Data Min Knowl Disc (2007) 15:217–247
DOI 10.1007/s10618-007-0077-7

Data mining with Temporal Abstractions: learning rules from time series

Lucia Sacchi · Cristiana Larizza · Carlo Combi · Riccardo Bellazzi

Received: 15 June 2006 / Accepted: 17 May 2007 / Published online: 25 June 2007

Abstract A large volume of research in temporal data mining is focusing on discovering temporal rules from time-stamped data. The majority of the methods proposed so far have been mainly devoted to the mining of temporal rules which describe relationships between data sequences or instantaneous events and do not consider the presence of complex temporal patterns into the dataset. Such complex patterns, such as trends or up and down behaviors, are often very interesting for the users. In this paper we propose a new kind of temporal association rule and the related extraction algorithm; the learned rules involve complex temporal patterns in both their antecedent and consequent. Within our proposed approach, the user defines a set of complex patterns of interest that constitute the basis for the construction of the temporal rule; such complex patterns are represented and retrieved in the data through the formalism of knowledge-based Temporal Abstractions. An Apriori-like algorithm looks then for meaningful temporal relationships (in particular, precedence temporal relationships) among the complex patterns of interest. The paper presents the results obtained by the rule extraction algorithm on a simulated dataset and on two different datasets related to biomedical applications: the first one concerns the analysis of time series coming from the monitoring of different clinical variables during hemodialysis sessions, while the other one deals with the biological problem of inferring relationships between genes from DNA microarray data.

Responsible editor: Eamonn Keogh.

L. Sacchi (✉)· C. Larizza · R. Bellazzi
Dipartimento di Informatica e Sistemistica, Università di Pavia, Pavia, Italy
e-mail: lucia.sacchi@unipv.it

C. Combi
Dipartimento di Informatica, Università degli Studi di Verona, Verona, Italy

Keywords Temporal data mining · Rule discovery · Temporal abstractions · Biomedical time series

1 Introduction

A large number of data analysis problems deal with the interpretation of time series data. This is crucial both for studying experimental data, where one analyzes the time course of a set of variables under controlled conditions, and for analyzing the so-called process data, where a number of variables are collected to monitor the functioning of processes of interest. Examples of experimental data are temporal profiles of gene expression measured in cell lines under stressful conditions, while examples of process data are the outputs of sensors during the microprocessor synthesis in the electronics industry. Within this context, it is often of interest to detect the occurrence of certain temporal patterns, i.e., time intervals in which one or more time series assume a behavior of interest. Examples are the presence of activation patterns in the expression of gene transcription factors, or the increasing temperature in a chemical process. The selection of these kinds of pattern is typically based on the background knowledge available on the problem domain and their detection may nowadays rely on a large number of powerful algorithms.

When the knowledge available is mainly of qualitative nature, it is possible to resort to specialized tools, designed to extract qualitative patterns from data, such as knowledge-based Temporal Abstractions or qualitative reasoning (Shahar and Musen 1996; Hau and Coiera 1997; Kuipers 1986). Even more interestingly, although not widely investigated, is the automated search for temporal relationships between such patterns (Höppner and Klawonn 2002a; Kam and Fu 2000; Bellazzi et al. 2005). More particularly, the discovery of a *temporal precedence* between patterns may greatly help in generating hypotheses about the causal interactions between the problem variables, thus enabling an effective mining of temporal data. Moreover, the analysis of temporal precedence is also very important when looking at the long-run behavior of single variables to understand the timing and sequence of different patterns which may occur in one time series. Within this context, goals of different nature may be pursued. When prior knowledge about the interesting patterns is available, i.e., when we know what we are looking for, at least in terms of qualitative shapes of the temporal profiles, the problem is to find the temporal precedence occurrences. Such problem requires the formalization of the notions of pattern and of temporal precedence, and an efficient strategy to search for the occurrences of temporal precedence between given patterns in multivariate time series. A slightly different, and apparently more complex, goal is related to the *discovery* of *frequent* occurrences of temporal precedence between any pattern, in particular when the number of patterns or the measurement time span is high. In this case, in addition to the formalization of patterns and of temporal precedence, and to the use of an efficient search strategy, it is also necessary to define the *frequency* of the considered occurrences. Finally, the two above mentioned goals must be complemented with a pre-processing step, which may extract the patterns characterizing the available time series (Chiu et al. 2003). We must note that such goals require background knowledge on the problem domain. As a matter of fact, even an intuitive notion of precedence requires to specify what kind of relationship we are interested into: do we look for patterns which may have a temporal gap between each other? How long should be this gap? What happens if there is another pattern during the time gap? For this reason, the search will be by nature knowledge-driven.

In this article we will deal with the problem of extracting frequent temporal precedence occurrences between patterns by resorting to the framework of knowledge-based Temporal Abstractions, originally studied in Haimowitz and Kohane (1993) and Larizza et al. (1992) and formalized by Shahar (1997). Such a framework allows one to naturally define temporal patterns and to apply Allen's temporal operators (Allen 1984) in order to look for temporal precedence. Temporal abstractions will be coupled with an Apriori-like algorithm to search over a set of multivariate time series. Our proposed algorithm will be able to extract temporal association rules, in which the antecedent of the rule is a multivariate complex pattern.

The algorithm described in the article may deal with both the temporal data mining goals described above, allowing to answer questions of the kind: does pattern A precede pattern B? Does pattern A frequently precede pattern B? Since the problem of defining the precedence relationships is highly context-dependent, our approach is aimed at providing different design parameters which can be set by the data analyst. The result will be a general purpose tool for mining time series databases; such tool will however require a fine tuning on the available problem in order to properly extract useful information.

The article is structured as follows: in Sect. 2, the knowledge-based Temporal Abstraction framework is introduced. We will then propose a new kind of temporal association rule and a method for their extraction from time series (Sect. 3). The method will be further tested (Sect. 4) on simulated data and on two applications related to the biomedical field: time series coming from hemodialysis monitoring and time series describing gene expression in human cell cycle. After a section on related work, the article will end with some concluding remarks.

2 Patterns and knowledge-based Temporal Abstractions

The problem of discovering occurrences of temporal relationships between patterns characterizing a time series needs the accomplishment of three conceptual and procedural steps. First of all, it is necessary to define the patterns and retrieve them in the time series; then a formal definition of the relationships of interest must be given and, finally, an algorithm for searching for frequent occurrences of such relationships in the dataset must be designed, implemented and run. As mentioned in the introduction, to define and describe the time series through complex patterns we exploited the knowledge-based Temporal

Abstractions technique (Shahar 1997; Bellazzi et al. 1998; Combi and Chittaro 1999); such methodology offers a natural way for defining a qualitative representation of temporal data and gives a sound basis for defining patterns and for specifying temporal relationships between them.

Intuitively, a pattern is a behavior or property that we may want to distinguish in the data. In temporal data, a pattern is usually associated to a time interval in which such behavior occurs. Moreover, a pattern is often related to a qualitative representation of the property that we are looking for, which may be interesting in the problem domain. For example, a pattern may be an increasing trend of a variable, or an up and down behavior repeated several times. In order to resort to a precise definition of pattern, able to preserve the qualitative nature of its interpretation, we give a formalization referring to the Temporal Abstractions (TAs) framework (Shahar 1997).

Temporal Abstractions provide a description of a (set of) time series through sequences of temporal intervals corresponding to relevant patterns detected in their time courses. The basic approach of the TAs framework is to move from a time-point to an interval-based representation of time series data. In this work we start from the data model proposed in Bellazzi et al. (1998), where temporal data are time-stamped entities, called *events*, while their abstract representation is given by TAs as a sequence of intervals, called *episodes*. Each episode corresponds to a specific behavior of interest detected in the time course of the data.

The generation of the episodes can be viewed as a Temporal Abstraction task; TA tasks can be divided into two subtasks, each one solved by specific *mechanisms*:

– Basic TAs—solved by mechanisms that abstract time-stamped data into intervals (input data are events and outputs are episodes);
– Complex TAs—solved by mechanisms that abstract intervals into other intervals (input and output data are episodes).

2.1 Formalizing temporal abstractions

Let us assume to have a time series TS: it is an ordered finite set $(\{(v_i, t_i)\}, <)$, where the couple (v_i, t_i) is composed by a value v_i belonging to the domain D of some numerical values,[1] and by a timestamp t_i belonging to the time domain T, isomorphic to the natural numbers. Let us assume that $\mathscr{TS}$ is the universe of all the possible time series $TS : \mathscr{TS} \equiv \{TS\}$. Moreover, we assume to have the domain $\mathscr{I}$ of intervals: $\mathscr{I} = \{[a, b]\}$ with $a \leq b, a$ and b ranking on the natural numbers, where $[a, b] \equiv \{x | a \leq x \leq b\}$.

A TA *abs* is a tuple $< \alpha_{abs}, \beta_{abs} >$, where α_{abs} is an abstraction specification, called *pattern*, and β_{abs} is the application of the pattern to some specific data.

[1] Even though we will consider only numerical values in the article, our framework can suitably handle abstractions on a domain of ordinal values.

As we distinguish basic and complex abstractions, we can recursively define specification and application as in the following.

Abstraction specification α_{abs} for a basic abstraction is a mapping

$$\alpha_{abs} : \mathscr{TS} \rightarrow 2^{\mathscr{I}}$$

while application β_{abs} is a tuple $\langle TS1, E\rangle$, where $TS1 \in \mathscr{TS}, E \subset \mathscr{I}$, and $\alpha_{abs}(TS1) = E$, with $E = \{[e.start, e.end]\}$. Each interval e is called *episode*.

A complex abstraction is composed by a specification of the form

$$\alpha_{abs} = \alpha_{a1}\varphi_{abs}\alpha_{a2}, \varphi_{abs} : 2^{\mathscr{I}} \times 2^{\mathscr{I}} \rightarrow 2^{\mathscr{I}},$$

where α_{a1}, α_{a2} are the patterns of the composing abstractions, and φ_{abs} is the specification of how to relate episodes of the two composing abstractions to build up the episodes of α_{abs}. Application β_{abs} is a tuple $\langle\beta_{a1}, \beta_{a2}, E\rangle$ where β_{a1} and β_{a2} are applications of two abstractions $a1$ and $a2$ (which cannot be defined in terms of abstraction *abs*), $E \subset \mathscr{I}$ is the set of episodes, and $\varphi_{abs}(E_{a1}, E_{a2}) = E$, being E_{a1} and E_{a2} the episode sets of the component abstractions $a1$ and $a2$, respectively.

The only requirement for the set E of episodes we consider is related to the fact that, being episodes an abstract representation of some time series, some kind of *maximality* is imposed: it is not possible to have in the set of episodes neither intervals one containing another nor contiguous intervals. More formally:

$$\forall e \in E(\neg\exists e_1 \in E(e_1 \,\mathtt{d}\, e \vee e \,\mathtt{d}\, e_1 \vee e_1 \,\mathtt{m}\, e \vee e \,\mathtt{m}\, e_1)),$$

where $\mathtt{d}$ and $\mathtt{m}$ stand for the Allen's relations *DURING* and *MEETS*, respectively (Allen 1984).

Different approaches can be taken for specifying patterns, which describe which sets of intervals correspond to different time series: they can be specified in a declarative manner by logical formulas; often they are described by specific mathematical formulas, where ad-hoc parameter values allow one to deal with different data acquired in several domains; moreover, they can be specified in an imperative way, by specifying step by step the algorithm which builds up the set E of episodes, given either a time series or the episode sets of other abstractions. In the following, we introduce the main basic abstractions we considered in this work and then discuss the complex abstractions we built up.

2.2 Basic Temporal Abstractions

Herein we distinguish two different types of basic TAs: *State* and *Trend*. *State* TAs extract the intervals in which values are within a predefined range. More formally, given a variable V (possibly) assuming continuous values, we can always perform a *qualitative description* on V. The qualitative description corresponds to a suitable discretization of the values $D(V) = \{[V_1, V_2), [V_2, V_3), \ldots, [V_{m-1}, V_m], [V_m, V_{m+1}]\}$ and to the association of a set of labels, or *states*, $V_L = \{V_{L1}, V_{L2}, \ldots, V_{Lm}\}$ to such discretization. A State TA pattern for a (state) abstraction V_{Li} is thus a specification of the form:

$$\begin{aligned}\forall TS \in \mathscr{TS}, \alpha_{VLi}(TS) = \; & \{[e.start, e.end] | \exists(v1, t1), (v2, t2) \in TS((\exists(v3, t3) \\ & \in TS((v3, t3) = \text{prev}((v1, t1))) \rightarrow v3 \notin [V_i, V_{i+1}]) \\ & \wedge (\exists(v4, t4) \in TS((v4, t4) = \text{next}((v2, t2))) \rightarrow v4 \\ & \notin [V_i, V_{i+1}]) \wedge \forall(v, t) \in TS(t \in e \rightarrow v \in [V_i, V_{i+1}]) \\ & \wedge e.start = t1 \wedge e.end = t2)\}\end{aligned}$$

where the functions $prev(\cdot)$ and $next(\cdot)$ defined for a time series TS return the previous element and the next one of a given element of the set TS, according to the given (temporal) order, respectively.

Given a time series TS_V of the variable V, i.e., a sequence of n values $v_1, v_2, \ldots, v_n$, holding at the time points $t_1, t_2, \ldots, t_n$, respectively, a *State* TA application looks for the time intervals in which V has values corresponding to only one given state V_{Li} in V_L. According to the previous formalization, the *State* TA application is thus a tuple $\langle TS_V, E_V \rangle$. State abstractions correspond to expressions like "*high* arterial pressure" or "*low* temperature."

Trend TAs represent increase, decrease, and stationary patterns in a numerical time series. As an example, a simple declarative definition of an I (*Increasing*) pattern could be:

$$\begin{aligned}\forall TS \in \mathscr{TS}, \alpha_I(TS) = \; & \{[e.start, e.end] | \exists(v1, t1), (v2, t2) \in TS((\exists(v3, t3) \\ & \in TS((v3, t3) = \text{prev}((v1, t1))) \rightarrow v3 > v1) \wedge (\exists(v4, t4) \\ & \in TS((v4, t4) = \text{next}((v2, t2))) \rightarrow v4 < v2) \wedge \forall(v, t)(v^*, t^*) \\ & \in TS((t1 \leq t \leq t^* \leq t2) \rightarrow v \leq v^*) \wedge e.start = t1 \wedge e.end = t2)\}\end{aligned}$$

where, for each time series, we associate maximal sets of values increasing over time with an episode ranging from the minimum to the maximum of the timestamps related to the considered values.

Of course, different trend patterns could be suitably defined, exploiting different strategies. In particular, they may be adapted to deal with noisy data, performing a sophisticated search for obtaining the largest possible intervals.

In our approach trend patterns are obtained in a procedural way: any given time series is processed in order to obtain a description of the variables in terms of a set of consecutive basic *trend* TAs.

A straightforward way of deriving trend abstractions on time series is to rely on a piecewise linear representation of raw temporal data. An overview of the most used segmentation algorithms proposed in the literature was offered by Keogh et al. (2004) and recently reviewed and updated by Mörchen (Mörchen 2006). The algorithms were classified into three methodological groups:

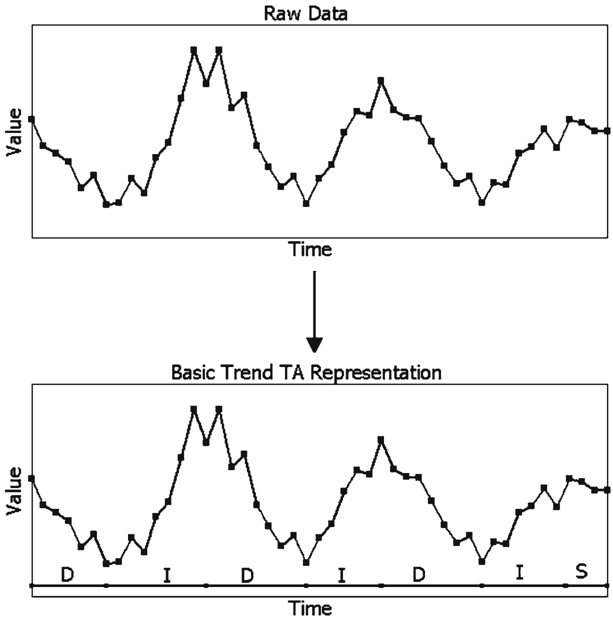

Fig. 1 Representation through trend TAs. A raw time series is processed in order to derive the sets of episodes of validity of trend abstractions in the set $\{\alpha_I, \alpha_D, \alpha_S\}$, where *I* stands for *Increasing, D* for *Decreasing*, and *S* for *Stationary*

– sliding window algorithms, that recursively expand segments by adding new points until some error bound is exceeded;
– top-down algorithms, that recursively partition the time series until some stopping constraint is satisfied;
– bottom-up algorithms, that, starting from the finest possible approximation, iteratively merge segments until some stopping condition is satisfied.

In Keogh et al. (2004), each methodology is evaluated on ten datasets with diverse characteristics and their performances are compared in order to highlight advantages and disadvantages of each technique; moreover, a new algorithm which is a mixture of the sliding window and the bottom-up approaches is introduced.

For what concerns the construction of the final trend TA representation, there are several possible algorithms that can be used for the extraction of the piecewise linear approximation; the choice is up to the user, and may depend on several characteristics of the dataset as the length of the time series, the expected level of noise, etc.

The results discussed in Sect. 4 are all obtained by processing the time series with a sliding window algorithm (Bellazzi et al. 2003). Figure 1 shows an example of a time series represented through trend abstractions; in particular, the episodes related to the specification set $\{\alpha_I, \alpha_D, \alpha_S\}$ (where *I* stands for *Increasing, D* for *Decreasing, S* for *Stationary*) are detected and the episodes identified by the corresponding labels are shown.

It is apparent from the description above that each basic TA requires the definition of several design parameters to completely specify the kind of pattern to be detected in dependence of the characteristics of the application. Usually, there is a set of parameters which are common to both State and Trend TAs, while some others are specific to the particular TA. The common parameters are: the *granularity*, which defines the maximum temporal distance allowed to aggregate two measurements in the same episode and the *minimal extent*, that specifies the minimal time-span of an episode to be considered relevant in the specific context. As an example of a TA specific parameter we can mention the *min-slope* parameter of Trend TAs which defines the minimum increase/decrease rate that triggers the detection of a pattern.

2.3 Complex Temporal Abstractions

Complex TAs are used to detect patterns characterized by behaviors which can't be represented by basic TAs. Complex TA applications need as input two episode sets, associated to two different TAs, and provide as output an episodes set, associated to the new TA. The episode set is evaluated according to a pattern specified between episodes of the two composing episode sets. The complex TA patterns are based on temporal relationships: more specifically, the temporal relationships investigated correspond to the 13 temporal operators defined in Allen's algebra (Allen, 1984). They include: *BEFORE, FINISHES, OVERLAPS, MEETS, STARTS, DURING*, their corresponding inverse relations, and the *EQUALS* operator. We will refer to the set of the 13 Allen's operators as *AO*. As a consequence, we define a complex TA mechanism for each temporal relationship.

Given two episode sets, $E_1 = \{[e_{11}.start, e_{11}.end], [e_{12}.start, e_{12}.end], \ldots, [e_{1h}.start, e_{1h}.end]\}$ and $E_2 = \{[e_{21}.start, e_{21}.end], [e_{22}.start, e_{22}.end], \ldots, [e_{2k}.start, e_{2k}.end]\}$, a complex TA application associates episodes of E_1 and E_2 according to the specified Allen's temporal operator, obtaining an ordered set of episodes in which the corresponding relationship holds. Time intervals over which complex TAs (i.e., complex episodes) hold are suitably defined with respect to the starting and ending points of the involved episodes of E_1 and E_2, considering also the adopted operator. More formally, such a pattern could be defined as:

$$\forall E1, E2 \in 2^{\mathrm{I}} \varphi_{abs}(E1, E2) = \{e \mid \exists\, e1 \in E1, \exists\, e2 \in E2(e1\,\theta\, e2 \wedge e = e1 \zeta\, e2)\},$$

where $\theta \in AO$ and ζ is any interval constructor (as intersection, union, or user-defined ones).

Each pattern for the 13 complex TAs requires the specification of one or more parameters used to make more or less restrictive the temporal relationship and, therefore, to make more flexible the definition of the pattern to be recognized. For example, the *FINISHES* operator could require as parameter the maximum temporal distance between the pair of interval starting points, while the

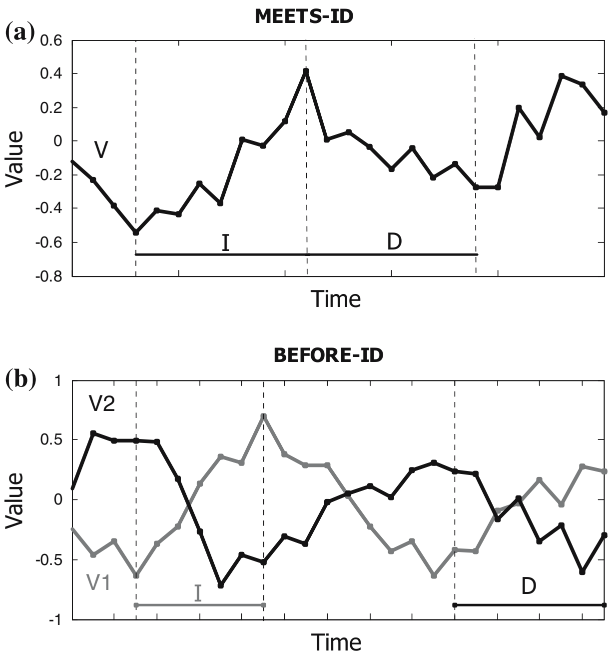

Fig. 2 Complex TAs used to detect patterns of complex shape both (**a**) on a single time series (*MEETS–ID*), and (**b**) on multidimensional time series (*BEFORE–ID*); in this case, an *I* episode in *V*1 occurs before a *D* episode in *V*2

BEFORE operator (possibly) requires the maximum time gap between the end point of the first interval and the starting point of the second one.

In addition, as the two series of intervals used as input to a complex abstraction can originate both from the same and from different time series, we can exploit this kind of TA to detect a great variety of patterns. Figure 2 shows patterns of complex shape which have been detected both on a single time series (Fig. 2a), and on multiple time series (Fig. 2b).

For example, considering the complex abstraction *BEFORE–ID*, depicted in Fig. 2b, we could formally define the complex pattern as:

$$\alpha_{BEFORE-ID} = \alpha_I \varphi_{BEFORE-ID} \alpha_D$$

$$\forall E1, E2 \in 2^{\mathrm{I}} \varphi_{BEFORE-ID}(E1, E2) = \begin{array}{l} \{e | \exists\, e1 \in E1, \exists\, e2 \in E2 \\ (e1\, \mathtt{b}\, e2 \wedge e = e1 \cup_{\mathrm{c}} e2)\}, \end{array}$$

where the operator $\cup_{\mathrm{c}}$ builds the minimal interval containing the two interval operands (in this case, the interval spans from the start of $e1$ to the end of $e2$) and $\mathtt{b}$ stands for the Allen's relation *BEFORE*. We recall here that given two episodes, $e_1 \equiv [e_1.start, e_1.end]$ and $e_2 \equiv [e_2.start, e_2.end], e_1 \mathtt{b}\, e_2 \Leftrightarrow e_1.end < e_2.start$.

Application $\beta_{BEFORE-ID}$ of this temporal abstraction is formally defined by the tuple $\langle\langle TS1, E_I\rangle, \langle TS2, E_D\rangle, E_{BEFORE-ID}\rangle$.

In the next section we will focus on how we may represent and mine time series within the TA framework relying on trend abstractions only.

3 Mining Temporal Abstractions

In our framework, the data mining task is performed through different steps. First, the user defines a set of complex patterns of interest that constitute the basis for the construction of temporal rules; such complex patterns are represented and retrieved in the data through the formalism of knowledge-based Temporal Abstractions. Then, temporal rules may be specified by setting some context-dependent parameters. Finally, after confidence and support for temporal rules on complex patterns have been specified, an Apriori-like algorithm looks for meaningful temporal relationships among the complex patterns of interest.

3.1 Complex TA representation

The description of the time series through basic *trend* TAs represents the starting point for the creation of the final representation of the complex TAs of interest. The core aspect of this phase is the definition of a set AoI of abstractions of interest. Since we have based the initial representation of the time series on trend TAs, each complex pattern α_{abs} for $abs \in AoI$ is defined in terms of *Increasing, Decreasing, Stationary*,[2] through the abstraction specifications α_I, α_D, and α_S, respectively. The set AoI may be either user-defined or automatically suggested to the researcher after a pre-processing of the initial qualitative representation of the time series in order to extract the most significant behaviors (Chiu et al. 2003; Sacchi et al. 2005a). As an example, let us consider a situation in which it is interesting to investigate whether a peak in the dynamics of a variable V_1 is often temporally related to an opposite peak of another variable V_2. This problem can be formalized by defining the set AoI as in the following:

$$AoI = \{\langle\alpha_{ID1}, \beta_{ID1}\rangle, \langle\alpha_{DI2}, \beta_{DI2}\rangle\} \text{ where}$$
$$\alpha_{ID1} = \alpha_I \varphi_{ID1} \alpha_D \text{ and}$$
$$\alpha_{DI2} = \alpha_D \varphi_{DI2} \alpha_I$$
$$\forall E1, E2 \in 2^{\mathrm{I}} \varphi_{ID1}(E1, E2) = \{e | \exists e1 \in E1, \exists e2 \in E2 (e1\, m\, e2 \wedge e = e1 \cup_c e2)\},$$
$$\text{and } \varphi_{DI2} \equiv \varphi_{ID1}.$$

[2] In general, the elements of AoI may be described also by other kind of abstractions (state TAs, or both state and trend), depending on the qualitative representation chosen; the algorithm that we are presenting is in fact flexible and can be easily adapted to different qualitative descriptions of the data.

As for the applications,

$$\beta_{ID1} = \langle\langle TS1, E_{IV1}\rangle, \langle TS1, E_{DV1}\rangle, E_{IDV1}\rangle$$
$$\beta_{DI2} = \langle\langle TS2, E_{DV2}\rangle, \langle TS2, E_{IV2}\rangle, E_{DIV2}\rangle$$

The extraction of significant temporal relationships will thus be performed on those episodes that verify α_{ID1} over the increasing and decreasing of V_1 and α_{DI2} over the decreasing and increasing of V_2, respectively.

As we did for the set AoI of the previous example, hereinafter we will focus on the presence of complex patterns by applying the Allen's relation m (*MEETS*). In other words, each φ_{abs} we will consider for complex abstractions compares intervals of the composing abstractions through the *MEETS* relation: the name we will use for the complex abstraction will be built by simply concatenating the names of the composing abstractions, as in the previous example.

We recall here that given two episodes, $e_1 \equiv [e_1.start, e_1.end]$ and $e_2 \equiv [e_2.start, e_2.end]$, $e_1 \mathtt{m}\, e_2 \Leftrightarrow e_1.end = e_2.start$.

An example of the steps that lead to the representation through *complex temporal abstractions* for a single time series is depicted in Fig. 3.

3.2 Definition and evaluation of Temporal Association rules

In order to systematically look for temporal relationships between the complex temporal patterns introduced in Sect. 3.1, we first need a formal definition of the notion of precedence we want to represent. To this aim, we consider temporal relationships expressed by the temporal operator *PRECEDES*, defined as follows (Bellazzi et al. 2005):

Definition 1 Given two episodes, $e_1 \equiv [e_1.start, e_1.end]$ and $e_2 \equiv [e_2.start, e_2.end]$, $e_1 PRECEDES e_2 \Leftrightarrow e_1.start \leq e_2.start \wedge e_1.end \leq e_2.end$.

According to this definition, the operator *PRECEDES* synthesizes 6 of the 13 Allen's temporal relationships, which are: *OVERLAPS, FINISHED-BY, MEETS, BEFORE, EQUALS*, and *STARTS*.

Based on the *PRECEDES* relation, the temporal association rule is defined as follows:

Definition 2 A *temporal association rule* is an implication of the form $A \rightarrow_P c$, where P is a triple (*LS, G, RS*), A is the set $\{a1, a2, a3, \ldots, an\} \subseteq AoI$ and $c \in AoI$.

The temporal association rule is evaluated on the set $\{E_{a1}, E_{a2}, E_{a3}, \ldots, E_{an}\}$ of episode sets corresponding to the abstractions $a1, a2, a3, \ldots, an$ and is satisfied when the following formula holds:

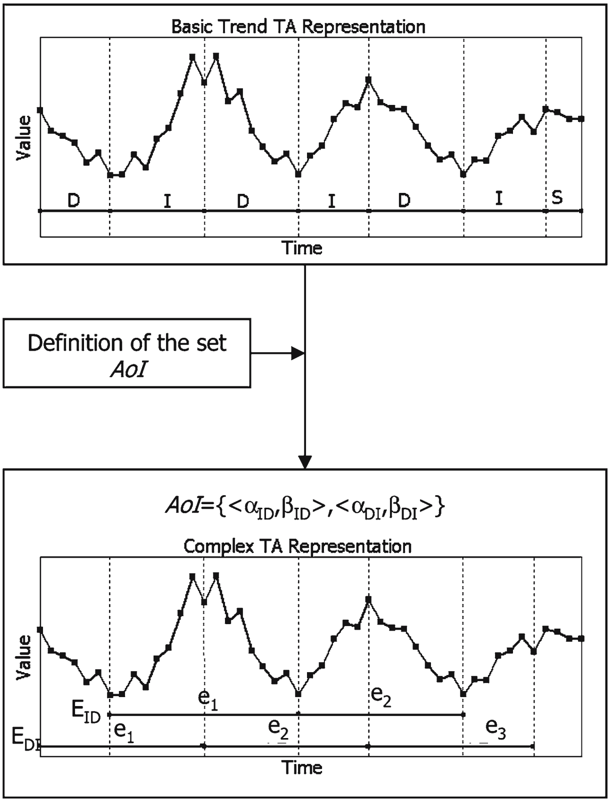

Fig. 3 Example of the representation through complex temporal patterns for one time series. After the definition of the set $AoI = \{< \alpha_{ID}, \beta_{ID} >, < \alpha_{DI}, \beta_{DI} >\}$ the set of episodes of validity of each abstraction (E_{ID}, E_{DI}) is determined

$$\begin{aligned}&\exists e^{a1},\ e^{a2}, \ldots, e^{an}, e^{c} \left(\cap_{i=1..n} e^{ai} \neq \emptyset \wedge\right.\\&\quad \left[max_{i=1..n}(e^{ai}.start), min_{i=1..n}(e^{ai}.end)\right] PRECEDES e^{c} \wedge\\&\quad (e^{c}.start - max_{i=1..n}(e^{ai}.start)) \leq LS \wedge\\&\quad (e^{c}.end - min_{i=1..n}(e^{ai}.end)) \leq RS \wedge |e^{c}.start - min_{i=1..n}(e^{ai}.end)| \leq G)\end{aligned}$$

where $e^{ai}(i = 1, \ldots, n)$ is ranging over episodes of the set E_{ai} for the abstraction ai.

Informally, a temporal association rule holds when all the patterns in the antecedent intersect and when the relation *PRECEDES* holds between the intervals of this intersection and an episode of the pattern in the consequent. As we observe from the above definition, the *PRECEDES* relationship may be conveniently constrained in the temporal association rule by the parameters set in the triple P, introduced to allow some restrictions on the mutual position of the involved intervals (see Fig. 4). These parameters are: the *left shift* (LS), defined as the maximum allowed distance between $e^{c}.start$ and $max_{i=1..n}(e^{ai}.start)$,

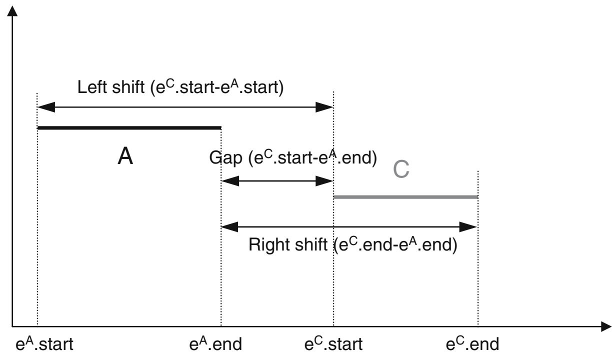

Fig. 4 Parameterization of the *PRECEDES* relationship through the triple (LS, G, RS) which fixes some restrictions to the mutual positions of the intervals

Table 1 Parameterization of the temporal association rule: specific constraints and allowed parameters

Temporal Operator	Specific constraints		
	LS	*G*	*RS*
BEFORE	–	>0	–
MEETS	>0	=0	>0
OVERLAPS	>0	<0	>0
FINISHED BY	>0	X	=0
EQUALS	=0	X	=0
STARTS	=0	X	>0

Herein the symbol – denotes that the corresponding parameter may assume any value, while the symbol X denotes that the parameter is not defined for the corresponding temporal operator

the *gap* (G), defined as the maximum allowed distance between $e^c.start$ and $min_{i=1..n}(e^{ai}.end)$ and the *right shift* (RS), defined as the maximum allowed distance between $e^c.end$ and $min_{i=1..n}(e^{ai}.end)$. Note that only the difference between $e^c.start$ and $min_{i=1..n}(e^{ai}.end)$ may assume negative values without violating the precedence constraints defined when introducing the *PRECEDES* operator.

As shown in Table 1, it is possible to select the subset of relationships to be evaluated during the analysis by properly tuning the three parameters G, RS, and *LS*; it is not indeed necessary to look for all the relationships covered by *PRECEDES* in every kind of analysis.

Another important feature of this parameterization is the possibility of avoiding (possible) ambiguous situations as the one depicted in Fig. 5. This picture shows two episodes $x, z \in E_{a1}$ which satisfy pattern α_{a1}, and the interval $y \in E_{a2}$ which satisfies α_{a2}. If not parameterized, we would obtain both $\{a1\} \rightarrow_P a2$ and $\{a2\} \rightarrow_P a1$, thus generating a possible ambiguity: indeed, it could be the case that the second rule is not meaningful in the considered domain, due to the

Fig. 5 Effects of the parameterization of the temporal association rule. If no parameter is set, both the rules $\{a1\} \rightarrow_{\mathrm{P}} a2$ and $\{a2\} \rightarrow_{\mathrm{P}} a1$ would be verified in the example. In fact both the pairs of intervals ($[x.start, x.end]$, $[y.start, y.end]$) and ($[y.start, y.end]$, $[z.start, z.end]$) satisfy the precedence constraints stated in the definition of *PRECEDES*. On the other hand, if suitable values for the parameters are fixed, it would be possible to extract only the relationship which connects $[x.start, x.end]$ with $[y.start, y.end]$

long temporal distance between y and z. If we properly set the parameters, we could be able to find only the two closest intervals; this would be very important especially when dealing with long time series and with large datasets.

Once the temporal association rule has been formally defined, the next step is to efficiently search for occurrences of the corresponding relationships between patterns in the data. This can be done both over multivariate datasets, where time series coming from the measurement of different variables are collected, but also on single time series, to study the timing of different patterns that may repeat frequently over time. To pursue this goal, what we need is to introduce a strategy for the search of frequent precedence relationships between complex patterns in the time series. It is therefore necessary to couple the complex TA representation and the temporal association rule with a search strategy for an efficient mining of temporal rules (Bellazzi et al. 2005; Höppner and Klawonn 2002a; Winarko and Roddick 2005; Kam and Fu 2000). We herein propose a method for temporal rule extraction based on an Apriori-like strategy, which looks for both the antecedent and the consequent of the rule coming from the episode sets of complex TAs that represent the time series. With respect to traditional algorithms for temporal rule extraction, the novel feature that is herein introduced is the possibility of creating arbitrarily complex patterns both in the antecedent and in the consequent thanks to the introduction of the set *AoI* of complex abstractions. The rule extraction strategy will then look for rules in which a set of intersecting TAs episodes (the antecedent) has a precedence temporal relationship with another TA episode (the consequent). Note that, since temporal rules are derived through the combination of complex temporal abstractions on the basis of a temporal relationship, they can themselves be considered as complex TAs.

3.3 Frequent complex patterns: confidence and support

In this section, we will briefly introduce the notions of confidence and support extended to the temporal domain. These concepts are of crucial importance for the definition of frequent patterns and, as in the Apriori algorithm (Agrawal and Srikant 1994), they are essential for an efficient search over the rule space. In more detail, in order to define confidence and support, we will first introduce some terminology. We will denote:

– *TSO*: the time span, i.e., the total duration, of the observation period over which the rule is derived;
– *RTS*: the rule time span, i.e., the time span corresponding to the union of the episodes in which both the antecedent and the consequent of the rule occur;
– *NAT*: the number of times (episodes) the antecedent occurs during *TSO*;
– *NARTS*: the number of times (episodes) the antecedent occurs during *RTS*.

We therefore define:

– *Support*[3] (Sup) = *RTS* / *TSO*;
– *Confidence* (Conf) = *NARTS* / *NAT*.

Intuitively, the support gives a measure of how the rule is ‘spread’ over the observation time span, while the confidence indicates the frequency of the rule over the total amount of episodes of the antecedent. It is important to note that the two quantities must always be evaluated together, since they both give important information about the quality of a rule. When dealing with long time series, it may in fact happen that a precedence relationship occurs several times over very short intervals; in this situation the support would be low, but the confidence may be very high.

Figure 6 shows a simple example of these definitions. In particular, we are interested into a precedence relationship between abstractions a and c in a multivariate set of time series. The picture shows that pattern α_a is verified over the two time intervals x and z, while pattern α_c is verified over y; the temporal association rule is instead satisfied only on two of the three intervals, x and y, since $x.start < y.start$ and $x.end < y.end$. The picture shows the time intervals defined as *TSO* and *RTS*, considering *RTS* as the time span covered by the intervals corresponding to the antecedent (x) and the consequent (y) of the rule. In this example the two quantities *NAT* and *NART* would be equal respectively to 2 (x and z) and 1 (only x is involved in the rule), so the confidence will be: *Conf* = 0.5.

3.4 Temporal rules extraction

Once we have stated the definitions for confidence and support, we can finally introduce the temporal rule extraction algorithm. As already mentioned, the method follows an Apriori-like search strategy where frequent patterns are selected on the basis of thresholds for confidence and support (*min_conf, min_sup*). As it will be clear in Sect. 4, also when dealing with the definition of these thresholds, the features of the data and the domain knowledge play a key role for the proper selection of significant rules. The pseudocode in Figure 7 illustrates the development of the rule extraction method.

[3] Several definitions of support can be considered; in our case, we chose to consider the real time span of the episodes, in order to take into account also low frequency episodes with long *TSO*.

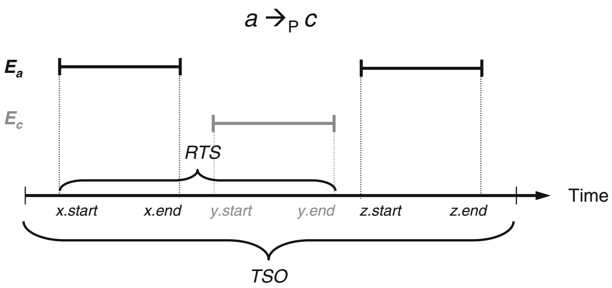

Fig. 6 Definition of confidence and support. We are interested into a precedence relationship between abstraction *a* and abstraction *c* over the two episode sets E_a and E_c. The temporal association rule is satisfied only on two of the three intervals, *x* and *y*. The picture shows the time intervals defined as *TSO* and *RTS*. In this example the two quantities *NAT* and *NARTS* are equal respectively to 2 (*x* and *z*) and 1 (only *x* is involved in the rule)

```
INPUT: the parameter set P, to evaluate the relationship PRECEDES; the set AoI of abstractions
of interest; the threshold values min_conf and min_sup, for Confidence and Support,
respectively.
OUTPUT: Result, the set containing all the frequent temporal association rules, extracted from
AoI

set Rul = Ø
for each a_i ∈ AoI do
   // Fix the Consequent
   set cons = a_i;
   // Creation of the (simple, i.e. with a single pattern as
   // antecedent) rules:
   set k = 1
   set A_k = Ø
   for each a_j ∈ AoI -{cons} do
       // Consider the PRECEDES relationship between
       // a_j ∈ AoI -{cons} and cons.
       if ((Sup({a_j}→_P cons) ≥ min_sup over the sets of episodes related to abstractions in AoI)
           set A_k = A_k ∪ {{a_j}}
           set Rul = Rul ∪ {({a_j} →_P cons)}
       endif
   endfor
   // Creation of the set of (complex) rules:
   Repeat:
       set k=k+1;
       set A_k = Ø;
       for each c_j∈A_{k-1} do
           for each {a_1}∈A_1 do
               if ((Sup(c_j∪{a_1} →_P cons) ≥ min_sup over the sets of episodes related to
                         abstractions in AoI)
                         set A_k = A_k ∪ {c_j∪{a_1}}
                         set Rul = Rul ∪ {(c_j∪{a_1} →_P cons)}
           endif
       endfor
   Until: A_k is empty
endfor
set Result  = {(A →_P C) | (A →_P C) ∈ Rul ∧ Conf(A →_P C) ≥ min_conf over the sets of episodes
                         related to abstractions in AoI}
output(Result)
```

Fig. 7 The pseudocode of the proposed algorithm for extracting temporal association rules

Note that, if we are only interested into the discovery of precedence relationships between the abstractions identified by the set *AoI*, for example when dealing with single time series, it is enough to extract the *basic set* (set of rules with antecedent of cardinality 1) and evaluate the results in terms of confidence and support. In the case of a multivariate dataset the method helps to efficiently extract occurrences of frequent precedence temporal rules between patterns that could be arbitrarily complex in the antecedent.

4 Results

The experimental part of this work was performed both to confirm the soundness of the proposed approach and to show the wide applicability of our framework. First, we considered simulated data consisting of univariate and multivariate time series. These simulation studies are directed to a double aim: first, the capability of the algorithm of correctly reconstructing temporal rules is evaluated over two datasets where the patterns of interest and the level of noise corrupting the data is known in advance. Second, both the univariate and the multivariate problem are illustrated to show the potential generality of the method on different applications.

We, then, considered two different kinds of experimental data: the first experimental setting is related to temporal mining of clinical data (blood pressures, heart rate, weight loss, etc.) acquired during the monitoring of hemodialysis sessions: in this case we have to deal with long and noisy time series of clinical variables. The second experimental setting consists of temporally mining genetic regulatory relationships from DNA microarray gene expression data. In this case, data consist of a huge amount of (relatively) short time series related to several genes.

4.1 Studies on simulated data

In this section we present the evaluation of the algorithm on two simulated datasets, where both the univariate and the multivariate problems were considered.

For both the simulation studies, prototypical simple patterns (i.e., trends in the set $\{\alpha_I, \alpha_D, \alpha_S\}$) were simulated using segments of fixed slope and specific duration; these segments, properly composed to form piecewise linear curves, were then used to generate the desired complex patterns.

In order to simulate realistic data, the patterns were added to a set of time series of 2,050 points extracted from the random walk process x_t specified by Eq. (1).

$$x_{t+1} = x_t + v_t, \qquad \text{where } v_t \sim N(0, (0.01)^2) \tag{1}$$

The final simulated data were thus obtained according to the schema introduced in Table 2.

The first dataset simulates a problem with a single time series where an up-and-down peak is frequently followed by an increase episode. To represent this behavior, the patterns α_{ID} and α_I were added to the random walk time series so that an *ID* episode is often followed by an *I* one. To simulate a realistic situation where 'false' precedence may occur (presence of the antecedent but not of the

Table 2 Characteristics of the two simulated datasets

Dataset	N^{O} of time series	AoI	Qualitative patterns added to data	Noise	Target rule
Simulated 1	1	$\{ID, I\}$	ID followed by I	ID	$\{ID\} \to_{\mathrm{P}} I$
Simulated 2	3	$\{ID_1, DI_2, ID_3\}$	TS_1: ID TS_2: DI TS_3: ID	–	$\{ID_1, DI_2\} \to_{\mathrm{P}} ID_3$

For each simulated dataset the Table reports the number of time series involved, the set of Abstractions of Interest (*AoI*) that has been generated, how the patterns were added to the random-walk time series and the target rule, that is the rule which we expect to find in the data (I for Increasing, ID for episodes of *Increasing* meeting episodes of *Decreasing*, DI for episodes of *Decreasing* meeting episodes of *Increasing*; for the second dataset TS_i is the i-th time series)

consequent) we also added some single ID episodes. In this case we evaluated the ability of the algorithm in reconstructing the relationship $\{ID\} \to_{\mathrm{P}} I$. A simulated time series for dataset 1 is shown in Fig. 8a.

The second study is aimed at simulating a multivariate problem with three time series, characterized by the fact that a contemporaneous (but opposite) peak in the first two time series (TS_1 and TS_2) precedes an up-and-down peak in the third one (TS_3). The patterns $\alpha_{ID1}, \alpha_{DI2}, \alpha_{ID3}$ were thus considered and suitably simulated on the initial time series (see Table 2); the capability of the algorithm in reconstructing the rule $\{ID_1, DI_2\} \to_{\mathrm{P}} ID_3$, considering time series TS_1 and TS_2 for the antecedent, and TS_3 for the consequent, respectively, was tested in terms of confidence and support. The simulated time series for dataset 2 are shown in Fig. 8b.

Tables 3 and 4 show the results obtained by running the rule extraction algorithm over the two simulated datasets. In the first case only the basic set (set of rules with only one element in the antecedent) was generated and confidence and support directly evaluated on it. The goal of the study is to evaluate whether the algorithm is able to find significant precedence relationships between the patterns present in the data; this property could be an important feature in the study of the timing of different patterns into a single time series. In the second study, instead, we tested the algorithm to check its capability of finding frequent rules with complex antecedents characterized by pair-wise intersecting patterns coming from different variables.

We achieved satisfactory results in both cases: for the univariate problem all the occurrences of the precedence relationships $\{ID\} \to_{\mathrm{P}} I$ were discovered (Table 3). Moreover, the information on the confidence assures that the algorithm had been able to extract all the occurrences of the pattern ID, but that some of them were not involved into the precedence relationships of interest.

For what concerns the multivariate problem (Table 4) the algorithm was able to derive all the precedence relationships between simple patterns (see Fig. 8b) and also a complex rule characterized by the combination of two intersecting patterns in the antecedent. In particular, this rule expresses the precedence

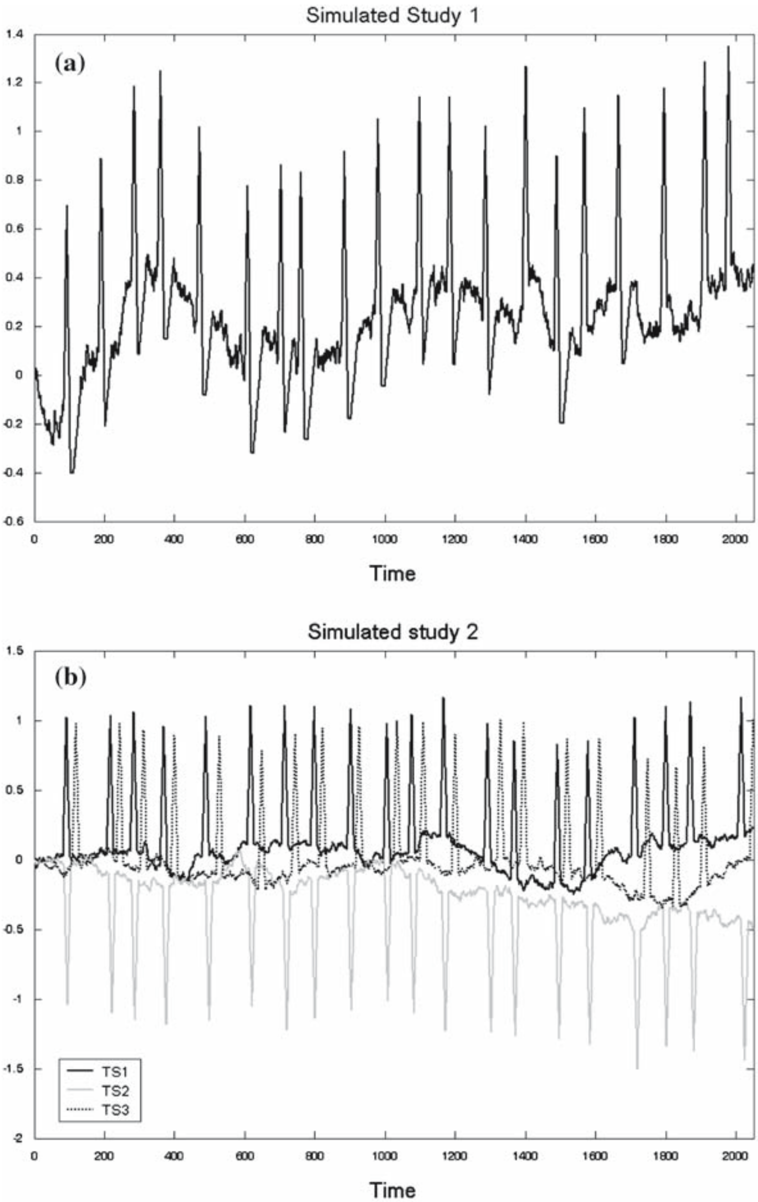

Fig. 8 Examples of the simulated datasets. (**a**) In the first dataset (one variable) the time series is characterized by patterns belonging to the set $AoI = \{<\alpha_I, \beta_I>, <\alpha_{ID}, \beta_{ID}>\}$. Both *ID* episodes followed by *I* trends and *ID* episodes alone can be found. (**b**) The second dataset represents a multivariate problem with 3 time series: *TS*1, *TS*2, and *TS*3. These time series are characterized by occurrences of the patterns belonging to the set $AoI = \{<\alpha_{ID1}, \beta_{ID1}>, <\alpha_{DI2}, \beta_{DI2}>, <\alpha_{ID3}, \beta_{ID3}>\}$ In particular, *TS*1 and *TS*3 present *ID* episodes, while *TS2* is characterized by patterns of an opposite shape *DI*

Table 3 Results for the first simulated dataset (*I* for Increasing, *ID* for episodes of *Increasing* meeting episodes of *Decreasing*)

Operator: *PRECEDES*				
Parameters: $min\_conf = 0.7$, $min\_sup = 0.2$, $LS = 30, G = 20, RS = 30$				
Rule		N° of rule episodes	Confidence	Support
Antecedent	Consequent			
ID	*I*	15	0.71429	0.34195
I	*ID*	0	–	–

Table 4 Results for the second simulated dataset (*ID* for episodes of *Increasing* meeting episodes of *Decreasing, DI* for episodes of *Decreasing* meeting episodes of *Increasing*; TS_i is the *i*th time series)

Operator: *PRECEDES*						
Parameters: $min\_conf = 0.7$, $min\_sup = 0.2$, $LS = 40$, $G = 45$, $RS = 40$						
Rule				N° of rule episodes	Confidence	Support
Antecedent		Consequent				
Time series	Pattern	Time series	Pattern			
TS_1	*ID*	TS_2	*DI*	20	1	0.27805
TS_1	*ID*	TS_3	*ID*	19	0.95	0.5
TS_2	*DI*	TS_3	*ID*	19	0.95	0.45122
TS_1 TS_2	*ID* *DI*	TS_3	*ID*	19	0.95	0.45122

relationship that occurs between the two intersecting patterns ID_1 and DI_2 in TS_1 and TS_2 and the pattern ID_3 in the third time series, TS_3.

The parameters (first row in Tables 3 and 4) were selected to prevent the algorithm from detecting rules that may lead to an ambiguous or contradictory interpretation of the results (e.g., in the first example the precedence rule $\{I\} \rightarrow_{\mathrm{P}} ID$ is excluded from the set of extracted relationships). Of course, the assignment of the parameters specifying the *PRECEDES* operator constraints (LS, G, RS) and the thresholds for confidence and support (*min_conf, min_sup*) is a procedure that should be strongly driven by the domain knowledge about the problem. A complete control on the problem domain should allow to extract the kind of rules one is interested into. In order to better highlight the influence of the choice of these parameters on the performances of the algorithm, a robustness analysis has been carried out by properly varying the values of LS, G, and RS for the time series in the first simulation study. Let us first note that the choice of very high values for one of the parameters makes the corresponding constraint never violated. According to this observation, we evaluated

Table 5 Robustness analysis on the parameters values (I for Increasing, ID for episodes of *Increasing* meeting episodes of *Decreasing*)

Target rule $\{ID\} \rightarrow_{\mathrm{P}} I$				
Parameters setting	# Episodes of the target rule	# Episodes of $\{I\} \rightarrow_{\mathrm{P}} ID$	Precision	Recall
$LS = 30$ $G = 4000$ $RS = 30$	15	0	1	1
$LS = 55$ $G = 4000$ $RS = 60$	16	30	0.3261	1
$LS = RS = 4000$ $G = 20$	16	–	0.9375	1
$LS = RS = 4000$ $G = 40$	20	30	0.3	1

the performance of the algorithm following two steps: first, the values for LS and RS were changed by letting G assume values greater than the length of the time series. Second, the values for G were changed by letting LS and RS assume values greater than the length of the time series.

Considering as a target rule $\{ID\} \rightarrow_{\mathrm{P}} I$, we computed the following indexes:

$$\text{Precision} = TP/(TP + FP);$$
$$\text{Recall} = TP/(TP + FN),$$

where:

– TP are the true positives, i.e., the occurrences of the target rule correctly extracted by the algorithm;
– FP are the false positives, i.e., the occurrences of the non target rules extracted by the algorithm;
– FN are the false negatives, i.e., the episodes of the target rule not found by the algorithm.

In Table 5, we report the results of the analysis with different values for the three parameters. As the value of the Recall is always 1 (i.e., the algorithm is able to find all the target precedence relationships), the main interesting parameter turns out to be the Precision. In particular it is possible to note that, as the values of the parameters increase to non suitable ranges, the computed Precision gets worst. This is due to the fact that the algorithm both starts to find episodes belonging to the non-target rule and also ($G = 40$) finds non consistent episodes for the target rule.

The evaluation confirms the crucial role of background knowledge in the precedence rule discovery process.

4.2 Analysis of experimental data

The next two sections present the results obtained by applying the method to two different problems, the first in a clinical domain, while the other one concerning the biological problem of inferring gene regulatory relationships from DNA microarray data. These examples allow one to understand the wide spectrum of applicability of the proposed solutions.

4.2.1 Analysis of time series coming from hemodialysis sessions monitoring

In this first application we examine the problem of finding interesting rules occurring between patterns found over a set of variables monitored during an Hemodialysis session. The data had been made available by courtesy of the Dialysis Unit of the A.O. of Vigevano, Italy. The dataset includes 36 patients, each one undergoing several hemodialysis treatments, during which many physiological variables are monitored in order to control the patient's conditions.

Our study is focused on three variables: systolic pressure (*SP*), diastolic pressure (*DP*) and heart rate (*HR*); the measurements are taken by a digital sphygmomanometer. From a clinical viewpoint, it is interesting to look for temporal relationships that may highlight a negative correlation between arterial pressure and heart frequency; such relationships may in fact be related to hypertension or hypotension episodes taking place during a single hemodialysis treatment. Relying on this prior assumption, the set of abstractions was defined as:

$$AoI = \{ISD_{SP}, ISD_{DP}, ISD_{HR}, DSI_{SP}, DSI_{DP}, DSI_{HR}\}$$

ISD and *DSI* abstractions are defined by the *MEETS* operator. For example, the generic specification of an *ISD* abstraction is defined as:

$\alpha_{ISD} = \alpha_I \, \varphi_{ISD} \, \alpha_{SD}$ and
$\forall E1, E2 \in 2^{\mathrm{I}} \; \varphi_{ISD}(E1, E2) = \{e | \exists \, e1 \in E1, \exists \, e2 \in E2(e1 \,\mathtt{m}\, e2 \;\wedge\; e = e1 \;\cup_{\mathrm{c}}\; e2)\}$,
$\alpha_{SD} = \alpha_S \, \varphi_{SD} \, \alpha_D$ and $\varphi_{SD} \equiv \varphi_{ISD}$.

These two abstract patterns represent a general up-and-down and down-and-up behavior of clinical variables; the addition of the Stationary TA allows one to catch behaviors presenting a sort of plateau between the Increasing and Decreasing trend episodes. Trends are detected through a sliding window algorithm, fixing a threshold of the 5% on the slope change to detect the patterns of increase and decrease. This choice is justified by the fact that a change in pressure values or in heart rate of 5 out of 100 units is already significant from a clinical viewpoint; such variation can be properly detected by considering the precision of the measurement instrument.

In order to obtain a description of the time course of each variable over different dialysis sessions, the data of all the dialysis sessions of the same patient were concatenated to get a single time series. We thus obtained 36 time series for each variable, each one with a number of measurements which depends on the number of treatments for the specific patient; in particular, we collected an average number of 70 treatments per patient and of 2,452 points per time series (see Table 6 for a detailed description of the dataset).

Temporal rules were evaluated on the complete dataset to discover significant relationships between opposite patterns occurring during the same dialysis session. From a computational viewpoint, the rule extraction algorithm works by searching opposite patterns belonging to the set *AoI* in the antecedents with respect to the consequents (e.g., *ISD* vs. *DSI*). The parameters in the triple *P* were set in order to extract temporal rules holding within a single dialysis session; considering an average treatment duration of 4 h, with measurements taken every 5 min, we set $LS = RS = 40$ and $G = 30$. Episodes starting during one treatment and ending in the consecutive one were automatically removed by the algorithm before the rule mining step.

Table 7 shows the results obtained fixing a threshold for the confidence, $Conf \geq 0.5$, and for the support, $Sup \geq 0.1$.

Interesting rules which describe relationships between complex patterns involving one or more variables were detected. The first rule extracts a contemporaneous pattern of *SP* and *DP*, in which a down and up pattern *DSI* is followed by an up and down pattern *ISD* of *HR*. From the clinical viewpoint, this rule highlights the occurrence of hypotension episodes taking place during dialysis treatments: when arterial blood pressure decreases, the organism reacts with an increase in the heart rate, which then goes back to normal values as soon as blood pressure increases. Other four similar rules, which relate *HR* with *SP* and *DP* were also found. These episodes are clinically relevant, since they correspond to the patient's response to blood pressure instability.

On the basis of these results, and in particular of the ones obtained for confidence and support, an interesting clinical question that may rise is whether there is a group of patients which are particularly prone to hypo or hypertensive episodes during hemodialysis. To tackle this problem, we ran the rule extraction algorithm on each patient separately and we then evaluated confidence and support of the obtained rules (results not shown). Following this strategy, we identified a group of 10 patients showing an high number of precedence rules involving blood pressure and heart rate, in which the variables present an opposite pattern. Table 8 shows the results obtained by running the rule extraction algorithm on this subset of 10 patients ($min\_conf = 0.7$, $min\_sup = 0.1$). The table shows a relevant improvement in the confidence and support of the rule in which a down and up pattern *DSI* in *SP* and *DP* is followed by an up and down pattern *ISD* for *HR*. In this last application, the proposed approach clearly

Table 6 Description of the hemodialysis dataset

Patient	# Dialysis	# Time points
1	85	2,742
2	52	1,520
3	91	3,784
4	77	2,887
5	80	2,776
6	80	2,679
7	92	2,896
8	77	2,463
9	22	946
10	91	3,685
11	75	2,776
12	90	2,963
13	71	2,804
14	72	2,935
15	91	3,305
16	83	3,443
17	69	2,903
18	82	2,641
16	90	2,861
20	88	3,061
21	65	2,169
22	83	2,810
23	85	3,378
24	75	2,861
25	81	2,827
26	68	2,037
27	84	3,146
28	22	706
29	86	2,963
30	74	2,625
31	80	2,241
32	14	198
33	78	2,509
34	1	32
35	33	972
36	33	749

Table 7 The rules derived from the analysis of the haemodialysis data (*ISD* and *DSI* stand for meeting episodes of patterns *Increasing, Stationary*, and *Decreasing*, in the specified order)

Operator: *PRECEDES*
$AoI = \{ISD_{SP}, ISD_{DP}, ISD_{HR}, DSI_{SP}, DSI_{DP}, DSI_{HR}\}$

Antecedent		Consequent		Confidence	Support
Parameter	Pattern	Parameter	Pattern		
SP *DP*	*DSI* *DSI*	*HR*	*ISD*	0.580	0.119
HR	*DSI*	*DP*	*ISD*	0.623	0.192
HR	*DSI*	*SP*	*ISD*	0.622	0.199
HR	*ISD*	*DP*	*DSI*	0.580	0.228
HR	*ISD*	*SP*	*DSI*	0.615	0.234

Table 8 The rules derived from the analysis of the haemodialysis data for 10 patients showing episodes of hypertension or hypotension in their dialysis history (*ISD* and *DSI* stand for meeting episodes of patterns *Increasing, Stationary* and *Decreasing*, in the specified order)

OPERATOR: *PRECEDES*
$AoI = \{ISD_{SP}, ISD_{DP}, ISD_{HR}, DSI_{SP}, DSI_{DP}, DSI_{HR}\}$

Antecedent		Consequent		Confidence	Support
Variable	Pattern	Variable	Pattern		
SP *DP*	*DSI* *DSI*	*HR*	*ISD*	0.728	0.154
HR	*DSI*	*SP*	*ISD*	0.734	0.342

shows its capability of answering to clinically relevant questions, together with extracting useful information from the data.

4.2.2 Reconstruction of gene regulatory relationships through gene expression data

The second study is about the attractive biological problem of inferring genetic regulatory relationships from DNA microarray gene expression data. DNA microarrays are a relatively new technique which allows the extraction of a genome-wide snapshot of gene activity under specific experimental conditions (Brown and Botstein 1999); by repeating observations at different time points it is possible to obtain the *gene expression profile*, i.e., a time series that describes the behavior of gene expression over a specific observation interval. Due mainly to economic reasons, temporal experiments with DNA microarrays give usually origin to very short time series (up to 50 time points), which need to be handled by specific methods designed to treat this kind of data. The aim of this study is to show how our algorithm can be adapted to deal with time series of gene expression data. In this domain, our algorithm could in fact be particularly suited since it allows the description of patterns of synchronization and precedence in gene expressions; such patterns might be the evidence of close relationships between genes. Moreover, by highlighting the relationships between synchronized gene sets, we can gain insight into the temporal sequence of macro-processes, potentially suggesting cause-effect relationships between the involved genes. An application of the method as a step in the process of deriving precedence temporal networks between genes involved in specific biological processes has been presented in (Sacchi et al. 2005b).

In this example we show the analyses performed on data coming from DNA microarray experiments on the human cell cycle,[4] presented in Whitfield et al. (2002) and available at `http://genome-www.stanford.edu/Human-CellCycle/Hela/`. From the whole dataset, we extracted 5 time series of 47 samples that correspond to a group of human genes which regulate the cell cycle (Tyson et al. 2001). Since these five genes are known to be characterized by a peak in their expression profiles taking place at different phases of the cell cycle, we focused on the extraction of rules between abstract patterns reflecting the intuitive notion of *peak*, i.e., $AoI = \{ID_{C\_A}, ID_{C\_B}, ID_{C\_E}, ID_{P27}, ID_{CDC25}, DI_{C\_A}, DI_{C\_B}, DI_{C\_E}, DI_{P27}, DI_{CDC25}\}$, where C_A, C_B, C_E, $P27$, and $CDC25$ denote the considered genes, as detailed in the following. These patterns are useful to highlight synchronization and phase shifts between genes during the selected process. The rules were derived with confidence $Conf = 1$ and support $Sup \geq 0.7$. These constraints are motivated by the nature of the problem: we are in fact dealing with a multivariate set of short time series, each gene corresponding to a single time series. The threshold on the confidence forces the algorithm to keep only rules where each episode of the antecedent is also involved in one episode of the rule, while the constraint on the support requires the time span of the rule to cover the 70% of the total observation period.

Rather interestingly, the most important known relationships between genes are automatically derived by the algorithm. Table 9 and Fig. 9 show some examples related to the gene for cyclin $E(C\_E)$: its activity is needed for the transition from phase G1 to phase S of the cell cycle. This gene is overexpressed at the G1–S phase boundary while it is degraded as the cell progresses through phase S. The expression profile of the gene which encodes for protein $P27$, which is able to bind to and to prevent the activation of cyclin E, is always in opposition to the one of cyclin E (Fig. 9a); this behaviour results into a temporal rule which expresses that a peak of one gene is found to always precede the peak of the other one. Cyclins A and B promote both cell cycle G1/S and G2/M transitions, and $CDC25$ is a protein which triggers the entry in the phase M from G2; the

[4] The cell cycle is the life cycle of the cells. It develops through four phases called G1, S, G2, and M. The last phase, M, corresponds to Mitosis, when the cell divides into two new cells. If a population of cells is observed over time, the overall behavior is therefore periodic.

Table 9 Examples of the rules extracted from the analysis of gene expression data of human cell cycle (*ID* and *DI* stand for meeting episodes of patterns *Increasing*, and *Decreasing*, in the specified order)

Operator: *PRECEDES*

$AoI = \{ID_{C\_A}, ID_{C\_B}, ID_{C\_E}, ID_{P27}, ID_{CDC25}, DI_{C\_A}, DI_{C\_B}, DI_{C\_E}, DI_{P27}, DI_{CDC25}\}$

Antecedent		Consequent		Confidence	Support
Gene	Pattern	Gene	Pattern		
P27	*ID*	Cyclin $E(C\_E)$	*DI*	1	0.915
Cyclin $A(C\_A)$ Cyclin $B(C\_B)$	*ID* *ID*	Cyclin $E(C\_E)$	*ID*	1	0.745
CDC25 Cyclin *A*	*ID* *ID*	Cyclin $E(C\_E)$	*ID*	1	0.745

expression time series for these genes reflect their synchronization, which is also translated by the algorithm into two temporal rules showing in the antecedent a contemporaneous peak for cyclin *A* and cyclin *B* and for cyclin *A* and *CDC*25. Such rules express also the periodical precedence between the complex patterns related to the three genes just mentioned and cyclin *E*, which instead is expressed in an earlier phase of the cell cycle (Fig. 9b, c).

5 Discussion and related work

The work presented in this article deals with several areas in the field of temporal data mining (TDM) (Lin et al. 2002; Roddick and Spiliopoulou 2002). The central goal of the article is to find interesting temporal rules between complex patterns found in a set of time series. The task of extracting temporal rules has been addressed by several authors: the Apriori-like technique originally exploited by Agrawal and Srikant (1995) to extract sequential patterns was further extended in Mannila and Toivonen (1996) to deal with the discovery of frequent episodes and episodes rules. Afterwards, several extensions on the extraction of temporal association rules and inter-transactional association rules were presented (Chen et al. 1998; Li et al. 2003; Tung et al. 2003). As regards the discovery of rules between temporal patterns, Guimarães and Ultsch (1999) and Guimarães et al. (2001) applied unsupervised neural networks to detect complex temporal patterns and generated temporal grammatical rules for a symbolic knowledge representation, underlining the usefulness of incorporating prior knowledge for the improvement of the performances of the algorithm.

The task of mining temporal rules on interval-based data has been tackled by several authors: after a definition of temporal pattern based on temporal relationships between interval-based events, Kam and Fu (2000) proposed an Apriori-like strategy to efficiently detect such patterns. In Villafane et al. (2000),

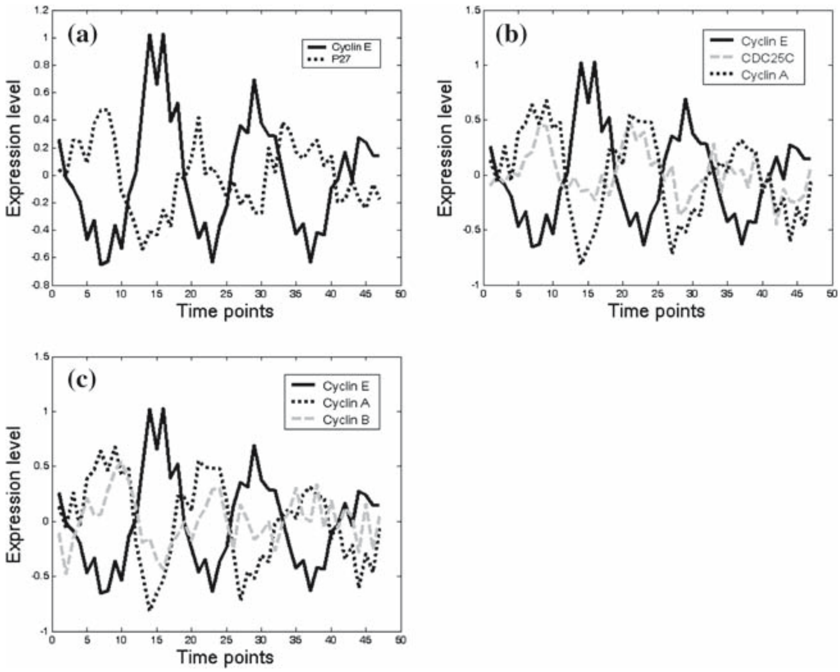

Fig. 9 Expression profiles for the genes involved in human cell cycle. (**a**) The expression profile of the gene which encodes for the protein *P*27, which is able to prevent the activation of cyclin *E*, is always in opposition to the one of cyclin *E*. (**b**) The expression time series for cyclin *A* and *CDC*25 reflect their synchronization; they are in fact both activated in the transition from phase G2 to M. The figure shows also a periodical precedence with cyclin *E*, which activity is needed in earlier phases of the cell cycle. (**c**) Cyclin *A* and cyclin *B* are both activated into the same phases of the cell cycle and their profile shows a phase shift with the one for cyclin *E*

the authors propose a mining technique to discover containment relationships in series of interval events; such events are derived from numerical time series through a quantization step. In Last et al. (2001), a general methodology for the entire process of knowledge discovery in time series databases, addressing both the preprocessing and the rule mining step, is presented. Cohen (2001) introduces the theory of fluent learning to extract common patterns (described as the ‘shape’ of episodes) in time series, a statistical technique which results well suited to deal with multivariate time series with binary variables. Höppner and Klawonn (2002b) and Höppner (2003) address the problem of discovering informative temporal rules on a given sequence of labeled intervals, making more flexible the definition of temporal pattern stated in Kam and Fu (2000). Starting from the work of Höppner and from an algorithm proposed in Lin and Lee (2005) for the discovery of temporal patterns from interval-based data, Winarko and Roddick recently proposed a new method for extracting frequent temporal patterns and then infer temporal rules from such patterns (Winarko and Roddick 2005).

Another recent work (Papapetrou et al. 2005) offers a novel formalization of the problem of mining frequent arrangements of temporal intervals. The method acts on a database of sequences of events, where each event occurs during a time interval, thus removing the assumption of handling only instantaneous events.

Among the above mentioned previous works, Höppner suggests a formulation of the problem of extracting rules from temporal patterns which is the closest to the one described in our article. In particular, the author proposes qualitative features to divide up the time series into segments accordingly, and a method for the mining of temporal patterns from which informative rules are derived. In Höppner (2003), an introduction is provided on how to learn qualitative labels (usually trends) from time stamped data, mentioning techniques such as clustering, piecewise linear approximation, smoothing and wavelets. By formalizing the framework of knowledge-based temporal abstractions, we herein introduce a more general environment, that allows one to extract qualitative labels from temporal data, potentially exploiting each of the methods introduced by Höppner. This technique allows us to embed into the same process both the extraction of episodes characterized by qualitative labels and a sound and general definition of the concept of pattern. Moreover, the process of deriving complex patterns is naturally contained into the definition of complex temporal abstractions and has not to be computed online with the rule mining step. The novel, knowledge-based definition of the set *AoI* is in fact performed prior to the rule extraction step; this, besides reducing the number of potential patterns to be retrieved in the time series, allows us also to avoid an online candidate generation during the process of rule extraction.

Following the ideas of Bellazzi et al. (2005), in our work we started from raw time series introducing a step for the extraction of an interval-based representation based on the formalism of TAs; also the *AoI* set is made up following such representation. In previous proposals, even when a qualitative representation of the time series is suggested (Höppner and Klawonn 2002b) or achieved through TAs (Bellazzi et al. 2005), the representation that is considered is always of a basic nature (e.g., intervals of *Increasing, Decreasing*, or *Stationary* trends for a single time series) and the temporal rules are always extracted between such simple patterns. In our work we extended this framework towards a representation through complex TAs and a rule detection which allows complex patterns in both the antecedent and the consequent of the rule itself.

6 Conclusions

In this article we presented a new kind of temporal association rule and the related algorithm for the extraction of temporal relationships between complex patterns defined over time series. The method is based on a qualitative representation of basic trends which relies on the formalism of knowledge-based Temporal Abstractions, which is coupled with an Apriori-like technique for the efficient mining of frequent occurrences of precedence between episodes of complex patterns. The knowledge-driven procedure that is applied throughout the article for the extraction of results can handle background knowledge in an explicit way; this facilitates explanation and user control on the output. The choice of TAs for the representation of the basic temporal information (i.e., basic trends) leads to a simple translation of the user's notion of interesting patterns into data description and to an intuitive understanding of the results. The presented approach can be used in a variety of application domains, and it was already tested on two different biomedical problems.

The great flexibility that characterizes the algorithm allows some immediate extensions, such as the consideration of both Trend and State TAs for the representation of the temporal profiles or the possibility of taking into account different types of relationships between the episodes. Moreover, we also plan to introduce an automated generation of the set *AoI* resulting from a pre-processing of the time series aimed at finding the most interesting qualitative behaviors in the whole set of profiles.

References

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proceedings of the international conference on very large databases. Santiago, Chile, pp 487–499

Agrawal R, Srikant R (1995) Mining sequential patterns. In: Proceedings of the 11th international conference on data engineering. Taipei, Taiwan, pp 3–14

Allen JF (1984) Towards a general theory of action and time. Artif Intell 23:123–154

Bellazzi R, Larizza C, Riva A (1998) Temporal abstractions for interpreting diabetic patients monitoring data. Intell Data Anal 2:97–122

Bellazzi R, Larizza C, Magni P, Bellazzi R (2003) Quality assessment of hemodialysis services through temporal data mining. In: Proceedings of 9th conference on artificial intelligence in medicine (AIME). Protaras, Cyprus, pp 11–20

Bellazzi R, Larizza C, Magni P, Bellazzi R (2005) Temporal data mining for the quality assessment of hemodialysis services. Artif Intell Med 34(1):25–39

Brown PO, Botstein D (1999) Exploring the new world of the genome with DNA microarrays. Nat Genet 21(1 Suppl):33–37

Chen X, Petrounias I, Heathfield H (1998) Discovering temporal association rules in temporal databases. In: Proceedings of the international workshop on issues and applications of database technology (IADT'98). Berlin, Germany, pp 312–319

Chiu B, Keogh E, Lonardi S (2003) Probabilistic discovery of time series motifs. In: Proceedings of the 9th ACM SIGKDD international confernence on knowledge discovery and data mining. Washington, USA, pp 493–498

Cohen PR (2001) Fluent learning: elucidating the structure of episodes. In Hoffmann F, Hand D, Adams N, Fisher D, Guimaraes G (eds), Proceedings of the 4th international conf. on Intelligent data analysis (IDA 2001). Cascais, Portugal, pp 268–277

Combi C, Chittaro L (1999) Abstraction on clinical data sequences: an object-oriented data model and a query language based on the event calculus. Artif Intell Med 17(3):271–301

Guimarães G, Ultsch A (1999) A method for temporal knowledge conversion. In: Proceedings of the 3rd international symposium on advances in intelligent data analysis (IDA 1999). Amsterdam, The Netherlands, pp 369–382

Guimarães G, Peter JH, Penzel T, Ultsch A (2001) A method for automated temporal knowledge acquisition applied to sleep-related breathing disorders. Artif Intell Med 23(3):211–37

Haimowitz IJ, Kohane IS (1993) Automated trend detection with alternate temporal hypotheses. In: Proceedings of the 13th joint conference on artificial intelligence (IJCAI 1993). Chambery, France, pp 146–151

Hau DT, Coiera EW (1997) Learning qualitative models of dynamic systems. Mach Learn 26(2–3):177–211
Höppner F, Klawonn F (2002a) Learning rules about the development of variables over time. In: Leondes CT (ed), Intelligent systems — techniques and applications, Vol 4. CRC Press, pp 201–228
Höppner F, Klawonn F (2002b) Finding informative rules in interval sequences. Intell Data Anal Int J 6(3):237–256
Höppner F (2003) Knowledge discovery from sequential data. PhD thesis, Technical University Braunschweig, Germany
Kam PS, Fu AWC (2000) Discovering temporal patterns for interval-based events. In: Proceedings of 2nd international conference on data warehousing and knowledge discovery (DaWaK). London, UK, pp 317–326
Keogh E, Chu S, Hart D, Pazzani M (2004) Segmenting time series: a survey and novel approach. Data mining in time series databases, World Scientific Publishing Company, pp 1–22
Kuipers B (1986) Qualitative simulation. Artif Intell 29(3):289–338
Larizza C, Moglia A, Stefanelli M (1992) M-HTP: a system for monitoring heart transplant patients. Artif Intell Med 4(2):111–126
Last M, Klein Y, Kandel A (2001) Knowledge discovery in time series databases. IEEE Trans Syst Man Cybernet 31(1):160–169
Li Y, Ning P, Wang XS, Jajodia S (2003) Discovering calendar-based temporal association rules. Data Knowl Eng 44(2):193–218
Lin W, Orgun MA, Williams GJ (2002) An Overview of Temporal Data Mining. In: Proceedings of the 1st Australian data mining workshop (ADM02). Canberra, Australia, pp 83–90
Lin M-Y, Lee S-Y (2005) Fast discovery of sequential patterns through memory indexing and database partitioning. J Inf Sci Eng 21(1):109–128
Mannila H, Toivonen H (1996) Discovering generalized episodes using minimal occurrences. In: Proceedings of the 2nd international conference on knowledge discovery and data mining (KDD-96). Portland, Oregon, pp 146–151
Mörchen F (2006) Time series knowledge mining. PhD thesis, Philipps University, Marburg, Germany
Papapetrou P, Kollios G, Sclaroff S, Gunopulos D (2005) Discovering frequent arrangements of temporal intervals. In: Proceedings of the 5th IEEE international conference on data mining (ICDM-05). Houston, Texas, pp 354–361
Roddick JF, Spiliopoulou M (2002) A survey of temporal knowledge discovery paradigms and methods. IEEE Trans Knowledge Data Eng 14(4):750–767
Sacchi L, Bellazzi R, Larizza C, Magni P, Curk T, Petrovic U, Zupan B (2005a) TA-clustering: cluster analysis of gene expression profiles through Temporal Abstractions. Int J Med Inform 74(7–8):505–517
Sacchi L, Bellazzi R, Porreca R, Larizza C, Magni P (2005b) Precedence Temporal Networks from gene expression data. In: Proceedings of 18th IEEE symposium on computer-based medical systems (CBMS). Dublin, Ireland, pp 109–114
Shahar Y, Musen MA (1996) Knowledge-based temporal abstraction in clinical domains. Artif Intell Med 8(3):267–298
Shahar Y (1997) A framework for knowledge-based temporal abstraction. Artif Intell 90(1–2): 79–133
Tung AKH, Lu H, Han J, Feng L (2003) Efficient mining of intertransaction association rules. IEEE Trans Knowl Data Eng 15(1):43–56
Tyson JJ, Chen K, Novak B (2001) Network dynamics and cell physiology. Nat Rev Mol Cell Biol 2(12):908–916
Villafane R, Hua KA, Tran D, Maulik B (2000) Knowledge discovery from series of interval events. J Intell Inform Syst 15(1):71–89
Whitfield ML, Sherlock G, Saldanha AJ, Murray JI, Ball CA, Alexander KE, Matese JC, Perou CM, Hurt MM, Brown PO, Botstein D (2002) Identification of genes periodically expressed in the human cell cycle and their expression in tumors. Mol Biol Cell 13(6):1977–2000
Winarko E, Roddick JF (2005) Discovering richer temporal association rules from interval-based data. In: Proceedings of the international conference on data warehousing and knowledge discovery (DaWaK). Copenhagen, Denmark, pp 315–325