Classification of ICU Patients via Temporal Abstraction and Temporal Patterns Mining

Robert Moskovitch[1], Niels Peek[2], Yuval Shahar[1]
[1]Medical Informatics Research Center, Department of Information Systems Engineering,
Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel
[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
{robertmo, yshahar}@bgu.ac.il, {n.b.peek }@amc.uva.nl

Abstract

The use of unsupervised mining, such as association rules and sequential mining was used in the past as a preprocessing step to the classification task. The recent growth in the use of temporal abstraction for temporal knowledge discovery from multivariate temporal data, through time intervals mining, is proposed to be employed for the classification of multivariate temporal data. We present KarmaLego, an efficient time intervals mining which discovers non-ambiguous patterns based on a flexible version of Allen's relations. Later the discovered patterns are used as features which represent the relations of the multiple variables. Several settings for the suggested approach are examined through a rigorous evaluation.

1 Introduction

With the increase in patient data logged overtime and the reduction of storage costs, classification of multivariate temporal data is very useful in many tasks. Temporal data provides more detailed description of the patient situation which is expected to lead to better diagnosis and decisions, however, analyzing multivariate temporal data, especially for classification is challenging, often made without consideration of the relations among the multivariate time series. Commonly the representation of time stamped data is made using time windowing, in which features such as the mean, minimal or maximal value is extracted. More sophisticated approaches represent the time window with discrete values, such as qualitative mean (e.g., low, medium, high) [20]. Other approaches transform the time series to the frequency domain by Furrier transform [2] for example.

However, determining the right time window size is commonly problematic. Then 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. Additionally, these approaches do not allow expressing the relations among the multi variables along time.

In this paper we present an approach, in which the time series are abstracted into time intervals series [18] and frequent temporal patterns of the multivariate time interval series are discovered. Then the discovered patterns are used as features to represent the classified entities. We demonstrate our approach in the domain of Intensive Care Unit. We start by surveying the background. In the methods section we describe the descretization methods we used for the temporal state abstraction and KarmaLego – the time intervals mining method. Finally, we describe the entire procedure through the rigorous evaluation and results on the ICU dataset.

2 Background

In our approach we first abstract the time series to *states*, then mine them to discover frequent patterns, which are later used as features in the classification task. Thus, in this section we refer to the domain of Temporal Abstraction [18], we survey the recent development in time intervals mining and refer to the concept of classification based on temporal patterns.

2.1 Temporal Abstraction

Temporal abstraction is the aggregation of time series to a summarized and a more comprehensive representation for a human or productive for further data mining tasks. The task of temporal abstraction corresponds to the task of segmenting the time series, having a meaningful symbol for each segment. Segmenting time series [7] is a representation of time series in a piecewise linear representation, which is the approximation of a time series length n with k straight lines, usually $k<<n$.

Knowledge-based temporal abstraction (*KBTA*), as presented by Shahar [18], infers domain-specific interval-based abstractions from point-based raw data, based on a formal domain-specific abstraction and interpolation. However, although the KBTA method applies the temporal-abstraction knowledge to create abstractions that are meaningful to the domain expert, such knowledge is not always available. Moreover, the knowledge provided by the domain expert is not always pertinent to the data-mining task, such as classification, but rather to the clinical expert's routine activities, such as diagnosis or therapy [20]. Thus, several automated data-driven methods, which provide abstractions are less semantically meaningful, but can be potentially useful for data mining and finally classification, which we will present here. This can be achieved for state abstraction, by simply applying unsupervised discretization methods, such as *equal width discretization* and *equal frequency discretization*. Other methods are *k-means clustering* [8], in which the time series values are grouped into *k* clusters from which the states can be deduced. More sophisticated methods refer to the temporal order of the time points in the time series, include: Symbolic Aggregate approXimation (SAX) [7] is a method for symbolic representation of time series and Persist [8].

2.2 Mining Time Intervals

Mining time intervals is a relatively young research field. One of the earliest studies in the area is that of Villafane et al. [20], which searches for *containments* of intervals in a multivariate symbolic interval series. Kam and Fu [6] were the first to use all of Allen's temporal relations [1]. Höppner [2] was the first to define a non-ambiguous representation of Allen's-based time intervals patterns by a k^2 matrix to represent all of the pairwise relations within a *k*-intervals pattern. Unlike Höppner's naïve mining method, Papapetrou et al. [15] presented a mining method, which results in an *enumeration tree* that enumerates all the symbols and their possible relations combinations, presenting a BFS, DFS and Hybrid approaches, using only five temporal relations. . Additionally, they relaxed the temporal relations with the notion of an epsilon, which we extended to all Allen's relations (fig 1).

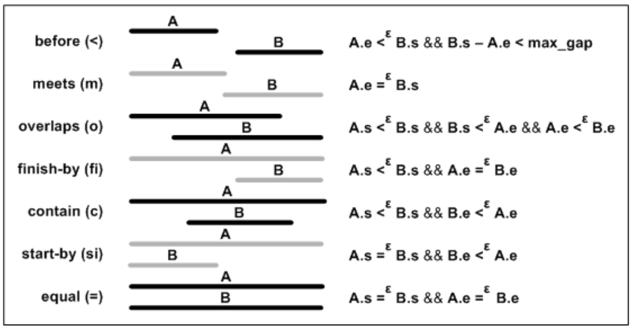

Figure 1 - A flexible extension of Allen's seven relations using the same Epsilon value for all relations.

ARMADA presented recently by Winarko and Roddick [23], uses a candidate generation and mining iterations approach. Mörchen [11] proposed an alternative to Allen's relations-based methods, in which time intervals are mined to discover coinciding symbolic time intervals, called *Chords*, while repeating partially ordered chords are called *Phrases*. Sacchi et al [18] use abstracted time series to find temporal association rules by generalizing Allen's rules into a relation called PRECEDES. In **Error! Reference source not found.** we present KarmaLego in more detail and evaluate its performance in comparison to Papapetrou's [15] and ARMADA's [23] methods.

2.3 Classification Using Temporal Patterns

Using unsupervised methods for knowledge discovery as a preprocessing step for classification was suggested first by Liu et al [8]. The authors used association rules mining to discover frequent association rules of each class, called Class Association Rules which later used as features in the classification task. Since then several approaches were proposed to employ association rules for classification, and later sequential mining was employed as well. Further to the recent growth of interest in the use of temporal abstraction as a preprocessing stage and time intervals mining for multivariate temporal knowledge discovery [12] , the idea of employing time intervals related patterns for multivariate temporal data classification was suggested very recently in [2]. The authors used domain expert for state abstraction and gradient abstraction [19] (e.g., increase, decrease) and further mined the intervals for complex patterns consisting on the before and overlap Allen's relations to overcome the lack of robustness in Allen's relations. Finally a Boolean representation of the temporal patterns is used, according to their existence in the classified entity.

3 Methods

3.1 State Abstraction

To abstract the temporal data into states we used a Knowledge Based approach, in which the data was abstracted according to the domain expert definitions, coming from his domain of diagnosis and treatment as described in [20]. As alternative to the knowledge based approach we used four discretization methods which are the focus of this section and this study.

Equal Width Discretization (EQW) determines the cut-points by dividing the value range into equal width bins. The number of values in each bin is based on the distribution of the values.

Equal Frequency Discretization (EQF) divides the value range into bins, so that the frequency of values in each bin is equal, thus, the number of values in each bin is equal.

Symbolic Aggregate Approximation (SAX) [7] consists of two steps. The first step is the Piecewise Aggregate Approximation, in which the granularity of the time series is reduced by averaging several time points into a single valued time point. The second and main step of the SAX method is the discretization of the PAA output. The discretization is based on the assumption that normalized time series have a Gaussian distribution, and on the desire to produce equal probability states. The time series is therefore normalized and discretized into a fixed number of states according to predetermined cut-points which produce equal-sized areas under a Gaussian curve.

Persist [10] is a univariate discretization method designed to maximize the mean duration of the resultant time intervals. The method measures the marginal probability P(Sj) and the self-transition probability $A(j,j) = P(s_i=s_j|s_{i-1}=s_j)$, where $S = \{s_1,\ldots,s_k)$ is the set of resultant states (symbols), assuming that k states are desired. The algorithm searches for the bins divisions which maximizes the self transition probabilities in comparison to the marginal probabilities, which is expected to maximize the mean duration of the resultant intervals, for the given number of states.

3.2 KarmaLego – Fast Time Intervals Mining

The discovery of temporal interval patterns is computationally highly demanding, since it requires generating all of Allen's seven basic temporal relations. To overcome this difficulty, we developed KarmaLego, a fast algorithm which enumerates all of the patterns whose frequency is above a given support threshold. Due to the lack of space in this paper, whose focus is on the use of KarmaLego for knowledge discovery in medical domains, we present the algorithm only briefly. A detailed report, including a rigorous comparison to several previous methods is available elsewhere [14]. KarmaLego is based on a flexible version of Allen's seven relations. This is achieved by adding an epsilon value to all seven relations, as shown in figure 2. We also limit the before relation by a maximal allowed gap (see figure 2), as proposed by Winarko and Roddick [23].

Definition 1. To define a flexible framework of Allen's temporal relations in the KarmaLego, we define two relations on time-stamped (point-based) data. Given two time-points t_1 and t_2, $t_1 =^{\varepsilon} t_2$ *iff* $|t_2 - t_1| <= \varepsilon$ and $t_1 <^{\varepsilon} t_2$ *iff* $t_2 - t_1 > \varepsilon$.

Definition 2. A *symbolic time interval*, *I* = <*s, e,sym*>, is an ordered pair of time stamps, start-time (s) and end-time (e), and a symbol (*sym*), which typically includes an abstraction within a context (e.g., Moderate-Anemia-Adult-Female).

Definition 3. A *lexicographic symbolic time- interval series* is a time-interval series, sorted in the order of the start-time, end-time using the relations $<^{\varepsilon}$, $=^{\varepsilon}$ and a lexicographic order of the symbols, $IS = \{I^1, I^2, .., I^n\}$, such that

$$\forall I^i, I^j \in IS\,(i < j) \wedge ((I_s^i <^{\varepsilon} I_s^j) \vee (I_s^i =^{\varepsilon} I_s^j \wedge I_e^i <^{\varepsilon} I_e^j) \vee (I_s^i =^{\varepsilon} I_s^j \wedge I_e^i =^{\varepsilon} I_e^j \wedge I_{sym}^i < I_{sym}^j))$$

Definition 4. A non-ambiguous *lexicographic Time Intervals Relations Pattern P* is defined as $P = \{\breve{I}, \breve{R}\}$, where $\breve{I} = \{I^1, I^2,.., I^k\}$ is a set of *k* symbolic time intervals ordered lexicographically and $\breve{R} = \bigcap_{i=1}^{k}\bigcap_{j=i+1}^{k} r(I_i, I_j) = \{r_{1,2}(I_1,I_2), r_{1,3}(I_1, I_3), .., r_{1,k}(I_1, I_k), r_{2,3}(I_2, I_3), r_{2,4}(I_2, I_4), .., r_{2,k}(I_2, I_k), \ldots, r_{k-1,k}(I_{k-1}, I_k)\}$, define all the relations among each of the $(k^2-k)/2$ pairs of symbolic time intervals in $\breve{I}$.

Definition 5. Given a database of $|E|$ entities, the *vertical support* of a TIRP *P* is denoted by the cardinality $|E^P|$ of the set E^P of distinct entities (e.g., different patients), having P at least once divided by the total number of the entities (e.g., patients) $|E|$: $ver\_sup(P) = |E^P| / |E|$.

When a symbol or a TIRP has vertical support above a minimal predefined threshold, we refer to it as *frequent* along the paper.

Definition 6. The *horizontal support* of a TIRP *P* for an entity e_i (e.g., a single patient's record) is the number of instances of the TIRP *P* found in e_i - $hor\_sup(P, e_i)$.

The *mean horizontal support* of a TIRP *P* is the average of the horizontal support of P for all the entities in E^P.

Problem Definition. Given a set of entities *E*, described by a symbolic time- intervals series *IS*, and a *minimum vertical support* threshold *min_ver_sup*, the goal is to find all the TIRPs whose frequency is above *min_ver_sup*.

KarmaLego – Fast Time Intervals Mining

The KarmaLego algorithm consists of two main steps. The first is called Karma[1], in which all the two-sized TIRPs – all the relations among two symbolic time intervals - are discovered, using a breadth-first search. In the second step, called Lego[2], the 2-sized TIRPs are used to recursively construct longer, more frequent TIRPs. In algorithm 1 (Karma), the first level of the enumeration tree is constructed based on the set of the sufficiently supported symbols T^1. Then all the 2-sized TIRPs are constructed by the symbols T^1 and related by the varying relations *R*. Then each frequent 2-sized TIRP *t* is extended using the Lego algorithm.

Algorithm 1 – Karma

Input: *db*; *min_ver_sup*; *epsilon*.

Output: T – an enumerated tree of TIRPs

T^2 – *the tree at second level*; InsVec – a vector containing all the TIRPs instances found in the data; P_vec – the parameters vector in a searched TIRP; Rel_vec – the relations vector of a TIRP.

1. $T^1 \leftarrow S \leftarrow min\_ver\_sup(db)$ //all frequent symbols in *db*
2. $T^2 \leftarrow T^1$ and enumerating all $s \in T^1$, above *min_ver_sup*
3. foreach $t \in T^2$
4. if t is above minimal vertical support
5. Lego(T^2, *t*, *3*, *min_ver_sup*, *S*)
6. end

Lego (algorithm 2) is called by *Karma* to extend each *t* in T^2, which recursively extends the TIRP. Thus, Lego receives a k-sized TIRP and extends it with all the possible symbols and relations among the additional intervals and the k intervals of the extended TIRP. Consequently, for each frequent new (extended) TIRP, Lego is recalled recursively, creating an enumeration tree like the one presented in Figure 4.

[1] Karma – The law of cause and effect originated in ancient India and is central to Hindu and Buddhist philosophies.

[2] LEGO – A popular game, in which modular bricks are used to construct different objects.

Algorithm 2 – Lego

Input: T^2, t, *level*, *min_ver_sup*, *S*

Output: void

1. foreach $s \in T^1$
2. foreach $r \in R$
3. Generate extended TIRPs $t^r{}_s$ from *t*, *r* and *s*
4. Search for supporting instances of t T^2
5. if(ver_sup($t^r{}_s$) > min_ver_sup)
6. Lego(T^2, $t^r{}_s$, level+1, min_ver_sup, S)
7. end

A major challenge in the candidate generation of the time-intervals pattern is the exponential growth of the number of relations in a TIRP, and as a result the number of candidates, with the number of k time intervals (def 5). Two main contributions make the KarmaLego algorithm significantly faster and more efficient [7]. First, a direct extension of the TIRPs is applied. Thus, extending a k-sized pattern to a k+1 sized pattern generates R^k candidates, unlike R^(k^2-k)/2 in Papapetrou's method [7,9]. Second, a naïve generation of candidates creates also logically contradictory TIRPs. To *not* generate such candidates, we exploit transitivity to eliminate such candidates [7].

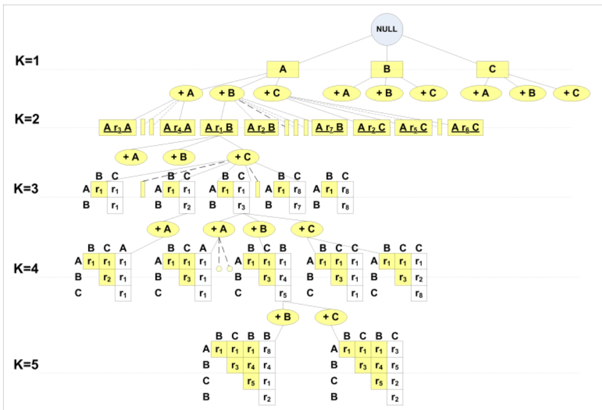

Figure 2 – A KarmaLego enumeration tree, in which the direct expansion of TIRPs is performed. Each node represents a TIRP as a half matrix of the temporal relations.

3.3 Bag of TIRPs

After discovering frequent TIRPs from the entities we can use them as features for the classification task. We adopt the idea of *bag-of-words* from the textual domain [22], in which terms are used to represent documents. In our case it is a *bag-of-tirps*, which are used to represent the patients for the classification task. The discovery process through KarmaLego results with a list of TIRPs, having its *vertical support*, which describes in how many patients it was discovered, and *horizontal support* for each patient, which describes how many times it was found for a specific patient within the period of time in the dataset.

When attempting to classify based on TIRPs a table has to be created, in which the columns (features) are the TIRPs and the last is the label, and the rows are the patients. In this study we suggest two types of representation for a TIRP, related to a patient:

- *binary*, in which a TIRP is represented by 1 it exists and 0 otherwise.
- *horizontal support* (HS), in which the value is the number of instances of the TIRP were discovered in the patient multivariate temporal data.

3.4 Feature Selection

The reduction of the number of TIRPs (features) as often is required in classification procedures is even more essential here since the features (TIRPs) are sometimes extensions of other TIRPs and thus refer to a subset of patients of the extended TIRP, which creates dependency within the features. Thus, we used several filtering feature selection methods, in which each feature is ranked based on some criterion that measures the correlation to the class to estimate the classification potential. Finally we extracted the top ranked features. We used a simple method which is based on the Vertical Support (VS), Gain Ratio and Fisher Score.

- *VS* - Based on the *vertical support* of each TIRP we selected the TIRPs having the highest VS, which were expected to represent more patients.
- *Gain Ratio* - was designed to overcome a bias in the Information Gain measure, and which measures the expected reduction of entropy caused by partitioning the examples according to a chosen feature [8].
- *Fisher Score* - The Fisher score ranking technique calculates the difference, described in terms of mean and standard deviation, between the positive and negative examples relative to a certain feature [4].

4. Evaluation

To evaluate the approach of the classification through temporal patterns on the ICU dataset, we designed an experiment which includes several settings for each parameter.

4.1 ICU 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* including 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%).

4.2 Experimental Plan

We wanted to answer several questions.

1. What is the best Discretization method and how many states are required?
2. What is the best epsilon value for the mining?
3. What is the best TIRP representation?
4. What is the best feature selection method and Top selection?

We used the temporal data of the last *three* hours ninth till twelfth hours. All the variables were abstracted using the four discretization methods [EQW, EQF, SAX, Persist] into 3 and 5 states. Then each abstracted dataset was mined using KarmaLego with *maximal gap* (on the relation *before* – see fig 1) of 100 seconds, minimum vertical support of 20% and the TIRPs were restricted to not more than 5 intervals. The mining was performed with three epsilon values 0, 5 and 10. After that the discovered TIRPs were used to create a matrix of the patients using the two representations: binary and horizontal support. Then the features were selected by the three feature selection methods Vertical Support, Gain Ratio and Fisher Score into three top selection levels 50, 100 and 150. This resulted into 243 evaluation runs, in which we used several classifiers, but we report the results of Random Forest using weka Classification method, which outperformed the other, with 10 cross validation.

5. Results

In the results analysis we compare the mean accuracy values of all the experiments according to the parameters. The parameters are analyzed according to the procedure described earlier.

5.1 Discretization methods

Figure 3 presents the mean accuracy of all the runs relating to the each discretization method and the states number. In most of the discretization methods abstracting into 5 states outperformed the 3 states, except for Persist.

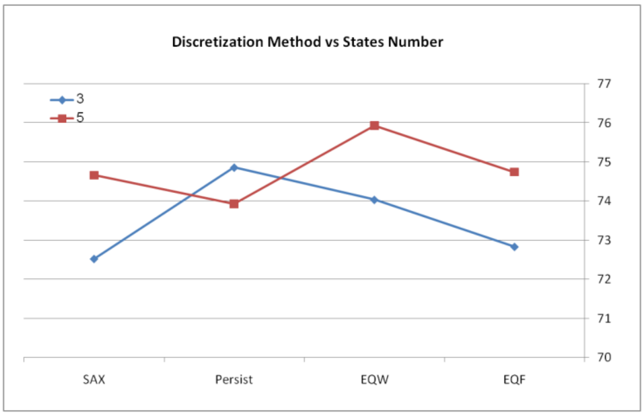

Figure 3 – Abstracting into five states outperformed for most of the methods, out of Persist.

5.2 Epsilon and TIRP representation

Figure 4 presents the mean accuracy for all the runs of the epsilon values and the TIRP representations. The Binary representation outperformed the HS. This indicates that the number of instances of the TIRP is being of less important to the classification task. While the Binary representation accuracy increased for lower epsilon it was slightly improved for the HS when the epsilon value increased. This is actually surprising since it was expected that having a larger epsilon will increase the robustness of the temporal relations and will enable more representative temporal patterns.

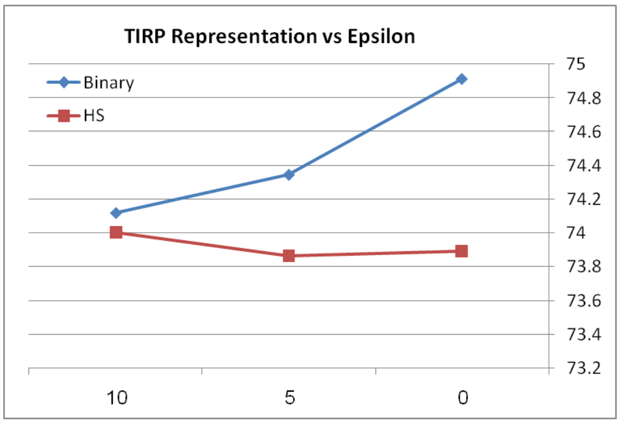

Figure 4 – The binary outperformed the HS representation, especially when the epsilon value was 0.

5.3 Feature Selection

The Fisher Score and the Gain Ratio outperformed the Vertical Support which selects the TIRPs with the highest vertical support. This can be explained by the imbalance of the classes, which might not represented by the TIRPs having the highest VS.

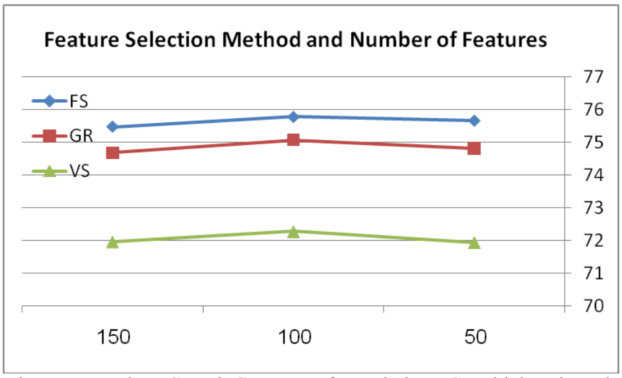

Figure 5 – The FS and GR outperformed the VS, which select the TIRPs with the highest VS.

We present in Table 1 the top six runs which brought the best accuracies. The top runs correspond to the mean accuracy analysis presented in the previous sections, which shows that the best settings for this dataset are abstracting using the EQW into 5 states, which is interesting since it is the simplest discretization method. Mining using low value of epsilon and using the Binary representation for the TIRPs. Finally to reduce the features both Gain Ratio and Fisher Score with 100 to 150 features were the best.

Table 1 – the top evaluation runs with the dominant settings in bold.

discM	**|s|**	**e**	**TIRP - Rep**	**|f|**	**FS**	**Acc [%]**
EQW	**5**	**5**	**Bin**	100	**GR**	79.64
EQW	**5**	10	**Bin**	**150**	FS	79.64
EQW	**5**	0	HS	**150**	**GR**	79.34
EQW	**5**	0	**Bin**	100	**GR**	79.34
EQW	**5**	**5**	**Bin**	**150**	FS	79.03
EQW	**5**	**5**	**Bin**	**150**	**GR**	79.03

6. Discussion and Conclusions

We presented an approach for classification of multivariate temporal data using time intervals patterns, which were discovered after abstracting the multivariate time series into time intervals. In addition to knowledge based state abstraction we used discretization methods. We presented an efficient time intervals mining algorithms, called karmaLego, for the discovery of non-ambiguous patterns consisting on a flexible version of Allen's temporal relations. The discovered patterns (TIRPs) are then used as features for the classification task. We presented two representation approaches, binary and horizontal support which uses the number of instances of the TIRP within the given temporal data. To reduce the number of features we used two commonly used feature selection measures and a measure consisting on the vertical support. Finally we used the Random Forest classification method for the classification task.

Our analysis shows that abstracting into five states was better for most of the methods out of Persist, which is motivated by creating long intervals (fig 3). The Binary representation though simpler outperformed the Horizontal Support, which can be explained by small values of HS and it might be too noisy. This happened especially for epsilon 0 (fig4). The feature selection measures were much better than the VS measure, which prefers TIRPs with high vertical support. This effect can be because this dataset is imbalanced (fig 5). Finally, we showed in table 1 the best runs with the best settings.

For future work we would like to perform another mining approach, in which each class patients are mined separately to discover its representative TIRPs, which can reduce the problem of imbalanced datasets. Then after discovering the TIRPs of each class to create a matrix based on their unification. Additionally, we plan to develop a discretization method that considers the class of each patient to perform an abstraction which maximizes the difference in the states distribution for each class, which is expected to discover different TIRPs for each class for better classification. Additionally we would like examine the use of smaller set of temporal relations which are more general to increase the number of discovered TIRPs.

References

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

[2]I. Batal., L. Sacchi, R. Bellazi, M. Hauskrecht, Multivariate Time Series Classification with Temporal Abstractions, Twenty-Second International FLAIRS Conference, Florida, 2009.

[3]R. N., Bracewell, The Fourier Transform and Its Applications, Boston: McGraw-Hill, 2000.

[4]Golub, T., Slonim, D., Tamaya, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and E. Lander, E., Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science, 286:531-537, 1999.

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

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

[7]J., Lin, E., Keogh, S., Lonardi, B., Chiu, 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, 2003.

[8]B. Liu, W. Hsu, Y. Ma, Integrating Classification and Association Rule Mining, In Proceedings of KDD-98, New York, 1998.

[9]Mitchell T. Machine Learning, McGraw-Hill, 1997.

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

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

[12] F. Mörchen, Unsupervised Pattern Mining From Symbolic Temporal Data, SIGKDD Explorations, 9 (1), 2007.

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

[14] R. Moskovitch, Y. Shahar, KarmaLego – An Algorithm for Fast Time Intervals Mining, Ben Gurion University Faculty of Engineering Technical Report 21843/2008, [http://medinfo.ise.bgu.ac.il/medLab/MembersHomePages/RobPapers/Moskovitch.KarmaLego-TechReport.pdf].

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

[16] D. Rajanish, Classification Using Association Rules, IIMA Working Papers 2008-01-05, Indian Institute of Management Ahmedabad, 2008.

[17] 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.

[18] L. Sacchi, C. Larizza, C. Combi, and R. Bellazi. Data mining with temporal abstractions: learning rules from time series. Data Mining and Knowledge Discovery, (15):217–247, 2007.

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

[20] M, Verduijn, L, Sacchi, N, Peek, R, Bellazzi, E, de Jonge, B, Mol, Temporal abstraction for feature extraction: A comparative case study in prediction from intensive care monitoring data. Artificial Intelligence in Medicine, 41(1): 1-12, 2007.

[21] 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.

[22] G., Salton, A., Wong, and C.S., Yang, A vector space model for automatic indexing. Communications of the ACM, 18:613-620, 1975.

[23] E. Winarko and J. Roddick. Armada - an algorithm for discovering richer relative temporal association rules from interval-based data. Data and Knowledge Engineering, 1(63):76–90, 2007.