Medical Temporal-Knowledge Discovery via Temporal Abstraction

Robert Moskovitch, Msc, Yuval Shahar, MD, PhD

Medical Informatics Research Center, Department of Information Systems Engineering

Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel

Abstract

Medical knowledge includes frequently occurring temporal patterns in longitudinal patient records. These patterns are not easily detectable by human clinicians. Current knowledge could be extended by automated temporal data mining. However, multivariate time-oriented data are often present at various levels of abstraction and at multiple temporal granularities, requiring a transformation into a more abstract, yet uniform dimension suitable for mining. Temporal abstraction (of both the time and value dimensions) can transform multiple types of point-based data into a meaningful, time-interval-based data representation, in which significant, interval-based temporal patterns can be discovered. We introduce a modular, fast time-interval mining method, KarmaLego, which exploits the transitivity inherent in temporal relations. We demonstrate the usefulness of KarmaLego in finding meaningful temporal patterns within a set of records of diabetic patients; several patterns seem to have a different frequency depending on gender. We also suggest additional uses of the discovered patterns for temporal clustering of the mined population and for classifying multivariate time series.

Introduction

The temporal dimension in medical data analysis, in particular, temporal relations among meaningful clinical concepts, is very important and requires explicit representation. Significant efforts have been made in this area, such as by capitalizing on knowledge-based temporal reasoning methods [11]. However, to apply these methods, a certain amount of *knowledge* about temporal relations must first be acquired. Acquiring such knowledge from medical experts often encounters serious difficulties, since these experts are not necessarily used to represent explicitly their knowledge in terms of complex patterns involving multivariate time-oriented clinical data.

In the light of this difficulty, we can turn to medical records, the option we explore in this study.

The increased use of electronic medical records presents a significant opportunity to discover knowledge from past time-oriented clinical data by using various data mining methods. Automated discovery of temporal knowledge could have significant implications, since such knowledge, although potentially quite valuable is often not explicitly available from a domain expert. However, discovering temporal knowledge presents several challenges: multivariate data is often acquired a-synchronously; thus, each variable is sampled in a different manner. Sampling might occur at a fixed frequency, often through electronic means, which may vary for each variable or at random periods, as often occurs in manual measurements. Frequently, certain time-stamped data points might be missing. Raw data might also be represented by time *intervals*, for example, the period over which a medication is administered. In order to discover knowledge, such as recurring patterns, all of the variables need to be similarly represented.

In the current study, we first propose, as a preprocessing stage, abstracting the raw, time-stamped multivariate data (e.g., time-stamped hemoglobin values) into time–interval-based abstract concepts (e.g., periods of moderate anemia) using *temporal abstraction* [11]. This step enables us to overcome the a-synchronicity problem and, in many instances, the missing values problem through the interpolation inherent in most temporal-abstraction methods. We then present a fast, time-interval mining method, called KarmaLego, and show how it can be used to discover temporal knowledge, i.e., knowledge about temporal relations among multivariate, interval-based concepts, for clustering the mined population, and for classifying multivariate temporal data.

Background

Temporal Abstraction

Temporal abstraction (TA) is the segmentation and/or aggregation of a time series into a succinct, symbolic representation, suitable for a human decision-maker, or for data mining. See Figure 1 for an example.

Knowledge-Based Temporal Abstraction

One way of performing TA is by relying on domain-specific TA knowledge, such as clinically meaningful cut-off values, to create interval-based TAs from point based raw data, by using the *Knowledge-based Temporal-Abstraction* (*KBTA*) method [11]. Thus, the cut-off values for determining symbolic states, such as "Low" or "Moderate Anemia" might be suggested, in a context-sensitive manner, by a domain expert (see Figure 1). (TA contexts are generated dynamically from the data, such as age, the gender, or other similar abstractions).

Four output classes of TAs can be generated by the KBTA method. A *state* (value) abstraction takes as input the values of one or more concept types and generates as output a discrete

value of the corresponding condition (e.g. high or low temperature). A *gradient* (first derivative sign) TA defines an interval during which the value of a parameter is changing (e.g., increasing or decreasing hemoglobin values). The *Rate* (first derivative amplitude) TA summarizes the rate of change, such as rapidly-changing blood pressure [11].

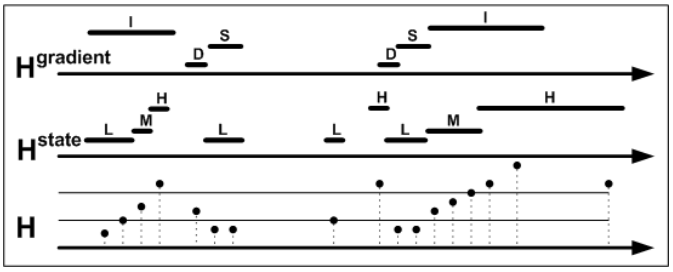

Figure 1. An example of a temporal abstraction of HbA1C (glycosilated hemoglobin) values into state-abstraction (Low, Medium, High) and gradient (trend)-abstraction (Inc, Dec, Stable) intervals.

Temporal Discretization

Although the knowledge-based approach is very useful for the discovery of clinically meaningful patterns, based on a medical knowledge base, it might be less effective for other tasks, such as classification, clustering, and prediction. Moreover, unlike *gradient* and *rate* abstractions, which can often be computed and defined mathematically, *state*-abstraction definitions are not always provided by an expert, either because the parameter is not often clinically used, or because there is no standard discretization for its value. Thus, an alternative to a knowledge-based human specification can be used, namely, a purely data-driven abstraction. *Temporal discretization* refers to the process of discretization of a time-series values, usually performed through unsupervised means, as a preprocessing step in transforming the time-stamped, raw-concept series into a set of symbolic, state-based time intervals. Several common discretization methods, such as *Equal Width* (which uniformly divides the ranges of each value) and *Equal Frequency Discretization*, do not consider the temporal aspect of the values; other methods, such as *Symbolic Aggregate approXimation* (*SAX*) (focusing on statistical discretization of the values) and *Persist* (maximizes the duration of the time intervals), explicitly consider the temporal dimension [4].

Mining Time-Intervals -Existing Methods

Mining time-intervals is a relatively young research field. Most of the methods use Allen's temporal relations [1] (shown in figure 2). One of the earliest studies in the area is that of Villafane et al. [13], which searches for *containments* of intervals in a multivariate symbolic interval series. Kam and Fu [3] were the first to use all of Allen's temporal relations. Höppner [2] was the first to define a non-ambiguous representation of time- interval patterns that are based on Allen's relations, by a k^2 matrix, to represent all of the pair-wise relations within a *k*-intervals pattern. Figure 3 presents an example of a *Time Interval Related Pattern* (*TIRP*). On the right there is a half matrix that presents all the pair-wise temporal relations which defines it in a non ambiguous, as we will define later in definition 4. Unlike Höppner's naïve mining method, Papapetrou et al. [9] presented the HFS mining method, which results in an *enumeration tree* that enumerates all the symbols using only five temporal relations.

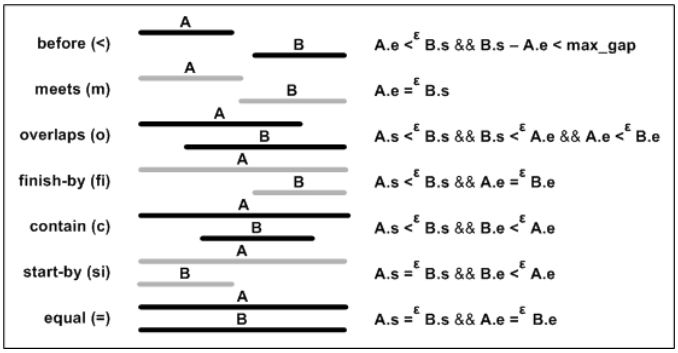

Figure 2. A flexible extension of Allen's seven relations using the same Epsilon value for all relations.

Subsequently, Papapetrou et al. relaxed the temporal relations with the notion of an epsilon value. ARMADA, by Winarko and Roddick [14], uses a candidate generation and mining iterations approach. Sacchi et al [10] use abstracted time series to find temporal association rules by generalizing Allen's rules into a relation called PRECEDES. Mörchen [5] proposed an alternative to Allen's relations-based methods, in which time intervals are mined to discover partially ordered coinciding symbolic time intervals. However, current methods are not sufficiently expressive and/or efficient when there is a need to consider TIRPS including all of Allen's relations, such as for predictive purposes, given the non-ambigious representation.

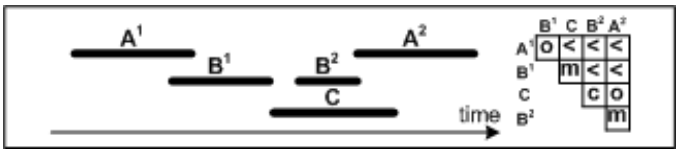

Figure 3. an example of a Time-Interval Related Pattern (TIRP), represented by a sequence of five lexicographically ordered symbolic time intervals and all of their pair-wise temporal relations.

Methods

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 [7]. 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 [14].

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

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.

[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 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.

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 $R^{\wedge}(k^2-k)/2$ 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].

The Diabetes Dataset

To demonstrate the usefulness of KarmaLego, we applied the method to the anonymous retrospective data of 2038 diabetic patients, collected over five years by Ben Gurion University's Soroka Academic Medical Center. The dataset includes monthly measurements of the (raw concepts) hemoglobina1c, blood glucose, and cholesterol values, and the medications the patients purchased, including diabetic (insulin-based) medications, statins, and beta blockers, represented by their defined daily dose (DDD). The measurements (raw concepts) in this particular domain and study were abstracted into three states, Low, Normal, High, according to a diabetes expert's definitions. The medications' DDDs were abstracted into three

states for each medication type, using equal width discretization. In addition, we abstracted the trends in the concepts' raw values into three symbolic values of the gradient (trend-direction) abstraction: Increasing, Stable and Decreasing. We ran KarmaLego with epsilon = 1 and maximal gap (of the Before relation) = 3, referring to the duration, in which the HBA1c measurements are relevant and have 5% minimal vertical support.

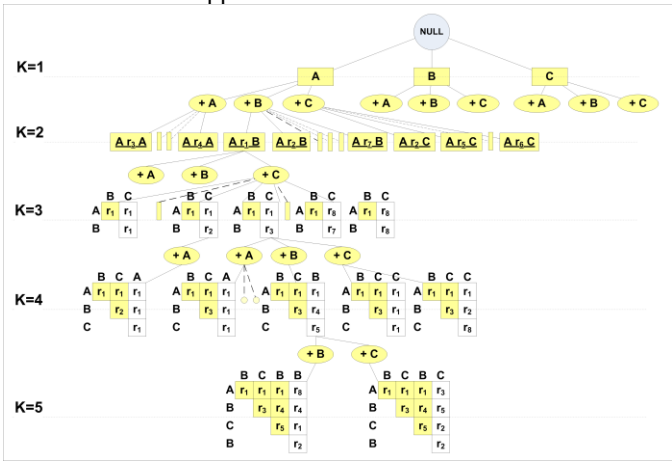

Figure 4. A KarmaLego enumeration tree, in which the direct expansion of TIRPs is performed. Each node represents a TIRP as a half matrix of the temporal relations.

The KarmaLegoV Visualization Tool

The most intuitive use of the patterns tree generated by KarmaLego is for temporal knowledge discovery, although other options are possible too, as we mention later. Thus, we developed *KarmaLegoV*, which makes it possible to visually browse the generated temporal patterns tree, from the root down to the leaves, through the intermediate extended TIRPs (nodes). The KarmaLegoV interface, shown in figure 5, presents at each moment the node (TIRP) that the expert is exploring. At the top-left frame, two lists of symbols are presented (sorted by various criteria, such as an interestingness measure): the symbols and relations that can extend the current TIRP, and, given a particular choice, further extending symbols and relations. The bottom-left frame presents a table of the explored TIRP instances, listed by patient identifier, including their properties, such as time duration and the patient static properties. The bottom-right panels present a graphical illustration of the average TIRP of the current type (top panel), and, below it, two instances of that TIRP, which were selected by the user at the bottom-left table. The top-right part presents the distribution of a selected static property (for example, age) for the patients in which the TIRP was found.

We present here an example from the diabetes dataset, based only on the HBA1c concept and the TIRPs of the Diabetes medication dose changes. Figure 6 presents a sub-tree of TIRPs, starting with the symbolic interval HBA1c increasing (HBA1c.inc), having a vertical support of 1702 patients, followed by three possible gradient abstractions of the diabetes medication dose: decreasing (DDD.dec), stable (DDD.stab) and increasing (DDD.inc). Extension of the current node by different temporal relations and/or symbol extensions imply a different TIRP. For example, the DDD.inc can start (almost) right at the end of the increasing HBA1c period, represented by the relation *meets* (*m*), and in another option can be *finished-by* (*F*) DDD.inc period.

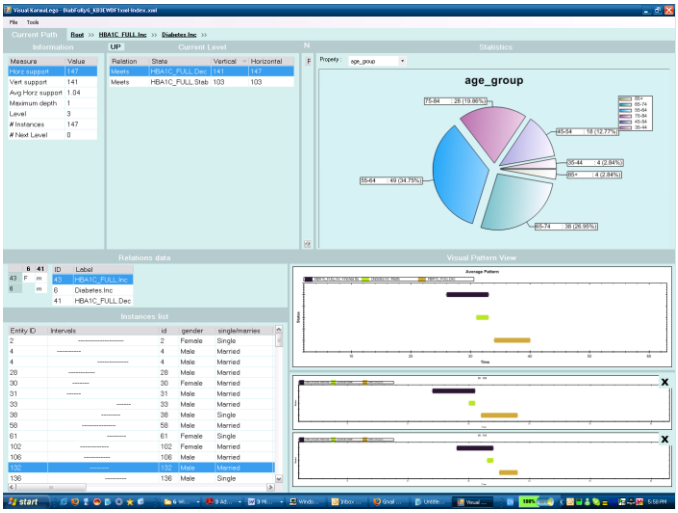

Figure 5. The KarmaLegoV tool presents to the domain expert the TIRP being browsed and the next TIRPs that can be browsed.

Each box presents a TIRP and shows at the top left corner its vertical support and its mean local horizontal support, as well as an illustration of the TIRP itself, including the mean duration of each symbolic interval. When descending deeper into the TIRP tree, the TIRPs are expanded into increasingly more specific patterns, whose vertical support is naturally decreasing. Interestingly, on the bottom left corner of the TIRP tree, one can note that when the diabetes medication dose was decreased, at the end of an increasing period of HBA1c, the HBA1c value was eventually decreased for a sufficient number of patients to cross the TIRP support threshold.

We have discovered more complex TIRPs, including also other types of symbolic intervals; we are currently in the process of examining these patterns, assisted by clinical experts. Finally, rules can be extracted from the discovered TIRPs which describe the probability of having the *next* symbolic interval, conditioned on all previous intervals that were encountered, when drilling down along the TIRPs tree (by using the vertical support of the next symbol).

Temporal Hierarchical Clustering

Each TIRP represents a cluster of patients who have similar temporal relations among their multivariate variables, such as a similar pattern of a reaction to a particular drug dose. Each cluster, or TIRP *P*, can be described by its *vertical support*, *mean horizontal support* and more interestingness measures. The extended TIRPs have equal or smaller vertical support; these are actually sub-clusters variations within the k-sized TIRP population. Another useful way to characterize a cluster of patients by their temporal pattern is by the distribution of their static, *non-temporal*, variables (e.g., gender, age). For example, it seems that several TIRPS are significantly more frequent in male rather than in female patients. Moreover, describing a temporal cluster by the dominant values of the static parameters of its patients will enable easy identification of potential patients. For example, we could characterize a group of patients knowing that a pattern of decreasing HBA1C following a decrease in the dose of the medication indicates mainly females in their sixth decade.

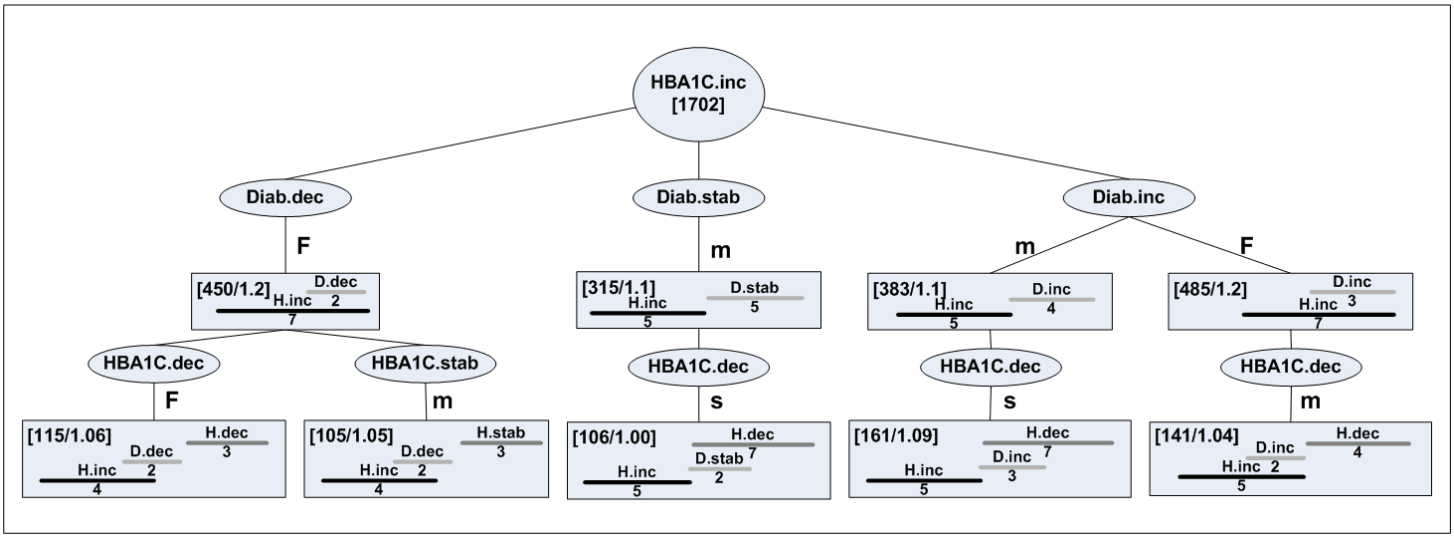

Figure 6. An example of discovered TIRPs from the Diabetes dataset. H = HBA_1C; DDD= Diabetes medication defined daily dose.

Classification of Multivariate Temporal Data

Classification of multivariate temporal data is often done using windowing techniques, in which representative statistical measures (mean, maximal value, minimal value) are extracted as features for the classification task [12]. This approach is often too general; the most representative time window size is unclear and the relationships among the varying variables are not represented. Using the KarmaLego framework we discover frequent TIRPs for each class and are then subsequently used as features for the classification task by standard ”static” classifiers (e.g., decision trees, support vector machines). We call this set of features a BAG-OF-TIRPs (inspired by the BAG-OF-WORDs from the textual domain). Several options for assigning values for the occurrence of a TIRP in a patient's data: *Binary* (e.g., True/False) and the TIRP's *horizontal support* for a given patient, which we successfully employed in the case of classification of extubation in the intensive care domain [8].

Discussion and Conclusions

We presented the idea of temporal knowledge discovery from multivariate temporal data via temporal abstraction, i.e., after abstracting the data into meaningful time intervals. The approach has significant potential, since temporal multivariate data often appear in varying granularities and frequencies. We presented KarmaLego – a fast time- intervals mining technique which extends TIRPs directly and exploits the transitivity of the temporal relations to eliminate generated candidates. We demonstrated the use of the method on a dataset of diabetic patients presenting a subtree of the discovered patterns. In addition, we described how the outcome of the mining can be used for hierarchical clustering of the mined population and for classification of multivariate time series of various types. Note that TIRPS could be automatically labeled, followed by an additional, second pass of temporal data mining.

In summary, automated discovery of complex temporal patterns from longitudinal clinical records is an exciting opportunity for extending current medical knowledge in new directions.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals, Communications of the ACM, 26(11): 832-843, 1983.
2. F. Höppner, Learning Temporal Rules from State Sequences, Proceedings of WLTSD-01, 2001.
3. P. S. Kam and A. W. C. Fu, Discovering temporal patterns for interval based events, In Proceedings DaWaK-00, 2000.
4. F. Mörchen, and A. Ultsch, Optimizing Time Series Discretization for Knowledge Discovery, In Proceeding of KDD05, 2005.
5. F. Mörchen, Algorithms for Time Series Knowledge Mining, Proceedings of KDD-06, 2006.
6. J. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. IEEE Transactions on Knowledge and Data Engineering, 4(14):750–767, 2002.
7. 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/Rob Papers/Moskovitch.KarmaLego-TechReport.pdf].
8. Robert Moskovitch, Niels Peek, Yuval Shahar, Classification of ICU Patients via Temporal Abstraction and temporal patterns mining, IDAMAP 2009, Verona, Italy, 2009.
9. P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos, Discovering Frequent Arrangements of Temporal Intervals, Proceedings of ICDM-05, 2005.
10. 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.
11. Shahar, Y., A framework for knowledge-based temporal abstraction, Artificial Intelligence, 90(1-2):79-133, 1997.
12. M. Verduijn, L. Sacchi, N. Peek, R. Bellazi, E. de Jonge,B. de Mol. Temporal abstraction for feature extraction: A comparative case study in prediction from intensive care monitoring data. In Artificial Intelligence in Medicine, 41, 112, 2007.
13. 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.
14. 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.