it adds information that is not explicitly stated in the written note. Research has shown that manual encoding of primary care encounter notes tend to be of high quality [Nilsson *et al.*, 2003]. Assuming that the diagnosis is a relevant descriptor of the encounter note, we seek to use classification as an estimate of the amount of overlap between the written note and the diagnosis. A lack of predictive power might indicate that the encounter note can not be viewed as a stand-alone entity and that surrounding information (diagnosis, prescriptions, etc) must also be taken into account.

The electronic patient record (EPR) has gradually attained widespread usage in primary care. In Norway, more than 90 % of primary care physicians are routinely using computer-based patient-record systems [Bayegan, 2002] and many have been doing so for more than 15 years. A typical feature of most commercial EPR systems in use today is that the encounter note, which is the main documentation of the doctor-patient consultation, is written as free-text narrative. There are perfectly practical reasons for this: Unstructured free-text is easy to write and represents the traditional way of documenting patient treatment. However, this makes the information within less suitable for automated processing and thereby keeps the EPR from fulfilling its potential as a useful tool for both research and clinical practice. Attempts have been made to create EPRs that impose varying degrees of structure on the clinical narrative, but with varying success so far.

To alleviate this problem, many researchers have turned to the use of natural language processing (NLP), text classification and text mining techniques on clinical narrative. Some NLP systems have proven very useful in a number of clearly defined domains, such as detection of bacterial pneumonia from chest X-ray reports [Fiszman *et al.*, 2000], finding adverse drug events in outpatient medical records [Honigman *et al.*, 2001] and discharge summaries [Melton and Hripcsak, 2005], and identifying suspicious findings in mammogram reports [Jain and Friedman, 1997]. A common feature of these systems is that they restrict themselves to a narrow clinical domain with a clearly defined vocabulary and a limited form of discourse, such as one would find in specialized hospital reports. Our long-term goal is to draw on research from these areas and explore the usefulness of similar techniques on the primary care patient record. However, the lack of empirical knowledge on the content in primary care documentation raises the need for preliminary investigations on its narrative structure. The main motivation behind this initial study is therefore to learn more about the informational value and underlying documentational patterns in primary care encounter notes.

2 Background

Among the characteristic features of primary care encounter notes are sparseness, brevity, heavy use of abbreviations and many spelling mistakes. The notes are normally written during the consultation by the treating physician, this in contrast with hospital patient records which are usually dictated by the physician and then transcribed by a secretary. A typical encounter note might look something like this:

Table 1: ICPC chapter codes.

Chapter code	Description
A	General and unspecified
B	Blood, blood-forming organs and immune mechanism
D	Digestive
F	Eye
H	Ear
K	Circulatory
L	Musculoskeletal
N	Neurological
P	Psychological
R	Respiratory
S	Skin
T	Endocrine, metabolic and nutritional
U	Urological
W	Pregnancy, child-bearing, family planning
X	Female genital
Y	Male genital
Z	Social problems

> Inflamed wounds over the entire body. Was treated w/ apocillin and fucidin cream 1 mth. ago. Still using fucidin. Taking sample for bact. Beginning tmnt. with bactroban. Call in 1 week for test results[1].

The encounter note will often follow the Subjective-Objective-Assessment-Plan (SOAP) structure, although not necessarily in a strict manner [Nilsson *et al.*, 2003].

To classify such notes we rely on the presence of manually coded diagnosis codes. The use of clinical codes in primary care is common in the United Kingdom, the Netherlands, and Norway [Letrilliart *et al.*, 2000]. The motivation for coding is both for reimbursement and statistical purposes. In our experimental dataset the notes are coded according to the ICPC-2 coding system. ICPC-2 is the second edition of the International Classification of Primary Care, a coding system which purpose is to provide a classification that reflects the particular needs and aspects of primary care. Using a single ICPC code, each health care encounter can be classified so that both the reasons for encounter, diagnoses or problems, and process of care are evident. Together, these elements make out the core parts of the health care encounter in primary care. Moreover, one or more encounters associated with the same health problem or disease form an episode of care [Hofmans-Okkes and Lamberts, 1996].

ICPC-2 follows a bi-axial structure with 17 chapters along one axis and 7 components along the other. The chapters are single-letter representations of body systems (Table 1) while the components are two-digit numeric values (Table 2). As an example, "R02" is the ICPC code for shortness of breath.

There are several examples of attempts to automate the coding of diagnoses [Franz *et al.*, 2000; Larkey and Croft, 1996; March *et al.*, 2004; Satomura and do Amaral, 1992; Vale *et al.*, 2003], all of which concern themselves with the alternative ICD code. ICD is a more complex code than ICPC and is more suited for specialized usage in hospitals. [March *et al.*, 2004] describes the use of Bayesian learning to achieve automated ICD coding of discharge diagnoses. [Franz *et al.*, 2000] compares coding methods with and without the use of an underlying lexicon and concludes that lexicon-based methods perform no better than lexicon-free methods, unless one adds conceptual knowledge. [Larkey and Croft, 1996] found that using a combination of different classifiers yielded improved automatic assignment of ICD codes. There is a practical purpose to automated ICD coding: ICD is a more complex code than ICPC and accordingly manual ICD encoding takes up a lot of time. There have also been other approaches towards automated coding of clinical text. [Hersh *et al.*, 1998] attempted to predict trauma registry procedure codes from emergency room dictations. [Aronow *et al.*, 1995] classified encounter notes in order to find acute exacerbations of asthma and radiology reports for certain findings, this through the use of Bayesian inference networks and the ID3 decision tree algorithm. Document classification and IR has been applied in other medical domains as well, such as clustering of medical paper abstracts [Makagonov *et al.*, 2004].

Examples of automated ICPC coding are less common. [Letrilliart *et al.*, 2000] describes a string matching system that assigns ICPC codes from free-text sentences containing hospital referral reasons, based on a manually created look-up table. We have not found examples of similar attempts at automated ICPC classification in the literature.

As for classification techniques, this study uses support vector machines (SVM). SVMs have proved useful and have shown good general performance for text classification tasks [Joachims, 1998] when compared with other methods. Our goal for this study is not to compare classification methods; this will be explored further in future work.

[1] Translated from the Norwegian.

Table 2: ICPC component codes.

Number	Range	Description
1	01-29	Complaint and symptom component
2	30-49	Diagnostic, screening, and preventive component
3	50-59	Medication, treatment, procedures component
4	60-61	Test results component
5	62-63	Administrative component
6	64-69	Referrals and other reasons for encounter
7	70-99	Diagnosis/disease component

3 Methods and Data

We have collected a dataset from a medium-sized general practice office in Norway. The data consists of encounter notes for a total of 10,859 patients in the period

Table 3: Number of ICPC codes per encounter.

Number of ICPC codes	Number of encounters
1	235,860
2	44,651
3	6,037
$\geq$4	1,320

from 1992 to 2004. All in all, there are 482,902 unique encounters. The Norwegian Health Personnel Act [Health Personell Act, 2001] requires that caregivers provide "relevant and necessary information about the patient and about the health care" in the patient record. In practice, this manifests itself as a combination of structured and unstructured information about the encounter. Information such as personal details about the patient, prescriptions, laboratory results, medical certificates and diagnosis codes is typically available in structured format, while encounter notes, referrals and discharge notes are in the form of unstructured free-text. For the purposes of this paper, we have only considered the encounter notes and the accompanying ICPC-2 diagnosis code.

A known source of noise is that a minority of the notes are likely to be written in Danish or *nynorsk* (literally "New Norwegian") rather than standard Norwegian (*bokmål*). There are also more than 20 different authors, so there may be differences in documentational style as well. Interns fresh out of medical school may for example be inclined to document more thoroughly than an experienced physician.

The dataset has been automatically anonymized using a custom-built anonymization tool [Tveit *et al.*, 2004]. Each word or token is controlled against a database of words that are known to be insensitive and a set of rules that deal with alphanumeric patterns such as medication doses, date ranges, and laboratory test values. Sensitive tokens are replaced with a general identifier or an identifier that shows the type of token that was replaced.

Each encounter will typically consist of a written note of highly variable length and zero or more accompanying ICPC codes. 287,868 of the available encounters have one or more ICPC codes (Table 3).

There are some notable differences in terms of code use between hospital and primary care settings. [Larkey and Croft, 1996] describes a test set of discharge summaries with a mean of 4.43 ICD-9 codes per document, while [Nilsson *et al.*, 2003] notes that a set of Swedish general practice patient records has a mean of 1.1 ICD-10 codes per record. While there may be regional and cultural differences with respect to coding practice, the latter corresponds with our findings of 1.2 ICPC-2 codes per note (Table 3).

Since we concern ourselves with the relation between the encounter note and the ICPC code, we discard all encounters with more than one code in order to avoid ambiguity in the training data. Of the 235,860 encounters that are left, 175,167 have an accompanying encounter note.

The use of ICPC codes as classification bins for encounter notes is essentially a multi-class classification problem. Since there are 726 distinct ICPC codes it becomes practical to reduce the class dimensionality. We choose to group codes according to their chapter value, so that we are left with the 17 single-letter body codes as classes.

Table 4: Average note length and class frequencies.

Chapter	Avg. words	St. dev.	Samples	Class freq.
N (Neurological)	40	33.2	5,637	3.2 %
D (Digestive)	39	30.0	11,386	6.5 %
Z (Social)	36	35.1	570	0.3 %
X (Female genital)	36	27.1	6,244	3.5 %
P (Psychological)	32	35.6	9,939	5.6 %
A (General)	32	28.9	12,052	6.8 %
Y (Male genital)	31	24.9	1,993	1.1 %
F (Eye)	31	23.5	4,998	2.8 %
L (Musculoskeletal)	29	26.8	36,493	20.8 %
R (Respiratory)	28	21.8	22,846	13.0 %
K (Circulatory)	27	25.6	21,089	12.0 %
H (Ear)	27	21.3	5,526	3.1 %
W (Pregnancy)	26	24.5	5,614	3.2 %
U (Urological)	26	25.2	4,502	2.5 %
T (Endocrine)	26	22.4	5,498	3.1 %
S (Skin)	26	20.3	18,432	10.5 %
B (Blood)	22	23.3	2,348	1.3 %

When grouping encounter notes by their ICPC chapter value we note that there is a varying degree of verbosity. The use of sparse encounter notes is often common in primary care, for instance when renewing recurring prescriptions. To determine average note verbosity for each ICPC chapter, all relevant encounter notes are tokenized. After removing stop words, whitespace and other noisy elements, the average length and standard deviation is calculated as shown in Table 4. The table also shows that the class frequency distribution is highly skewed, with the top three classes (L, R and K) covering 45.8 % of the selected encounter notes.

We note that Larkey's discharge summaries [Larkey and Croft, 1996] has a mean length of 633 words, which is more than an order of magnitude higher than for the notes in our dataset. Notwithstanding cultural and institutional differences, this highlights how hospital discharge summaries usually provide a more self-contained description of the patient. In the Norwegian health care system the patient will typically use just one primary care physician who acts as a gatekeeper for admittance to specialized hospital care. This implies that the primary care physician is highly involved in most phases of a patient's contact with health services. The effect of this persistent doctor-patient relationship on primary care documentation is that the information found in a single encounter note will often just add to information found in previous encounter notes, thereby offering just a small glimpse of the complete situation.

Since many classification techniques, including support vector machines, are restricted to dealing with binary classification tasks, we have to reduce our multi-class classification task into a set of binary tasks. For each pair

Table 5: L versus D classifier, 20 most relevant features.

Original n-gram	Appr. Eng. translation	Comment
ve kne	left knee	Abbr
bevegelighet	movability	
celeston	celeston	
hø kne	right knee	Abbr
kne	knee	
ankel	ankle	
kneet	the knee	
fot	foot	
skulder	shoulder	
hø skulder	right shoulder	Abbr
kiropraktor	chiropractor	
hofte	hip	
lat	lateral(?)	Abbr
nakke	neck	
ryggsmerter	backache	
traume	trauma	
stiv	stiff	
lår	thigh	
falt	fell	
hevelse	swelling	

of classes $(i, j) : i, j \in \{A, B, \ldots, Z\}$ where $i, j = 1 \ldots c, j \neq i$ we create a two-class classifier $< i, j >$. If c is the number of classes, we end up with $c(c-1)$ binary classifiers, or $17 \times 16 = 272$ in this case. This technique is known as double round robin classification [Fürnkranz, 2002]. The classifier $< i, j >$ will then solely consist of training examples from encounter notes with ICPC chapter codes i and j. To determine the final predicted class of any given note we feed it through each classifier and record the result. The class that receives the highest number of predictions is chosen to be the most likely one. In case of ties we choose the class with the highest number of occurrences in the training set, or, as a last resort, pick one at random. To build and run the classifiers we used the SVM-Light[2] toolkit with the default parameter settings.

We use word and phrase frequencies as the base component when constructing feature vectors for the classifiers. If we were to rely on single words alone we would lose some contextual information [Hersh *et al.*, 1998], so frequency counts are performed on all unigrams, bigrams and trigrams in the encounter note, excluding stop words. The occurrence of an n-gram is recorded as a *true* value in the feature vector. While n-grams may be a simplistic way of representing context, it still allows us to catch phrases and turns of words that may have discerning qualities.

As is common with word-based feature vectors, it is useful to apply some dimension-reducing technique to limit the size of the vector. The challenge lies in pruning those features that are the most inconsequential to the classifier's predictive qualities. For this experiment we adapt a technique described in [Kruger *et al.*, 2000]. For each classifier the frequency of all unigrams, bigrams and trigrams occurring in all training notes for both classes are counted. If an n-gram occurs in more than 7.5 % of either the true or the false class notes it is tagged as a likely candidate for inclusion. All candidates are then ranked according to their true class frequency to false class frequency ratio. Finally the top 100 candidates are chosen as the most relevant features.

[2]http://svmlight.joachims.org/

Table 6: Classification results.

Training examples	Test examples	Correct	Accuracy
320	320	116	36.3 %
3,145	3,145	1,458	46.4 %
36,846	10,000	5,162	51.6 %
173,167	2,000	994	49.7 %

Four different experiments were run, using training/test ratios of 320/320, 3,145/3,145, 36,846/10,000 and 173,167/2,000. The class frequency distribution for each training and test data set was kept approximately consistent with the overall dataset frequency distribution.

4 Results

Table 6 shows the prediction accuracy for all four experiments, while Table 7 shows the results for the experiment with 36,846 training cases. As a comparison, guessing for the most frequent chapter code (L) all the time will yield an accuracy of 19.9 %.

5 Discussion and Future Work

Increasing the amount of training data will to a certain extent improve the classifier. Our best results were with 36,846 training examples, which yielded an accuracy of 51.6 %. For this classifier we note that the accuracy is highly variable for the individual chapters; from no correct predictions at all (B, U, Y and Z) to 91.2 % in the best case (L). The most notable feature is how the L (musculoskeletal) class appears to soak up the majority of the misclassified cases. The L class is the largest group in the training set, indicating that a certain bias towards this class should be expected but not dramatically so. When attempting to perform the same classification task without the L cases the S group became the major misclassification bin, but in a less prominent fashion; the overall accuracy rate rose to 57.5 %.

A possible explanation is that the notes with L-chaptered ICPC codes deal with a broader range of bodily experiences; that is, they tend to cover more ground than the more specialised chapters. Table 5 shows the 20 most relevant features for one of the L-classifiers. We notice that the features associated with the L class describe several different parts of the body (knee, ankle, foot, shoulder, hip, thigh, neck) and different kinds of pain experiences (swelling, trauma, backache). If some of these features are typical for other diagnosis types as well it would seem natural that the L class absorbs encounter notes that do not have unambiguous, unique features.

Moreover, the L-related features may be terms that are typically used to describe the patient's subjective experiences. [Nilsson *et al.*, 2003] notes that the subjective and

Table 7: Results of training with 36,846 notes and testing with 10,000.

Correct ICPC chapter	Predicted ICPC chapter																	Sum	Percent correct
	A	B	D	F	H	K	L	N	P	R	S	T	U	W	X	Y	Z		
A	**52**	0	9	0	0	125	273	1	20	118	57	0	0	11	8	0	0	674	7.7 %
B	1	**0**	4	0	0	9	73	0	3	6	5	0	0	34	0	0	0	135	0.0 %
D	7	0	**247**	0	0	32	256	1	15	33	31	0	0	4	5	0	0	631	39.1 %
F	1	0	1	**50**	0	11	127	1	5	9	61	0	0	0	0	0	0	266	18.7 %
H	1	0	0	0	**22**	13	141	3	1	57	32	0	0	1	0	0	0	271	8.1 %
K	5	0	6	0	0	**924**	243	3	8	14	28	0	0	1	2	0	0	1,234	74.8 %
L	3	0	4	0	0	58	**1,821**	2	21	35	52	0	0	0	0	0	0	1,996	91.2 %
N	4	0	2	0	0	72	189	**25**	8	14	21	0	0	0	4	0	0	339	7.3 %
P	3	0	2	0	0	29	325	2	**143**	9	5	0	0	0	1	0	0	519	27.5 %
R	8	0	4	0	0	30	291	0	16	**849**	28	0	0	0	0	0	0	1,226	69.2 %
S	6	0	5	0	0	20	346	0	4	22	**648**	0	0	1	4	0	0	1,056	61.3 %
T	3	0	1	1	0	70	133	1	24	1	7	**5**	0	7	3	0	0	256	1.9 %
U	3	0	5	0	0	17	221	0	5	23	15	0	**0**	1	7	0	0	297	0.0 %
W	2	0	3	0	0	53	224	0	7	7	13	0	0	**184**	27	0	0	520	35.3 %
X	3	0	9	0	0	28	152	0	10	11	16	0	0	19	**192**	0	0	440	43.6 %
Y	2	0	2	0	0	6	80	0	1	4	10	0	0	0	1	**0**	0	106	0.0 %
Z	0	0	1	0	0	0	28	0	5	0	0	0	0	0	0	0	**0**	34	0.0 %

objective descriptions makes up the bulk of the encounter note. Again, when unable to find features that are characteristic for the other classes, the more general, subjective features will make the classifier default to L. This might indicate that the diagnosis code does in fact add extra information; in many cases the encounter note alone will not be sufficient to give a complete picture of the encounter. This also corresponds with Nilsson's findings, where the assessment (or diagnosis) part of the encounter note was shown to be relatively small in comparison with the other parts.

There are several other possible approaches to approving the predictive quality of the classifier. We made no attempts to normalize the vocabulary in the training data. Techniques such as stemming or mapping terms to a common controlled vocabulary would reduce the number of relevant features. This would also involve dealing with common misspellings [Hersh *et al.*, 1997] and dialect terms, both of which are quite common in our dataset. [Wilcox and Hripcsak, 2003] notes that the use of expert knowledge can provide a significant boost to medical text report classifiers.

We made no efforts to control the amount of noise in the classifiers or to screen the notes in the test data set. Very short notes and notes with non-standard language use were not discarded. In the case of short notes, the diagnosis code will nonetheless be the main information carrier for the consultation. Also, we did not consider the number and distribution of different writers on our dataset. As have been noted, there might be differences in documentational style; training and testing on notes written by the same physician could have been attempted for the sake of comparison.

The a priori anonymization could also influence the results. Since the anonymization tool only allows known non-sensitive words, it is likely that special and unusual words are lost. Such words may have a higher predictive effect than more common words. Comparing the classifier on a non-anonymized dataset could possibly indicate how much of destructive effect that is incurred due to anonymization.

The choice of ICPC chapter codes as class indicators is not necessarily a natural choice. Alternatives include grouping according to ICPC component codes or, as a natural follow-up, attempting to classify into the full ICPC codeset of 726 different codes. One could also consider classifying into several classes rather than discarding notes with more than one ICPC code as in this experiment.

The use of round-robin all-vs-all classification can be argued; a simpler one-vs-all scheme might work just as well [Rifkin and Klautau, 2004] with the added bonus of being less computationally expensive. Moreover, we made no attempts to evaluate different feature selection mechanisms and classification methods; this is scheduled for future work.

Finally, we must bear in mind that the results are from single test runs rather than using e.g. cross-validation techniques.

In general, our naive, largely domain-ignorant approach granted results that are interesting enough to legitimate further work in this area. Given the findings it would be worth investigating if the use of accompanying information from the EPR, such as lab results and prescriptions, can help improve classification quality. Another possible approach is to view the encounter note in its longitudinal context by also considering notes from previous (and following) encounters related to the same episode of care.

Acknowledgments

Thanks go to Amund Tveit, Ole Edsberg, Inger Dybdahl Sørby and Gisle Bjørndal Tveit for comments and suggestions.

References

[Aronow *et al.*, 1995] D. B. Aronow, S. Soderland, J. M. Ponte, F. Feng, W. B. Croft, and W. G. Lehnert. Automated classification of encounter notes in a computer based medical record. *Medinfo*, 8 Pt 1:8–12, 1995.

[Bayegan, 2002] Elisabeth Bayegan. *Knowledge Representation for Relevance Ranking of Patient-Record Contents in Primary-Care Situations*. PhD thesis, Norwegian University of Science and Technology (NTNU), 2002.

[Fiszman *et al.*, 2000] M. Fiszman, W. W. Chapman, D. Aronsky, R. S. Evans, and P. J. Haug. Automatic detection of acute bacterial pneumonia from chest x-ray reports. *J Am Med Inform Assoc*, 7(6):593–604, 2000. Evaluation Studies Journal Article.

[Franz *et al.*, 2000] Pius Franz, Albrecht Zaiss, Stefan Schulz, Udo Hahn, and Rüdiger Klar. Automated coding of diagnoses - three methods compared. In *Proceedings of the Annual Symposium of the American Society for Medical Informatics (AMIA)*, Los Angeles, CA, USA, 2000.

[Fürnkranz, 2002] Johannes Fürnkranz. Round robin classification. *J. Mach. Learn. Res.*, 2:721–47, 2002.

[Health Personell Act, 2001] Health Personell Act. Act of 18 may 2001 no. 24 on personal health data filing systems and the processing of personal health data, 2004.04.12 2001.

[Hersh *et al.*, 1997] W. R. Hersh, E. M. Campbell, and S. E. Malveau. Assessing the feasibility of large-scale natural language processing in a corpus of ordinary medical records: a lexical analysis. *Proc AMIA Annu Fall Symp*, pages 580–4, 1997.

[Hersh *et al.*, 1998] W. R. Hersh, T. K. Leen, P. S. Rehfuss, and S. Malveau. Automatic prediction of trauma registry procedure codes from emergency room dictations. *Medinfo*, 9 Pt 1:665–9, 1998.

[Hofmans-Okkes and Lamberts, 1996] I. M. Hofmans-Okkes and H. Lamberts. The international classification of primary care (icpc): new applications in research and computer-based patient records in family practice. *Fam Pract*, 13(3):294–302, 1996.

[Honigman *et al.*, 2001] B. Honigman, P. Light, R. M. Pulling, and D. W. Bates. A computerized method for identifying incidents associated with adverse drug events in outpatients. *Int J Med Inform*, 61(1):21–32, 2001. Journal Article.

[Jain and Friedman, 1997] N. L. Jain and C. Friedman. Identification of findings suspicious for breast cancer based on natural language processing of mammogram reports. *Proc AMIA Annu Fall Symp*, pages 829–33, 1997.

[Joachims, 1998] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, London, UK, 1998. Springer-Verlag.

[Kruger *et al.*, 2000] Andries Kruger, C. Lee Giles, Frans Coetzee, Eric Glover, Gary Flake, Steve Lawrence, and Cristian Omlin. Deadliner: Building a new niche search engine. In *Ninth International Conference on Information and Knowledge Management, CIKM 2000*, Washington, DC, 2000.

[Larkey and Croft, 1996] Leah S. Larkey and W. Bruce Croft. Combining classifiers in text categorization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 289–97, Zurich, Switzerland, 1996. ACM Press.

[Letrilliart *et al.*, 2000] L. Letrilliart, C. Viboud, P. Y. Boelle, and A. Flahault. Automatic coding of reasons for hospital referral from general medicine free-text reports. *Proc AMIA Symp*, pages 487–91, 2000.

[Makagonov *et al.*, 2004] Pavel Makagonov, Mikhail Alexandrov, and Alexander Gelbukh. Clustering abstracts instead of full texts. *Lecture Notes in Computer Science*, 3206:129–35, 2004.

[March *et al.*, 2004] Alan D. March, Eitel J. M. Laura, and Jorge Lantos. Automated icd9-cm coding employing bayesian machine learning: a preliminary exploration. In *Simposio de Informtica y Salud 2004*, 2004.

[Melton and Hripcsak, 2005] G. B. Melton and G. Hripcsak. Automated detection of adverse events using natural language processing of discharge summaries. *J Am Med Inform Assoc*, 12(4):448–57, 2005.

[Nilsson *et al.*, 2003] G. Nilsson, H. Ahlfeldt, and L. E. Strender. Textual content, health problems and diagnostic codes in electronic patient records in general practice. *Scand J Prim Health Care*, 21(1):33–6, 2003. Journal Article.

[Rifkin and Klautau, 2004] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–41, 2004.

[Satomura and do Amaral, 1992] Y. Satomura and M. B. do Amaral. Automated diagnostic indexing by natural language processing. *Med Inform (Lond)*, 17(3):149–63, 1992.

[Tveit *et al.*, 2004] Amund Tveit, Ole Edsberg, Thomas Brox Røst, Arild Faxvaag, Øystein Nytrø, Torbjørn Nordgård, Martin Thorsen Ranang, and Anders Grimsmo. Anonymization of general practioner's patient records. In *Proceedings of the HelsIT'04 Conference*, Trondheim, Norway, 2004.

[Vale *et al.*, 2003] Rodrigo F. Vale, Berthier A. Ribeiro-Neto, Luciano R.S. de Lima, Alberto H.F. Laender, and Hermes R.F. Junior. Improving text retrieval in medical collections through automatic categorization. *Lecture Notes in Computer Science*, 2857:197–210, 2003.

[Wilcox and Hripcsak, 2003] A. B. Wilcox and G. Hripcsak. The role of domain knowledge in automating medical text report classification. *J Am Med Inform Assoc*, 10(4):330–8, 2003.

Event Chart Explorer: A Prototype for Visualizing and Querying Collections of Patient Histories

Ole Edsberg[1),3)*], **Stein Jakob Nordbø**[1),3)], **Øystein Nytrø**[1),3)] **and Anders Grimsmo**[2),3)]

[1)] Department of Computer and Information Science

[2)] Department of Community Medicine and General Practice

[3)] Centre for EHR Research

Norwegian University of Science and Technology, Trondheim, Norway

Abstract

We present our work in progress on a system for visualizing and querying collections of patient histories. The main features of the system are: 1) Compact LifeLines-like visualization of histories as explorable and configurable time lines above a common time axis, with any query hits outlined, and 2) Operations for search, selection, sorting and alignment of the histories based on temporal queries.

1 Introduction

The ability to query and visually explore collections of patient histories is potentially useful in several types of tasks: When faced with a difficult clinical decision, one could search for similar fragments from other histories in the database and explore them to learn from what happened in other cases. In quality assurance of a clinical practice, one could search for deviations from guidelines and explore the result to see if the deviations were justified. In preparing research on clinical processes, one could search for relevant history fragments and explore them to improve one's understanding of the subject matter and get ideas for research hypotheses and analysis methods.

The well-known LifeLines system [Plaisant *et al.*, 1998] provides a time line visualization of the elements of a history. Event charts [Lee *et al.*, 2000] provide a static visualization of a *collection* of histories as a set of stacked and possibly aligned lines above a common time axis, with events represented by glyphs on the lines. The visualization part of our approach can be seen as an attempt to combine the information rich, interactive LifeLines visualization with the event charts' ability to visualize many histories. Related to the query part of our system, the literature describes a query system based on the event calculus that allows users to query collections of series of measurements for patterns of temporal abstractions [Combi and Chittaro, 1999]. Our query language is mainly intended for searching for patterns in the categorical event and interval data of the patient record, and it consequently contains a different set of constructs. We also have a different approach to result visualization.

*Contact email: `edsberg@idi.ntnu.no`

2 The visualization and query system

Our data model contains point events for contacts, diagnoses and lab results, and interval events for prescriptions. (Our data source unfortunately often requires some guesswork in determining end points of prescriptions.) Figure 1 shows and explains the main view of our system, leaving the query language and query-based operations for the rest of this section. Applying a query to a history results in a set of matches, where a match is defined by its starting and ending points in time. The query language consists of primitive constructs matching data elements in themselves and composite constructs specifying temporal constraints on their sub-queries. Recursively, a query may match:

- a point event, such as a diagnosis, lab test or prescription, or
- an interval of medication with a specified drug type, or the beginning or end of such an interval, or
- a sequence of the matches of two sub-queries, with possible constraints on the time that can pass between them, or
- the parallel or alternative occurrences of the matches of two sub-queries, or
- a window of a specified length, within which a sub-query does, or does not, match, or
- a sub-query's first match in the entire history.

Here is an informal description of an example query: *Find all history fragments where the patient first has a one-year time window without at least three positive blood pressure measurements, and then is prescribed blood pressure-related medication for the first time in the history*. (We omit the syntax, since it is currently under revision.)

The prototype provides the following query-based operations: With the `search` operation, the user submits a query, and red boxes are drawn around the matches of the query. The user can cycle through the matches. With the `select` operation, the user submits a query, and a new tab is opened, containing a visualization of only those histories that contained a match of the query. With the `align` operation, the user submits a query, and the histories are synchronized so that the start points of the matches line up vertically. The time axis changes to shows the number of time units relative to the alignment point. With the `sort` operation, the user submits a query with a sequence as the

Figure 1: Screenshot of the main view in the prototype. Each of the horizontal bands corresponds to a history. The lower left panel contains a legend of the information types selected to be displayed, in this case specialized for hypertension. In the bands, tall, narrow rectangles indicate diagnoses, coloured subdivision of the background medication, and arrows blood pressure measurements, with colour showing value category and orientation showing trend. The bottom panel shows the journal note and the upper left panel shows details about the events at the current position of the cursor. The `search` operation has been used to mark the hits of the query informally described as *Find all history fragments where the patient get his first hypertension diagnosis and then, sometime later, gets his first diagnosis for a hypertension-related complication*. The `select` operation has been used to extract the 359 histories containing a hit of this query. Then, the `align` operation has been used to synchronize the histories on the first part of the query. Finally, the `sort` operation has been used to sort the histories according to the distance between first and second part of the query. Menus not shown provide other possibilities, such as zooming, jumping to a journal-like view or changing the information types shown.

top-level construct, and the histories are sorted according to the distance between the matches of the sub-queries. By using these four operations, the visualization can be incrementally narrowed down and adapted to suit the problem at hand. Figure 1 shows a screen shot from our application of the prototype to a data set of about 10000 patient records, in collaboration with a general practitioner wanting to investigate the treatment of hypertension at his health centre.

3 Current work

We are currently working on 1) refining the query language according to our improved understanding of the users' needs, 2) grounding its semantics in the event calculus, and 3) creating a query editor that allows users to design queries in a flowchart-like visual language.

References

[Combi and Chittaro, 1999] Carlo Combi and Luca Chittaro. Abstraction on clinical data sequences: object-oriented data model and a query language based on the event calculus. *Artificial Intelligence in Medicine*, 17:271–301, 1999.

[Lee *et al.*, 2000] J. Jack Lee, Kenneth R. Hess, and Joel A. Dubin. Extensions and applications of event charts. *The American Statistician*, 54(1):63–70, 2000.

[Plaisant *et al.*, 1998] Catherine Plaisant, Richard Mushlin, Aaron Snyder, Jia Li, Dan Heller, and Ben Shneiderman. Lifelines: Using visualization to enhance navigation and analysis of patient records. In *Proc AMIA Annual Fall Symp.*, pages 76–80, 1998.

On the Arguments Against the Application of Data Mining to Medical Data Analysis

Anna Shillabeer, John F. Roddick and Denise de Vries
School of Informatics and Engineering
Flinders University,
PO Box 2100, Adelaide, South Australia 5001,
Email: {anna.shillabeer, roddick, denise.devries}@infoeng.flinders.edu.au

Abstract

There is a variety of criticisms of medical data mining which has led, in some cases, to the technology being overlooked as a tool. This paper presents a discussion of six of the strongest arguments against the application of data mining to the complex field of human medicine. The aim of the paper is to raise the predominant issues and suggest solutions whilst also opening the issues for further consideration by both medical and information technology communities.

The Arguments

1. Data mining outcomes are seen as generalisations and not verified for medical validity or accuracy. [Elwood and Burton, 2004; Milloy, 1995]

Medicine is a highly complex domain for which data mining processes were not designed. In many cases they originated in response to changes in commerce or management practices and there was no real need to substantiate results on the basis of protocols or domain knowledge. Medicine has requirements which are outside of the original scope of the technology, and to be applicable to a science which is concerned with critical decision making there is a need to modify the technology to reflect this different environment. Whilst this first argument is a serious issue, it is often borne from misrepresentation of the results of data mining rather than from the process itself. There is a heightened need for careful consideration of the language used when reporting results [Raju, 2003; Maindonald, 1998]. It is possible for the results to be specific but for the language of reporting to generalise the message. For example, the MJA described a case where a mining outcome showed that smoking does not have a direct link with skin cancer [Elwood and Burton, 2004], however the resulting media story reported that smoking is not linked with cancer generally. While a scientific data mining process was applied the language of information presentation was misleading and the resultant reporting was inaccurate and medically invalid. Medicine is especially sensitive to this form of information distortion and the consequences have the potential to be life threatening, politically sensitive, costly and persistent which is rarely the case in other domains.

There is little to sustain this argument in light of recent work in the field. By the application of suitable statistical methods, evaluation of all results and applying industry accepted standards there is no reason to believe that data mining cannot provide effective validation and accuracy checking processes [Shillabeer and Roddick, 2006; Gebski and Keech, 2003]. Three steps have been suggested to safeguard against this particular criticism [Smith and Ebrahim, 2002].

1. Results should not be published on the basis of correlation alone.
2. An explanation should be provided with the results to provide clarification e.g. A definition of the unique quality of the allergen that triggers the alleged immune response.
3. Results should be replicated, confirmed and documented prior to publication.

These steps are not part of standard data mining methodologies but are required to be undertaken if the mining of medical data is to overcome criticism, be viewed as 'good science' and gain trust in the medical community.

2. Associations are not representative of other similar attributes and do not consider other potential contributors. [Milloy, 1995; Raju, 2003; Smith and Ebrahim, 2002]

In a medical context, relationships found between one allergen and symptoms must be substantiated through analysis of similar allergens or the same allergen in other temporal, spacial or demographic instances. If this cannot be shown it suggests that there is not a conclusive argument for cause and effect or that some other catalyst or cause has been missed [Raju, 2003; Smith and Ebrahim, 2002]. Again, data mining was not designed to do this however this should not be a preventive. Methods are available to achieve this where it is important to determine the semantic closeness of results [Shillabeer and Roddick, 2006]. Criticism often focuses on data dredgers who promote results as facts rather than being indicative of a possible scenario requiring further investigation [Raju, 2003]. Where an association is found it is important to compare this with other associations or to apply a clustering algorithm to group semantically and determine where there is similarity or otherwise to other attributes or rules.

3. P-values are set arbitrarily and therefore the results cannot be trusted. [Milloy, 1995; Smith and Ebrahim, 2002]

The P-value is applied to the statistical testing of a null hypothesis to gauge the probability of the result happening by chance in a total population. Data mining provides a similar function through the use of support and confidence values although these apply only to the data set being mined, where support is the percentage of the data transactions *under analysis* that hold true for the association, and confidence (a.k.a. conditional probability) is the percentage of data transactions containing a specific attribute value that also contain another specific attribute value. Support and confidence values are thresholds set for *reporting purposes*

and are not p-values, although they are liable to attract the same criticism. P-values, support and confidence may be applied in two ways: to evaluate and discriminate the acceptability of data analysis results for follow-up research and, as a guideline or tool for reducing the number of irrelevant outcomes. Data mining can also be applied in divergent modes; to show what the common patterns in data are, or to show where common patterns are refuted in the data. It is important to always set heuristic thresholds in context of the specific analysis being done and in fact a calculation applied should not be used alone [Shillabeer and Roddick, 2006; Gebski and Keech, 2003]. In the medical domain attribute value relationships which occur frequently, and hence have high support and confidence, as well as a low p-value, are likely to be known already and would generally be of little if any interest. This is a major difference between traditional data mining applications, where generally the events which occur most frequently are of the greatest interest and hence have a similar support threshold, and applications in the medical domain where frequency is not a conclusive determinant in defining the usefulness, validity or applicability of results and hence may require varying threshold values.

4. Associations between attributes are dependent upon the data set being analysed and are not representative. [Raju, 2003; Smith and Ebrahim, 2002]

There is often a poor approach to the collection and description of data sources and samples which is not consistent with the process of data mining or other scientific methodologies [Milloy, 1995; Maindonald, 1998]. For results to be accepted the data source should be from an identifiable population with defined characteristics e.g. location, demographics, and proportions [Smith and Ebrahim, 2002]. In a clinical research setting this is overcome by the use of protocols and guidelines to ensure that results are representative and able to be replicated. One such protocol is CONSORT which is used globally by medical researchers and is endorsed by a number of prominent journals.

Data mining provides validation through the application of tools such as artificial intelligence and neural nets to the knowledge mining step to sample the data, provide outcomes then automatically test them on the whole data source to show that the outcome holds true for all available data not just one small subset [Smith and Ebrahim, 2002]. Data mining is a highly intensive machine process which utilises huge processing power, memory and time. Data sampling is often used as an initial step to reduce these constraints but correct utilisation may help to overcome this criticism also.

5. Data mining is simply a desperate search for something interesting without knowing what to look for. [Milloy, 1995; Smith and Ebrahim, 2002]

Exploratory mining, which is not constrained by user expectations, can uncover unexpected or unknown knowledge with wide reaching benefit and can be utilised to review and extend current medical knowledge. With the wealth of data being produced daily in the medical field the argument that it should not be used in an exploratory fashion to at least note important changes in data patterns demonstrates a misunderstanding of the potential value held therein. It is argued by some [Maindonald, 1998; Smith and Ebrahim, 2002; Shillabeer and Roddick, 2006] that it can be beneficial to look simply for something interesting rather than make an assumption about what is present in the data as if we only ever look for what is known we will potentially never find anything new and progress cannot be made. Provided this is a result of a scientific process then further mining or clinical trials can be undertaken for evidence to substantiate the initial findings. This criticism is only valid where the search is for anything interesting even if only minimally and where there is little or no validation.

6. Data mining displaces research and testing and presents results as facts requiring no further justification. [Milloy, 1995]

Contrary to the criticism, data mining in medicine is generally viewed as an efficient tool for enhancing the work done in the field rather than as a replacement for it [Maindonald, 1998]. Its value is seen as a process of *automated serendipity* that stimulates and supports testing rather than replaces it. When considering the use of mining outcomes there are two questions often asked; is this result representative of what has been recorded over time?, and can the analysis outcome be verified through real world application? [Raju, 2003; Smith and Ebrahim, 2002]. Whilst the first can be answered with some conviction by data mining the second requires clinical input and hence the process of providing trusted knowledge from data requires a collaborative effort by automated and clinical processes. When we consider that time from hypothesis to application of new knowledge is often measured in decades we should feel compelled to find new knowledge as quickly as possible and data mining offers the ideal tool for this.

Conclusion

This paper has presented six common criticisms of medical data mining in an effort to demonstrate that as technologists we need to be aware of the social environment in which we work and to give a suggestion of the importance of continuing to work on making the technology applicable to this complex domain. We should not be disheartened by the criticism which surrounds the field in which we work but should take the criticisms on board, work with them and provide an outcome which is beyond reasonable reproach.

References

[Elwood and Burton, 2004] J.M. Elwood and R.C. Burton. Passive smoking and breast cancer: is the evidence for cause now convincing? *Medical Journal of Australia*, 181(5):236–237, 2004.

[Gebski and Keech, 2003] V.J. Gebski and A.C. Keech. Statistical methods in clinical trials. *Medical Journal of Australia*, 178(4):182–184, 2003.

[Maindonald, 1998] J. Maindonald. New approaches to using scientific data- statistics, data mining and related technologies in research and research training. Occasional paper, Australian National University, 1998.

[Milloy, 1995] Steven Milloy. *Science without sense - The risky business of public health*. Cato Institute, Washington DC, 1995.

[Raju, 2003] S. Raju. Data flaws. Technical report, American Council on Science and Health, 2003.

[Shillabeer and Roddick, 2006] Anna Shillabeer and John F. Roddick. Towards role-based hypothesis evaluation for health data mining. *Electronic Journal of Health Informatics*, 1(1):e6, 2006.

[Smith and Ebrahim, 2002] G.D. Smith and S. Ebrahim. Data dredging, bias or confounding. *British Medical Journal*, 325(21-28 December 2002):1437–1438, 2002.

Software Demos

APPLICATION OF A COMMERCIAL SOFTWARE TO THE ANALYSIS OF INFRARED SPECTRAL IMAGES ON LYMPH NODE TISSUES: A PRELIMINARY STUDY

Emilio Burattini[1], Marco Chilosi[2], Carla Conti[3], Paolo Ferraris[3], Flaminia Malvezzi Campeggi[1], Francesca Monti[1], Giorgio Tosi[3], Alberto Zamò[2]

Department of [1]Computer Science and Department of [2]Pathology–Section of Pathological Anatomy
Università di Verona - Verona, Italy
[3]Department of Material Science - Università Politecnica delle Marche - Ancona, Italy

Abstract

In this work we present preliminary results obtained using a commercial software to analyze infrared spectra and infrared spectral images collected on lymph node tissues by using either a single detector or a multielement Focal Plane Array detector (FPA). Our results indicate that Cluster Analysis and Principal Component Analysis can allow to distinguish different compartments in the sample, as well as to classify tumoral and non tumoral samples. The need to treat huge amounts of spectral data, especially when the FPA is used for data acquisistion, would require to improve the software performance.

1 Introduction

Non-Hodgkin lymphomas constitute about 4% of all human malignancies. Morphological analysis coupled to immunohistochemistry is the most widely used approach for their classification, but these analyses are somehow prone to subjective interpretation. An automated technique such as Fourier Transform Infrared (FTIR) microscopy could bring the benefits of objective tissue evaluation, together with the possibility of using this technique as a first-line screening methodology to distinguish normal versus neoplastic samples [Conti et al., 2003].

As it is known, position and intensity of infrared absorption bands, particularly in the midle infrared region (i.e. in the wavenumber range from 4000 to 400 cm^{-1}), carry information on the biochemical status of organic samples, since they are related to the presence and amount of specific molecular groups and to their chemical environment [Gunzler-Gremlich, 2002]. The use of the interferometric technique (FTIR) allows to acquire in a reasonable time (say from a few minutes to half an hour) well resolved spectra with high signal to noise ratio from sample areas as small as about 10x10 μm^2. If the interferometer is coupled to a visible/IR microscope it is possible to obtain a spectral map by scanning a selected sample region. Recently, spectral images data can be obtained more rapidly using a multielement detector (Focal Plane Array, FPA) [Jackson et al., 2002], which allow to acquire simultaneously up to 4096 spectra from a 180x180 μm^2 sample area (down to 3x3 μm^2 spatial resolution). In this case, a huge amount of spectral data has to be analysed: for each point, the instrument acquires an interferogram (thousands to more than one hundred thousand data points, depending on the desired spectral resolution) which has to be Fourier Transformed to obtain the absorption spectrum (hundreds to thousands of data points, depending on the resolution and on the extension of the spectral region of interest). File dimensions go from hundreds of Kb for a single point spectrum up to tenths of Mb for a FPA acquisition.

2 Experimental and data analysis

This preliminary study included six samples from six different patients: three neoplastic samples of follicular lymphoma, as classified by morphology and immunoistochemistry [Jaffe et al., 2001], and three non-neoplastic samples of reactive lymph nodes. For each sample, our attention was focussed on three tissue compartments: intrafollicular zone, mantle and interfollicular zone. About 7 µm thick sections were measured in transmission mode on BaF_2 supports in the 4000-700 cm^{-1} spectral region with 4 cm^{-1} resolution using a Bruker Vertex-70 spectrometer coupled to a Hyperion 3000 microscope and equipped with a single element HgCdTe detector and with a HgCdTe FPA multidetector of 4096 elements. The Bruker 5.5. OPUS software was utilized for data analysis.

2.1 Point by point spectra

For each compartment of each one of the six samples, ten to twenty point by point absorption spectra (typically from a sample area of 50 x 50 μm^2 or 25x 25 μm^2) were acquired with the single element detector, as the ratio of the transmitted to the incident photon intensity vs wavelength. Cluster Analysis (using Euclidian distance to calculate spectral distances and the Ward's algorithm to determine the degree of heterogeneity) and Principal Component Analysis (PCA) applied on the spectra, by selecting various spectral regions and choosing different preprocessing methods (such as straight line subtraction, vector normalization, first or second derivative), allowed to:

- identify homogeneous groups of spectra (Figure 1) and exclude outliers before making an average to obtain a representative spectrum of each compartment of a sample.

The spectra representative of various sample regions can be further analysed for a more detailed interpretation of the behaviour of the bands.
- identify wider classes of spectra from different samples, such as “tumoral” and “non tumoral” (Figure. 2). In particular, our data indicate that non tumoral samples show a higher biological variability, as it is expected from pathology studies.

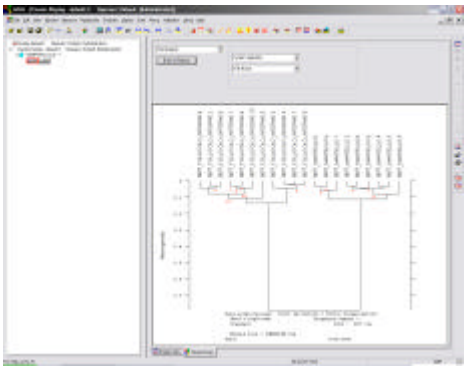


Figure 1: Spectra from mantle and intra-follicular zone of a sample are well distinguished by Cluster Analysis through 1st derivative and vector normalization in the range 1800-900 cm^{-1}.

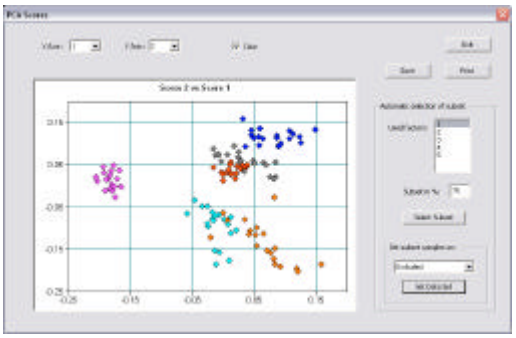


Figure 2: Two classes can be identified by PCA on intra-follicular spectra after straight line subtraction in the range 980-1350 cm^{-1}: in this score diagram blue, grey and red refer to point spectra from three different tumoral samples; magenta, cyan and orange to three different non tumoral samples.

2.2 Hyperspectral images

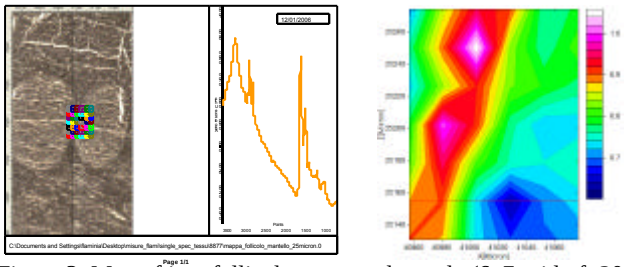


Figure 3: Map of intrafollicular zone and mantle (6x7 grid of 20 μm^2 points): in this example integration was performed at the 1080 cm^{-1} band. A typical spectrum is also shown in the center.

Hyperspectral images can be acquired either by mapping a limited selected region using the single detector or by mapping a wider region using the FPA multidetector.
It is then possible to draw a map of the peak height or of the intensity, after integration, of a single band (Figure 3 and Figure 4). As it can be seen, different compartments of the same sample can be clearly distinguished by FTIR.

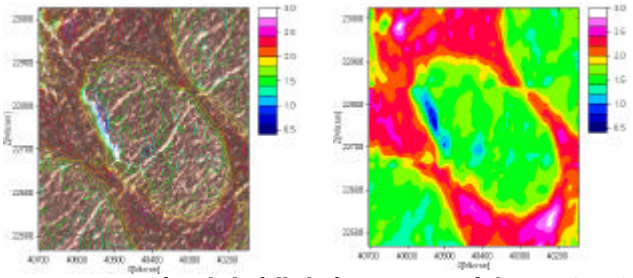


Figure 4: Map of a whole follicle from an area of about 550x550 μm^2 obtained with a 3x3 FPA grid, at 12x12 μm^2 spatial resolution. In this example integration was performed at the 1240 cm^{-1} band.

3 Conclusions

Present results are very promising as regards the use of FTIR microscopy for objective tissue evaluation and first-line screening to distinguish non-neoplastic versus neoplastic tissues. To this aim, an improved performance of the software would also be very useful: for example, to perform clustering directly on the acquired hyperspectral image of the same sample, in order to draw a color map of the clusters; and to run PCA analysis directly on the hyperspectral data from different samples to characterize tumoral and non tumoral classes.

Acknowledgments

This work was entirely supported by “Fondazione Cariverona” in the framework of “Funds 2003 for Research Projects of Biomedical Interest”.

References

[Conti et al., 2003] C. Conti, E. Giorgini, T. Pieramici, C. Rubini, G. Tosi, J. Mol. Struct., 744-7, 187 (2005).]; 1st DASIM Workshop (Diagnostic Applications of Synchrotron Infrared Microspectroscopy), Daresbury Lab.s, Manchester, July 2005; RSC Faraday Division. Faraday Discussion 126, Nottingham (2003).

[Gunzler-Gremlich, 2002] H. Gunzler, H.U. Gremlich, IR Spectroscopy, An Introduction, Wiley-Vch, 2002

[Jackson et al., 2002] M. Jackson, H.H. Mantsch, Pathology by Infrared and Raman Spectroscopy, in Handbook of Vibrational Spectrosc., J.M.Chalmes, P.R. Griffiths, Wiley, Chichester, 2002, Vol. 5, 3227-3245.

[Jaffe et al., 2001] E. Jaffe, N. Harris, H. Stein (editors): WHO classification of tumours: pathology and genetics of tumours of haematopoietic and lymphoid tissues. Lyon, France, IARC Press, 2001