Analysis of ICU Patients Using the Time Series Knowledge Mining Method

[1]Robert Moskovitch, [1]Dima Stopel, [2]Marion Verduijn, [2]Niels Peek, [3]Evert de Jonge, and [1]Yuval Shahar

[1]Medical Informatics Research Center, Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel

{robertmo,stopel,yshahar}@bgu.ac.il

[2]Dept of Medical Informatics, [3]Dept of Intensive Care Medicine, Academic Medical Center, University of Amsterdam, P.O.B. 22700, 1100 DE Amsterdam, The Netherlands

{m.verduijn,n.b.peek,e.dejonge}@amc.uva.nl

Abstract

Time oriented data presents a more detailed description of problems, while presenting challenges in the computational needs for a successful analysis, in which the time is explicitly analyzed. Commonly temporal datasets are converted into a static representation and being analyzed by common static data mining methods, such as decision trees. Abstracting time series into time intervals, using temporal abstraction, enables to analyze the data explicitly along time. We apply here a mining method, which discovers partially ordered coinciding time intervals, consisting on the Time Series Knowledge Mining (TSKM) method, presented by Mörchen [2006] on temporal data of intensive care patients, using human defined and two types of data driven temporal descretization methods as a preprocessing step. The Persist discretization method results with the best knowledge discovery outcome.

1 Introduction

The reduction in storage cost and the growth in logged temporal data present the opportunity to analyze data along time. Often the analysis of time stamped data is made using a time window, extracting features which describe the time series within the time window and enter them into a "static" data mining algorithms, such as decision trees or naïve bayes. However, determining the right time window size is commonly problematic and extracting features from the time series within a given time window, such as minimal value, or transformations such as wavelets or Fourier transform, do not allow an explicit temporal analysis. Alternatively, abstracting time series to meaningful time intervals enable to mine temporal data in a different approach, which not necessarily enforces to use windowing, which results in explicitly mining the data along the time axis [Moskovitch & Shahar, 2005].

: Representing time series through time intervals yields compact summary representation of the time series. For example, having values within the same range for a period of time can be represented in a time interval in which there is a period of stable values. Thus, a more compact representation is presented, preserving the time axis explicitly and enabling further mining of the time intervals. Time intervals can represent events having duration, or an abstracted time series, which we refer to in this study.

Within the recent half decade a growing interest in mining time intervals was observed. Most of the methods currently use Allen's temporal relations [Allen, 1983] for the representation of temporal knowledge. While Allen's relations were used and applied widely in *temporal reasoning* it has some disadvantages in the task of mining time intervals, as presented by Mörchen [2006b], on which we will elaborate later. These include mainly *ambiguity* and lack of *robustness*, to which he presents an alternative based on partially ordered coinciding time intervals [Mörchen 2006a,b]. In this study we applied Mörchen's method to a set of monitoring data from the intensive care unit (ICU) to examine its capabilities and advantages over Allen's temporal relations. This dataset was previously analyzed in comparative studies on temporal abstraction procedures for predicting the risk of prolonged mechanical ventilation after cardiac surgery [Verduijn et al, 2005; Sacchi et al, 2006].

We start by surveying the background. In the methods section we describe the descretization methods we used, the ICU dataset, the evaluation measures and the experimental plan. Finally, we report the results and discuss Mörchen's method as an alternative to Allen's.

2 Background

2.1 Time Intervals

Time intervals are defined often as a triple $ti=<ti._{start}, ti._{end}, ti._{symbol}>$, having a start-time, end-time and a symbolic value. Time intervals can represent an event in real life which has duration or an abstraction of time series data. Real life events have duration, represented by time interval, e.g., the duration in which the temperature of a patient increases. However, since the measurements are instantaneous resulting in time series, temporal abstraction is used to represent them as time intervals [Shahar, 1997].

2.2 Temporal Knowledge Representation

Temporal knowledge representation is an essential tool in the task of temporal data mining, which influences the entire mining process. Often Allen's thirteen temporal relations *before, meets, overlaps, starts, during, finishes*, and their corresponding *inverse* [Allen, 1983] are used for the representation of the time interval temporal patterns, however, in the context of knowledge discovery Allen's relations have some disadvantages, which we discuss in the following sections.

2.2.1 Mörchen's TSKM

In a very recent paper Mörchen [2006a] criticizes Allen's temporal relations as a tool for temporal knowledge discovery, specifying three main aspects:

(1) *Robustness*, claiming they are not robust since part of them requires the equality of two or more interval end-points. For example the relations *overlap*, *during* and *finishes* can describe a very similar situation, as shown in examples 1.a,b,c in figure 1, which illustrate the relations among two time intervals A and B.

(2) *Ambiguous*, since the same relation of Allen can visually and intuitively represent very different situations. Examples 2.a,b,c in figure 1 (modified from [Mörchen, 2006b]) illustrates the different situations which are all defined as *overlap*, while the different *overlap* may have different meanings.

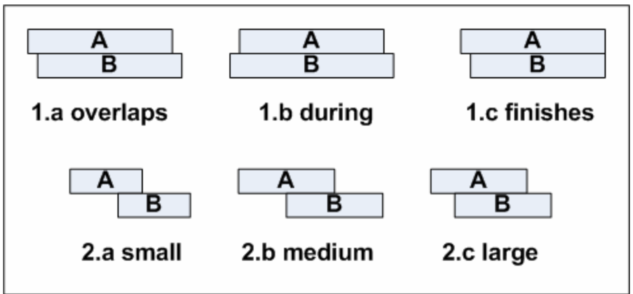

Figure 1 Examples 1.a,b,c illustrating the lack of *robustness* in Allen's relations, in which *overlap*, *during* and *finishes* represent a very similar situation. 2.a,b,c presents examples justifying the *ambiguity* in Allen's temporal relations.

(3) *not easily comprehensible*, representing a temporal pattern of time intervals using Allen's temporal relations requires the definition of all the pair-wise relations among each pair of time intervals [Höppner, 2001] which grows exponentially with the size of the pattern.

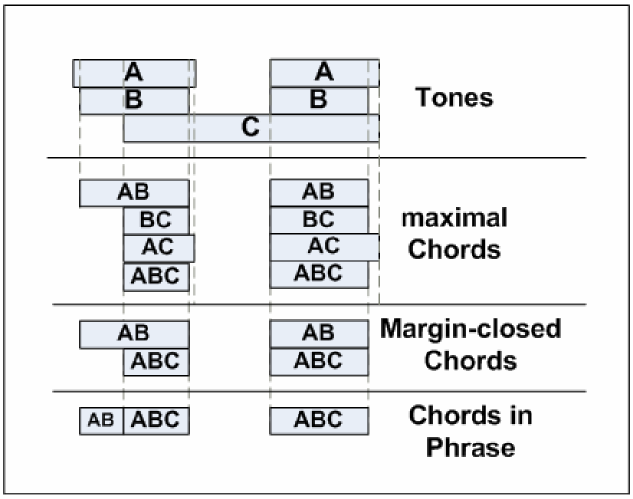

Figure 2 First we have Tones, which are labeled time interval appearing ini' the raw data. Maximal coinciding chords become Chords. A partial order of Chords constructs a Phrase.

Instead of Allen's thirteen temporal relations, Mörchen introduced a new hierarchical language for the representation of temporal knowledge based on time intervals, called the *Time Series Knowledge Representation* (TSKR). The TSKR is a hierarchical interval language describing the temporal concepts of *duration*, *coincidence*, and *partial order* in interval time series. The TSKR consists on three components: *Tones*, *Chords* and *Phrases*. A Tone is the basic primitive component, which is a labeled interval representing duration. Simultaneously occurring Tones form a Chord, representing *coincidence*. According to [Mörchen, 2006a,b] a Chord is defined by the sum of maximal appearances, thus no explicit restriction on the length of a Chord is defined. Several Chords connected with a partial order form a Phrase. A Phrase is defined by the partial order of the Chords, but the durations of the gaps are not restricted. Figure 2 (modified from Mörchen [2006b]) illustrates the process of the knowledge representation in the TSKR method. At the top there are the Tones, labeled time intervals (raw data), which based on their maximal coincidence Chords are constructed. Eventually a Phrase is constructed based on a partial order of Chords. The discovered Phrases are represented by a directed graph of Chords, in which the edges are Chords and their partial order is presented by the curves connecting them, as shown in figure 4.

2.3 Mining Time Intervals

2.3.1 Allen's based Time Intervals Mining

The problem of mining time intervals, a relatively young field, is attracting a growing attention recently. Generally, the task is, given a database of symbolic time intervals, to extract repeating temporal patterns. One of the earliest works was made by Villafane et al [1999], which searches for *containments* of intervals in a multivariate symbolic interval series. A containment model is constructed from the intervals, and rules are mined. Kam and Fu [2000] were the first to use all Allen's relations to compose interval rules. In their search for patterns it is restricted to right concatenation of intervals to existing extended patterns, called *A1* patterns. Additionally, they restrict the length of a pattern to a maximal length. Höppner [2001] introduced a method using Allen's relations to mine rules in symbolic interval sequences, in which the time series are restricted by a sliding window, and the patterns are mined using an Apriori algorithm. In contrast to Kam and Fu [2000] rules are generated and their interestingness is measured based on the intermediate confidence within the time intervals. Höppner uses a k^2 sized matrix to represent the relations of a k intervals sized pattern. Additionally, Höppner proposes how to abstract the patterns or make them more specific. In addition, Papapetrou et al [2005] rediscovered the method of mining time intervals using Allen's relations. However, their contribution was in presenting a new mining method consisting on the SPAM sequential mining algorithm, which results in an enumeration tree which spans all the discovered patterns.

2.3.2 Mining Coincidence of Time Intervals

Recently, an alternative method to Allen's relations based mining methods, was proposed by Mörchen's, which is the technique we focus on in this paper. As we explained earlier, all the mining methods which we described in section 2.3.1, consisting on Allen's temporal relations, suffer from the failures listed earlier. In addition to his *TSKR* temporal knowledge representation method, Mörchen proposed a mining algorithm, called TSKM, to mine Tones, Chords, and Phrases [2006a,b]. The input for mining coincidence in the form of Chords is a set of Tones with the respective symbolic interval sequence. A Chord is observed if a set of coinciding intervals exceed a minimal length, and frequent if they all exceed a minimal level of support threshold. To mine Phrases, a partial order mining algorithm is required, which is similar to mining *Episodes* of instantaneous temporal data. However, unlike in Episodes in which they consist on time points, here is the input is time intervals. Mörchen extended the existing CHARM algorithm for mining Episodes. The approach was evaluated on a dataset of roller bladders, and analyzed by a sport physician, arguing that this approach is better than Allen's temporal knowledge representation.

In this study we applied Mörchen's mining method on a clinical dataset to examine the method in a new setup. While Morchen in his experiments tried it on a single dataset, which was split into time windows, in our study the dataset is constructed from independent patients in which we want to discover repeating Phrases. Additionally, we wanted to examine Mörchen's method in the light of his criticisms on Allen's representation based mining methods.

3 Methods

3.1 Abstracting Time Series

To transform the time series into time interval (Tones) series we performed *state* temporal abstraction, in which the time series are discretized into several states based on categories given from an expert or from a data driven computational source, as a preprocessing stage. Each time series went through three types of state abstraction: discretization given by a human expert (physician) and data driven methods using *SAX* [Lin et al, 2003] and *Persist* [Mörchen, 2005]. We briefly describe these approaches.

3.1.1 Symbolic Aggregate AproXimation

Recently the Symbolic Aggregate aproXimation (SAX) has been presented [Lin et al, 2003], which is based on the Piecewise Aggregate Approximation (PAA) [Keogh et al, 2001]. The PAA is a *dimensionality*[1] *reduction* method proposed for the problem of similarity search in large time series databases, in which it is crucial to reduce the amount of time units which represents the time series for the purpose of efficient manipulations, while maintaining a proper approximation commonly made through satisfying the *lower bounding* criterion introduced by Faloutsos et al [1994]. In the dimensionality reduction process a time series of n dimensions is transformed into N dimensions ($N<<n$). PAA enables the reduction of the dimension of a time series through splitting the time series into fixed length frames which are represented by the mean of the values within the frame. SAX first uses PAA to reduce the dimensional representation, then after normalizing the resulted values to the mean of zero and variation one, achieves interval representation by discretizing the normalized value range into equal sized areas ender the Gaussian curve, similarly to EFD, resulting in alphabetical symbols for each state. The frame size and the alphabet size create a tradeoff between efficiency and approximation accuracy. SAX is the first symbolic representation of time series with an approximate distance function that lower bounds the Euclidean distance. While SAX is one the first discretization methods designed specifically for time series data, the temporal aspect (order of values) of the data are taking into account only in the preprocessing stage of the PAA, thus not being an explicitly temporal method.

3.1.2 Persist

In a recent study Mörchen and Ultsch [2005] proposed *Persist*, a new *univariate* discretization method designed specifically for the purpose of knowledge discovery in time series, which for the first time explicitly considers the order of the values in the time series. Given a set of possible (discrete) symbols $S = \{S_1,\ldots,S_k\}$ of a time series of length n, Persist computes the marginal probability $P(S_j)$ of a symbol S_j and the transition probabilities given in a $k\times k$ matrix $A(j,m) = P(s_i=S_j|s_{i-1}=S_m)$, in which the self transitions are the values on the main diagonal of A. In this approach the assumption is that if there is no temporal structure in the time series, the symbols can be interpreted as independent observations of a random variable according to the marginal distribution of symbols, thus, the probability of observing each symbol is independent from the previous state, i.e. $P(s_i=S_j|s_{i-1},\ldots,s_{i-m}) = P(S_j|s_{i-1})$. Based on this Markovian model if there is no temporal structure, the transition probabilities should be close to the marginal probabilities, otherwise if the states show persistence behavior, which is expected to result in long time intervals, the self transition probabilities will be higher than the marginal probabilities. The Persist algorithm is based on a measure based on the Kullback-Leibler Divergence, which indicates which cutoffs lead to long time intervals. The method is compared to common discretization methods, such as EQW, SAX, HMM and more simple ones, and results in higher accuracy [Mörchen, 2006].

However, Persist assumes that any time series come from uniform sampling, in which the duration between each time point is fixed, which is not always the situation, especially in "*slow*" domains (sampled infrequently and commonly manually), such as the medical domain or other in which the sampling is made manually in varying periods of time. Thus, a more generalized framework should be developed which considers the distance among the time points with the time series.

3.2 Data Set

An ICU dataset was used of patients who underwent cardiac surgery at the Academic Medical Center in Amsterdam, the Netherlands, in the period of April 2002-May 2004. Two types of data were measured: *static data* in-

[1]Dimensionality in the context of time series refers to the amount of time units a time series is represented by, unlike in statistics and machine learning, in which it refers to the amount of independent variables (features).

cluding details on the patient, such as *age*, *gender*, *surgery type*, whether the patient was mechanically ventilated more than 24 hours during her postoperative ICU stay, and *temporal data*, measured each minute along the first 12 hours of the ICU hospitalization, including: mean arterial blood pressure (ABPm), central venous pressure (CVP), heart rate (HR), body temperature (TMP), and two ventilator variables, namely fraction inspired oxygen (FiO2) and level of positive end-expiratory pressure (PEEP). The data contains 664 patients, among which 196 patients were mechanically ventilated for more than 24hr (29.5%). The objective in this case study on the ICU dataset was to discover common temporal patterns for patients who were mechanically ventilated for more than 24hr for patients who were detubated within 24hr.

3.3 Evaluation Measures

We used the unsupervised TSKM mining method (2.2.1) which results with a set of Phrases, having each a support value above a given threshold. Evaluating knowledge discovered from a mining process is challenging since it is hard to estimate the quality of the discovered knowledge in quantitative terms, such as accuracy in classification. Since in our task the objective was to discover the common patterns of the two types of patients, we defined two measures to estimate the distance among two sets of Phrases. The first measures the pair-wise distances among each pair of Phrases in both sets, and the second is based on the minimal pair-wise distances.

Real examples of Chords and Phrases discovered in this study are presented in figure 3 and 4 respectively. Chord C1 describes the duration, in which ABPm, CVP, and HR at level_2 and TMP level_3 coincide. C4 describes the duration, in which FiO2 at level 2 and PEEP and TMP at level 3. These Chords appear in the examples in Figure 4, which presents two examples of Phrases. We will use these examples for the explanation of the measures.

C1 (#292, 0.80):
hf-ABPmTones is Level_2
hf-CVDTones is Level_2
hf-HRTones is Level_2
hf-TempTones is Level_3

C4 (#251, 0.16):
hf-FiO2Tones is Level_2
hf-MpeepTones is Level_3
hf-TempTones is Level_3

Figure 3 Example of two chords, C1 and C4. The support of C1 is 0.8. The support of C4 is 0.16.

Let $P^+=\{P^+_1, P^+_2,.., P^+_n\}$ and $P^-=\{P^-_1, P^-_2,.., P^-_m\}$ be the set of Phrases discovered from the patients, who were mechanically ventilated for more than 24 hours, and the ones who received ventilation less than 24 hours, respectively. We define a distance measure $d(P_i,P_j)$ among two (single) Phrases, inspired by measures which are commonly used to compare graphs.

$$d(P_i,P_j)=\frac{2\cdot I(|E(P_i)|,|E(P_j)|)}{|E(P_i)|+|E(P_j)|} \quad (1)$$

Where $I(E(P_i,)E(P_j))$ is the amount of common directed edges in P_i and Pj, and |E(P)| is the total number of directed edges in Phrase P. We double $I(E(P_i,)E(P_j))$ to have a measure in [0,1] range.

As an example for the calculation of $d(P_i,P_j)$ we will use the two Phrases presented in figure 4. The number of mutual directed edges is 2: The edges 1→8 and the edges 8→9. The total number of edges in *P4* is 6 and in *P5* is 4. Thus, $d(P4,P5)$ equals $2*2/(6+4) = 0.4$.

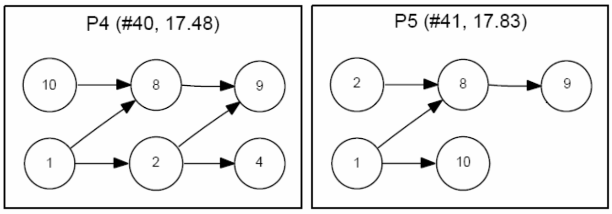

Figure 4 Example of two phrases, P4 and P5. The nodes (directed edges) are the chords, having an id, and connected according to their partial order.

3.3.1 Cross Product Distance

To measure the distance among the two sets of Phrases we measure the pair-wise distances of all the *n*×*m* pairs of Phrases. We started by computing the mean of all of the *nxm* pair-wise distances, as the distance between P^+ and P^- which resulted in $CPD_1(P^+,P^-)$ presented in equation 2.

$$CPD_1(P^+,P^-)=\frac{\sum_{p^+\in P^+}\sum_{p^-\in P^-} d(p^+,p^-)}{|P^+|\cdot|P^-|} \quad (2)$$

The obvious drawback of D_1 is the lack of consideration of the *support* of each Phrase. The intuition guided us in the extension of CPD_1 was that the support value of a Phrase should be considered as a weight representing its significance among the others. Thus, each pair of Phrases CPD_1 measure is multiplied by their average of supports values, which represents their weight in the final distance measure among the two sets of Phrases presented in CPD_2 measure, as shown in equation 3.

$$CPD_2(P^+,P^-)=\frac{\sum_{p^+\in P^+}\sum_{p^-\in P^-}\left[d(p^+,p^-)\left(\frac{p^+_{\text{sup}}+p^-_{\text{sup}}}{2}\right)\right]}{\sum_{p^+\in P^+}\sum_{p^-\in P^-}\frac{p^+_{\text{sup}}+p^-_{\text{sup}}}{2}} \quad (3)$$

3.3.2 Minimal Distance

In addition to the cross product distance we defined a more compact and extreme measure, called *minimal distance*, in which we measure the distance for each Phrase with the closest Phrase from the other set, as shown in equation 4.

$$MD_1(P^+,P^-)=\sum_{i=1}^{k}\min[d(P_i^+,P_j^-)], 1\le j\le m \quad (4)$$

Note that in this measure, several Phrases from P^+_i can be coupled to the same P^-_j. Similar to CPD_2 we extended MD_1 to MD_2, in which the support values are used again to represent the importance of both Phrases.

3.4 Experimental Plan

To compare between the knowledge discovered from the positive and the negative patients we used Mörchen's [2006a] TSKM method in a slightly different approach. While typically the method computes the Tones, Chords, and Phrases from the entire dataset, since here the dataset was split into two classes and in order to be able to compare the discovered Phrases, we discretized the time series into Tones and discovered Chords based on the entire dataset. Then the Phrases were discovered separately for each group of the positive and negative patients, which yielded eventually in two sets of Phrases, P^+ and P^-, which were constructed from the same set of Chords (and Tones).

This discovery process was applied to the datasets after applying three types of preprocessing, in which the high frequency variables abstracted into Tones, including human expert (EdJ, the intensive care physician involved in the study), SAX, and Persist. The human expert discretization is provided in Table 1. The temporal data was abstracted according to the cut points presented in the table.

Finally, each type of a discretization yielded in two sets of Phrases, which we wanted to measure the distance among them to estimate the expected detection accuracy. The discretization which would result with the highest level of distance was expected to be the results in the best separation of the knowledge discovered from both groups.

Table 1: Cut-points determined by human expert.

Variable	Human expert cut-points
hf-ABPm	60,90
hf-CVP	5,17
hf-FiO2	41,60
hf-HR	60,110
hf-PEEP	7.333
hf-TMP	35.5,38.5

4 Results

We first refer to the preprocessing stage, in which the time series were abstracted to Tones. We measured the results of the application of each abstraction method to the time series by the mean and standard deviation of the resulted time intervals (tones), as shown in table 2.

Table 2: The mean length of the intervals for each variable and each discretization method

Variable	Human	SAX	Persist
hf-ABPm	17.03±78.17	8.78±39.98	21.98±92.55
Hf-CVP	15.6±92.85	8.54±38.34	25.1±186.25
Hf-FiO2	48.44±244.85	49.58±225.32	126.97±348.93
hf-HR	30.91±205.98	12.45±83.37	31.04±199.47
hf-PEEP	117.33±346.6	35.89±191.49	76.07±282.06
Hf-TMP	174±498.9	73.03±241.37	179.72±509.43

In most cases the Persist method achieved the longest intervals. However, note that for the PEEP measure the human expert discretization achieved much longer intervals than other discretization techniques.

Table 3 presents the average number of intervals and number of Chords for each discretization method. As expected the methods having low number of time intervals have also the longest length in the results presented in table 2.

While, a relatively significant difference in the average number of intervals was observed, where Persist has the minimal amount, then the human expert and finally SAX with the largest amount of intervals, the amount of the discovered Chords was quite similar.

Table 3: Average number of intervals and number of chords for each discretization method.

Discretization	Avg. # of intervals	# of Chords
Human Expert	13,171.16	12
SAX	25,853.50	11
Persist	9,988.83	10

Table 4 presents the amount of Phrases discovered based on each discretization method. The Phrases where discovered separately from positive and negative patients, thus their number is different for each group.

Table 4: Number of Chords and Phrases for each discretization technique.

Discretization	# of P^+	# of P^-
Human expert	23	32
SAX	17	33
Persist	21	16

Although we defined two types of measures D_1 and D_2, in which the support value is considered to represent the relative importance of a Phrase, we present only the D2 types. This is as a result of the lack of room and the magnitude of the results was the same. Table 5 presents the D_2 values computed for each discretization method. Persist results with the highest distance among the positive and negative patients, according to the D_1 measure. SAX resulted with the lowest distance, and the human expert based abstraction yielded in a relatively high separation level.

Table 5: Distance values $MD_2(P^+,P^-)$ and $CPD_2(P^+,P^-)$ for each discretization method.

Discretization	MD_2	CPD_2
Human expert	111.37	0.21
SAX	38.46	0.08
Persist	172.87	0.30

5. Discussion

We presented the problem of mining time series represented by time intervals through temporal abstraction. We presented Mörchen's TSKM and applied it to the problem presented in the ICU dataset, in which there is a dataset of patients classified into two classes. The temporal abstraction of the data was made based on three inputs: a *domain expert*, and two data driven discretization methods: *SAX* and *Persist*. The TSKM mining method was applied on the resulted time interval series. First, Chords were discovered from the *entire dataset* to enable comparison of the further discovered Phrases, which were discovered separately from each class of patients. according to their duration being mechanically ventilated. Knowledge was discovered in the form of a set of Phrases from each class of patients. To evaluate the differences in the discovered knowledge we defined a distance measure among two Phrases and two sets of Phrases.

In this study, the cutoffs in the three discretization methods were fixed along the entire time series. This is a limitation of the study, as during the postoperative ICU stay of cardiac surgical patients, the definitions of 'normality' and 'abnormality' change during the twelve-hour period for variables such as PEEP and TMP [Verduijn et al, 2005] (creating a context [Shahar, 1997]). In this study, we used the mean value of these cut-points for these variables in the human discretization method. Discretization with dynamic cut points is an important topic of our future work. However, note that while these contexts are relevant for physicians' for diagnosis and prognosis purposes, it might not be always (and for any variable) relevant in the task of temporal knowledge discovery or classification tasks.

The results of the study indicate that the data which was discretized by the Persist method resulted with the highest level of separation, followed by the human expert and SAX. While the human expert descretization was expected to be more meaningful, it is not necessarily expected to lead to better knowledge discovery since it was defined for diagnosis purposes and not knowledge discovery. SAX while aiming to be a discretization method for time series does not explicitly considers the temporal order of the time series, which might result in the pure results.

Referring to Mörchen's method in the light of his criticisms [Mörchen, 2006a,b] on Allen's relations, our observation is that the TSKM suffers from part of the aspects indicated by Mörchen on Allen's relations. While it does not suffer from robustness, having only two operators representing *coincidence* and *synchronity* (*partial order* of chords), similar to Allen's *equal* and *before* respectively, it is not expressive enough to show the actual relations among the time intervals. Additionally, it is ambiguous as well as Allen's relations since Chords discovered can have different durations which may reflect different meanings, as well as in Phrases, in which the gap duration among Chords in a Phrase may vary, similar to the Allen's *before*.

8. Conclusions and Future Work

In this study we showed the ability to analyze time series using temporal abstraction. We applied the TSKM on a dataset including two classes of patients. Phrases were discovered from each class and the distance among the discovered sets were measured. As future work, we would like to measure the accuracy of the classification using the discovered Phrases. In addition we develop an Allen based time interval mining method which is expected to overcome the criticisms presented by Mörchen.

Acknowledgements

For this study we used the implementation of the TSKM and the discretization methods in Matlab, provided by Mörchen on his website.

References

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

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

[Kam and Fu, 2000] P. S. Kam and A. W. C. Fu, Discovering temporal patterns for interval based events, In Proceedings DaWaK-00, 2000.

[Lin et al, 2003] Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13.

[Mörchen and Ultsch, 2005] F. Mörchen, and A. Ultsch, Optimizing Time Series Discretization for Knowledge Discovery, In Proceeding of KDD05, 2005.

[Mörchen, 2006a] F. Mörchen, Algorithms for Time Series Knowledge Mining, Proceedings of KDD-06, 2006.

[Mörchen, 2006b] F. Mörchen, A better tool than Allen's relations for expressing temporal knowledge in interval data, Proceedings of the KDD06 Workshop on Temporal Data Mining 06, 2006.

[Moskovitch and Shahar, 2005], R. Moskovitch, and Y. Shahar, Temporal Data Mining Based on Temporal Abstractions, (IEEE) ICDM-05 workshop on Temporal Data Mining, Houston, US, 2005.

[Papapetrou et al, 2005] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos, Discovering Frequent Arrangements of Temporal Intervals, Proceedings of ICDM-05, 2005.

[Sacchi et al, 2006] L. Sacchi, M. Verduijn. N. Peek, E. de Jonge, B. de Mol, R. Bellazzi. Describing and modeling time series based on qualitative temporal abstraction. Workshop notes of the IDAMAP workshop, 2006.

[Shahar, 1997] Shahar, Y., A framework for knowledge-based temporal abstraction, Artificial Intelligence, 90(1-2):79-133, 1997.

[Verduijn et al, 2005] M. Verduijn. A. Diagliati, L. Sacchi, N. Peek, R. Bellazzi, E. de Jonge, B. de Mol. IC prediction from patient monitoring data: a comparison of two temporal abstraction procedures. Proceedings of the AMIA 2005 Annual Symposium, 2005.

[Villafane et al, 2000] R. Villafane, K. Hua, D Tran, and B. Maulik, Knowledge discovery from time series of interval events, Journal of Intelligent Information Systems, 15(1):71-89, 2000.