A Temporal Data Mining Framework for Analyzing Longitudinal Data

Corrado Loglisci, Michelangelo Ceci, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70125 Bari - Italy

Abstract. Longitudinal data consist of the repeated measurements of some variables which describe a process (or phenomenon) over time. They can be analyzed to unearth information on the dynamics of the process. In this paper we propose a temporal data mining framework to analyze these data and acquire knowledge, in the form of temporal patterns, on the events which can frequently trigger particular stages of the dynamic process. The application to a biomedical scenario is addressed. The goal is to analyze biosignal data in order to discover patterns of events, expressed in terms of breathing and cardiovascular system time-annotated disorders, which may trigger particular stages of the human central nervous system during sleep.

1 Introduction

Domains of the real world that evolve over time, such as biomedical processes, human beings behaviours, physical and natural phenomena, can be described by a finite set of variables whose repeated measurement generates a particular class of multidimensional time-series known as *longitudinal data* [10]. Normally, longitudinal data represent the complete evolution or dynamics of a process over time and therefore they can convey relevant information. However, the complexity of longitudinal data makes their interpretation difficult and resorting to automatic techniques of analysis becomes necessary. Traditionally, most of attention has been paid by the classical computational statistics techniques, which anyway can suffer from problems coming from the hight dimensionality of the collected data, from heterogeneity of data types and from the need of handling the intrinsic temporal nature of longitudinal data.

In this regard, a relevant role can be played by data mining approaches. One of the first proposed methods is the querying and mining system described in [3] where the authors investigated three different tasks for temporal association rules discovery, namely the discovery of valid time periods during which association rules hold, the discovery of possible periodicities that association rules have, and the discovery of association rules with temporal features, where the analyzed data are represented in the simplified representation of the transactions. Another interesting approach is presented in [8] which reports a methodology to pre-process time-series and discover frequent patterns from the pre-processing

A. Hameurlain et al. (Eds.): DEXA 2011, Part II, LNCS 6861, pp. 97–106, 2011.

results. In particular, the patterns are organized according to an hierarchical structure built on the temporal concepts of duration, coincidence and partial order.

In this paper, we propose a temporal data mining approach that aims at supporting the tasks of analyzing and interpreting the evolution of a dynamic process. It mines time-varying data and discovers patterns of time-annotated *complex events* which can trigger particular *stages* of the process. A complex event is associated to a variation or change while a stage corresponds to a specific state of the process which holds in a period of time. Two consecutive stages represent two different states and together they depict a transition in the process. So, given two consecutive stages, we assume that whatever happens in the first stage may affect the second one, and, since two consecutive stages are different each other, the events which occur in the first stage and do not occur in the second one can be responsible of the transition of the process towards the second stage. Therefore, the transition can be ascribed to these events.

Patterns are discovered from the events detected on a collection of pairwise stages of interest. Such a collection is properly created in order to consider only pairs of stages which depict similar transitions. The usage of pattern discovery is therefore addressed to find out the most frequent (and maybe significant) complex events which can determine similar transitions and, thus, can trigger analogous stages.

The paper is organized as follows. In next section we define the problem in terms of four sub-problems. The computational solution for them is described in Section 3. An application to the case of a biomedical scenario is presented in Section 4. Finally, conclusions are drawn.

2 Problem Formulation

Before formally defining the problem of interest, we introduce some necessary concepts. Let $P : \{a_1, \ldots, a_m\}$ be the finite set of real-valued variables (e.g., $\{$*blood oxygen, heart rate, respiration rate*$\}$), longitudinal data form a collection Mp of time-ordered measurements of the variables in P.

A stage S_j is a 4-tuple $S_j = \langle ts_j, te_j, C_j, SV_j \rangle$, where $[ts_j..te_j]$ $(ts_j, te_j \in \tau, ts_j \leq te_j)$[1] represents the time-period of the stage, while $C_j : \{f_1, f_2, \ldots\}$ is a finite set of *fluents*, namely facts or properties in terms of variables P that are true during the time-period $[ts_j..te_j]$. SV_j is the set $\{sv_1, \ldots, sv_k, \ldots, sv_m\}$ containing m symbolic values such that sv_k is a high-level description of the parameters $a_k \in P$ during $[ts_j..te_j]$.

An example of stage is $S_1 : \langle t_1, t_{10}, \{$ *blood oxygen* $\in$ *[6500;6700], heart rate* $\in$ *[69;71], respiration rate* $\in$ *[2300;5500]*$\}$, $\{$*blood oxygen is INCREASE, heart rate is STEADY, respiration rate is INCREASE*$\}$ $\rangle$ which can be interpreted as follows: S_1 is associated with the period of time $[t_1, t_{10}]$ and is characterized by the fact (fluent) that the variables *blood oxygen*, *heart rate* and *respiration rate*

[1] τ is a finite totally ordered set of time-points. Henceforth, the corresponding order relation is denoted as $\leq$.

have values respectively in [6500; 6700], [69; 71], [2300; 5500] and have increasing, steady and increasing trend, respectively.

An event e is a signature $e = \langle t_F, t_L, Ea, IEa, SEa \rangle$, where $[t_F..t_L]$ is the time-interval when event e occurs $(t_F, t_L \in \tau)$, $Ea : \{ea_1, \ldots, ea_k, \ldots, ea_{m'}\}$ is a subset of P and contains m' distinct variables which take values in the intervals $IE_a : [inf_1, sup_1], \ldots, [inf_k, sup_k], \ldots [inf_{m'}, sup_{m'}]$, respectively, during $[t_F..t_L]$. Finally, $SE_a : \{sv_1, \ldots, sv_k, \ldots, sv_{m'}\}$ is a set of m' symbolic values associated to Ea. In particular, IE_a is a quantitative description of the event, while SE_a is a qualitative representation of the trend of values taken by each ea_k during $[t_F..t_L]$.

Two examples of events are $e_1 : \langle t_1, t_5, \{bloodoxygen\}, \{[6300; 6800]\}, \{DECREASE\}\rangle$ and $e_2 : \langle t_6, t_{10} \{bloodoxygen\}, \{[6600; 7000\}, \{INCREASE\}\rangle$ which can interpreted as follows: e_1 (e_2) is associated with the time-period $[t_1, t_5]$ $([t_6, t_{10}])$ during which the variables *blood oxygen* ranges in [6300; 6800] ([6600; 7000]) and has a decreasing (increasing) trend. Trivially, a sequence $\langle e_1, e_2 \rangle$ is an ordered list of events when, given $[t_{F1}..t_{L1}]$, $[t_{F2}..t_{L2}]$ of e_1 and e_2 respectively, t_{F1} is the immediate predecessor of t_{F2} in τ.

The notions above introduced suggest to resort to representation formalisms able to suitably handle the complex formulation of the events. Indeed, we resort to first-order logic formalism and approaches synthesized in *Inductive Logic Programming* (ILP)[9] which permit us to naturally deal with the intrinsic complexity of the longitudinal data and handle the structural and relational aspects of events and sequences as above defined. The events are modeled in a logical formalism (Datalog language [2]) and represented as *ground atoms.* A ground atom is an n-ary logic predicate symbol applied to n constant terms, while a non-ground atom is an n-ary predicate symbol applied to n constant and variable terms. For instance the sequence e_1, e_2 before introduced is so represented:

sequence(seq$_1$). event(seq$_1$,e$_1$). time_tF(e$_1$,1). time_tL(e$_1$,5). parameter_of(e$_1$,p$_1$). is_a (p$_1$,blood_oxygen). value_interval(p$_1$,'[6300;6800]'). symbolic_value(p$_1$,'DECREASE'). event(seq$_1$,e$_2$). time_tF (e$_2$,6). time_tL(e$_2$,10). parameter_of(e$_2$,p$_2$). is_a(p$_2$, blood_oxygen). value_interval(p$_2$,'[6600;7000]'). symbolic_value(p$_2$,'INCREASE').

where *sequence(seq$_1$)* is the atom which identifies the sequence seq$_1$ through the predicate *sequence*(); *event(seq$_1$, e$_1$)* is the atom which relates the sequence seq$_1$ to the event e$_1$ through *event*(); *time_tF (e$_1$, 1)* is the atom which assigns the specific value 1 to the attribute *time_tF* of e$_1$ through *time_tF*(); *variable_of(e$_1$, p$_1$)* is the atom which relates the event e$_1$ to the variable p$_1$ through *parameter_of*(); *is_a(p$_1$, blood_oxygen)* is the atom which assigns a specific value *blood_oxygen* to p$_1$ through *is_a*(), *value_interval(p$_1$,'[6300;6800]')* is the atom which assigns a specific interval of values [6300;6800] to p$_1$ through *value_interval*() and *symbolic_value(p$_1$,'DECREASE')* is the atom which assigns a specific symbolic value DECREASE to p$_1$ through *symbolic_value*().

We can now formally define a temporal pattern: a temporal pattern T_P is a set of atoms $p_0(t_0^1)$, $p_1(t_1^1, t_1^2)$, $p_2(t_2^1, t_2^2)$, $\ldots$, $p_r(t_r^1, t_r^2)$, where p_0, p_i, $i = 1, \ldots, r$, are logic predicate symbols while t_i^j are either constants or variables, which identify sequences, events or variables in T_P. Among these logic predicates we can have predicates able to express possible temporal relationships between two events e$_1$, e$_2$ according to the Allen temporal logic[1]. For instance, the temporal pattern

Tp: sequence(Q), event(Q, E1), event(Q, E2), before(E1, E2), parameter_of(E1, P1), is_a(P1, blood_oxygen), value_interval(P1,'[6300;7000]'), symbolic_value(P1, steady), is_a (P2, respiration_rate), value_interval(P2,'[2300;5500]'), symbolic_value(P2, strong_increase)

expresses the fact that, for a subset of sequences, the event E_1 is followed by E_2, where in E_1 the blood oxygen has steady trend and ranges in [6300;7000] while in E_2 the respiration rate is strongly increasing with values in [2300;5500].

Considering the concepts so far defined, the problem of interest in the proposed framework can be divided in four sub-problems formalised as follows:

1. *Given*: longitudinal data $Mp : \{Mp_{t1}, Mp_{t2}, \ldots, Mp_{tn}\}$; *Find*: a finite set $S : \{S_1, S_2, \ldots\}$ of consecutive stages which represent distinct sub-sequences of *Mp*.
2. *Given*: a criterion CS to collect pairwise stages of interest from S; *Find*: a collection R of pairwise stages (S_j, S_{j+1}) which satisfy the criterion CS.
3. *Given*: the collection R; *Find*: a set *ES* of sequences $\langle e_1, e_2, \ldots\rangle$ of events for each pair (S_j, S_{j+1}) in R.
4. *Given*: the set *ES* and a user-defined threshold *minF*; *Find*: temporal patterns in *ES* whose support exceeds the threshold *minF*.

A computational solution to these sub-problems is described in Section 3.

3 Temporal Data Mining Framework

3.1 Determination of Stages

A stage can be seen as one of the steps of dynamics characterized by numerical (C_j), symbolic ($\{sv_1, ..., sv_h, ..., sv_m\}$) and temporal ($[ts_j..te_j]$) properties. In other words, a stage corresponds to one of the distinct segments of *Mp*. The components ts_j, te_j, C_j are obtained by resorting to the method we proposed in [5] which is here shortly described. The periods of time $[ts_j..\ te_j]$ are obtained by means of a two-stepped technique of temporal segmentation. In particular, it first identifies a series of change-points and recursively partitions *Mp* in a succession of multi-variate segments until the variability of each variable a_h does not exceed a user-defined threshold ω. Then, it merges together consecutive segments if the variables in the segments are statistically correlated w.r.t. user-defined maximum threshold ρ of correlation. The duration $[ts_j..te_j]$ of each stage is forced to be bigger than a user-defined minimum threshold *minSD*. This segmentation produces a sequence of segments of *Mp* that differ each other, and it guarantees that two consecutive segments have different fluents: given three consecutive segments, $[ts_{j-1}..te_{j-1}]$, $[ts_j..te_j]$, $[ts_{j+1}..te_{j+1}]$, the fluents C_j associated to $[ts_j..te_j]$ are conditions which hold in $[ts_j..te_j]$ but neither in the previous $[ts_{j-1}..te_{j-1}]$ nor in the next $[ts_{j+1}..te_{j+1}]$ segments. The generation of fluents C_j is solved with the inductive logic programming approach used in [5] which permits to determine C_j as the set of interval-valued atomic formulae which *characterizes* the measurements included in $[ts_j..\ te_j]$ and *discriminates* them from those of $[ts_{j-1}..\ te_{j-1}]$ and $[ts_{j+1}..\ te_{j+1}]$. This way, we can determine each stage of dynamics and distinguish it from each other with a rigorous description. Finally, the values of the elements SV_j of S_j are derived by means a function $\Theta : \Pi \rightarrow \Lambda$ which provides an high-level representation $\lambda \in \Lambda$ of the most relevant features $\pi \in \Pi$ of data: Θ returns, for each variable a_k, a representation of the slope of the regression line built on the values taken by a_k in the time interval $[ts_j..te_j]$. For instance, the slope values ranging in the interval (0.2, 1] are described as INCREASE.

3.2 Collection of Pairwise Stages

A collection R of pairwise stages is properly created in order to study the transition between similar pairs of stages through the discovery of the events which most frequently trigger analogous stages.

Pairwise stages appropriate for R are identified on the basis of a similarity value: pairs whose first stages and second stages have similarity value which exceeds a user-defined numerical threshold CS ($CS \in [0; 100]$) are considered. For instance, two pairs (S_j, S_{j+1}), (S_k, S_{k+1}) are collected in R if the similarity between S_j and S_k and the similarity between S_{j+1} and S_{k+1} exceeds CS. In this work the similarity between two stages S_j and S_k corresponds to the similarity between their fluents C_j, C_k[2] under the assumption that the symbolic values SV_j, SV_k are identical. Since the fluents are sets of interval-valued data (section 3.1), the similarity between C_j and C_k is so formulated: $Sim(C_j, C_k) = (\sum\limits_{f_j \in C_j, f_k \in C_k} (1 - Diss(f_j, f_k))/(|C_j| * |C_k|)) * 100$, where $f_j(f_k)$ is a single interval-valued formula of C_j (C_k). To compute $Diss(f_j, f_k)$ we resort to dissimilarity functions specific for interval-valued data. In particular, we consider the Gowda and Diday's [4] dissimilarity measure defined as: $Diss(f_j, f_k) = \sum\limits_{h=1...|P|} \delta(f_{j_h}, f_{k_h})$, where, f_{j_h}, f_{k_h} are the intervals assumed by the parameter a_h, $|P|$ is number of intervals (variables), and $\delta(f_{j_h}, f_{k_h})$ is obtained considering three types of dissimilarity measures incorporating different aspects of similarity, namely $\delta(f_{j_h}, f_{k_h}) = \delta_\pi(f_{j_h}, f_{k_h}) + \delta_s(f_{j_h}, f_{k_h}) + \delta_c(f_{j_h}, f_{k_h})$, $(\delta_\pi, \delta_c,\ \delta_s \in [0, 1])$. It should be noted that several collections of similar transitions can be actually created from the pairs of stages in S: the resulting collection R is selected by the user in the set of possibly overlapping collections.

3.3 Detection of Complex Events

Once the collection R of pairwise stages has been identified, for each pair (S_j, S_{j+1}) we look for events which may trigger the transition from S_j to S_{j+1}.

[2] The notion of similarity between two stages does not concern the time-periods $[ts_j..te_j]$, i.e., two stages can be similar although they are associated to different time-periods.

Events are detected by resorting to the method we proposed in [6] which permits us to exploit the assumption for which events occurring during the time interval $[ts_j..te_j]$ should not occur in $[ts_{j+1}..te_{j+1}]$. The blueprint is to mine first candidate events then to select from these the events deemed *statistically interesting.* The algorithm for mining candidate events $\{e \mid e = \langle t_F, t_L, Ea, IEa, SEa\rangle\}$ proceeds by iteratively scanning the measurements included in the stages S_j (i.e., $\{Mp_{ts_j}, \ldots, Mp_{te_j}\}$) and S_{j+1} (i.e., $\{Mp_{ts_{j+1}}, \ldots, Mp_{te_{j+1}}\}$) with two adjacent time-windows which slide back in time. The candidates are identified by finding variations in the measurements between the windows w and w'. At the first iteration, the time-windows w, w' (w' immediately follows w) correspond to the last part of S_j and to the complete S_{j+1}, respectively. If a candidate is found then the next candidate is searched for the pair (w'',w), where the new time-window w'' has the same size of w. Otherwise, the next candidate is searched for the pair (w'',w'), where w'' is strictly larger than w. At the end of a single scan a sequence of candidates is obtained.

The intuition underlying the detection of candidate events for a given couple of windows (w, w') is that the intrinsic dependence of two variables in P may change between the two adjacent time-windows. This idea is implemented in the following strategy: for each variable a_i two multiple linear regression models are built on the remaining variables in P by considering the distinct measurements in w and w' respectively:

$$a_i = \beta'_0 + \beta'_1 a_1 + \ldots + \beta'_{i-1} a_{i-1} + \beta'_{i+1} a_{i+1} + \ldots + \beta'_m a_m,$$
$$a_i = \beta''_0 + \beta''_1 a_1 + \ldots + \beta''_{i-1} a_{i-1} + \beta''_{i+1} a_{i+1} + \ldots + \beta''_m a_m,$$

The couple of regression models which guarantees the lowest predictive information loss is selected. Let a_h be the variable for which the lowest predictive information loss is obtained, the set of parameters $Ea = \{a_k \in P - \{a_h\} \mid \ |\beta'_k - \beta''_k| \leq \sigma_k\}$[3] is selected and associated with the time window $w : [t_F..t_L]$ to form the event $e : \langle t_F, t_L, Ea, IEa, SEa\rangle$. The set Ea is further filtered in order to remove those parameters for which no interval of values which discriminates the measurements in w from those in w' can be generated. This permits also to determine the element IEa. In particular, for each $a_k \in Ea$ the interval $[inf_k, sup_k]$ is computed by taking the minimum (inf_k) and maximum (sup_k) value of a_k in w. If $[inf_k, sup_k]$ is weakly consistent with respect to values taken by a_k during the time window w' then a_k is kept, otherwise it is filtered out. Weak consistency is verified by computing the weighted average of the zero-one loss function on the measurements in w', where weights decrease proportionally with the time points in w'. Finally, the filtered set of m' variables will be associated with a set of intervals $\{[inf_1, sup_1], \ldots, [inf_k, sup_k], \ldots, [inf_{m'}, sup_{m'}]\}$, which corresponds to the quantitative description IEa of the event $e : \langle t_F, t_L, Ea, IEa, SEa\rangle$. The set $SEa : \{sv_1, \ldots, sv_k, \ldots, sv_m\}$ is determined through the same technique of temporal abstraction introduced in the section 3.1. It contains a symbolic value

[3] σ_k is automatically determined and is the standard deviation of the k-th coefficient of linear regression models computed on non-overlapping time-windows of size $t_L - t_F$ over (S_j, S_{j+1}).

for each a_k and each sv_k denotes the slope of the regression line built on the data in $[t_F..t_L]$.

Once the candidates for each single pair (S_j, S_{j+1}) in R have been generated, the sequence with the most statistically interesting events is identified by selecting the *most supported* events. An event e_u is *most supported* if it meets the following two conditions: 1) there exists a set of candidates $\{e_1, e_2, \ldots, e_t\}$ which contains the same information of e_u, that is: $\forall e_q, q = 1, \ldots, t$, $e_q \neq e_u$, the set of parameters Ea associated to e_q includes the set of parameters associated to e_u, the time interval $[t_F, t_L]$ associated to e_q includes the time interval associated to e_u, and, finally, the set of symbolic values SEa and the intervals IEa associated to the parameters of e_q coincide; 2) no event e_v exists whose information is contained in a set of candidates $\{e_1, e_2, \ldots, e_{t'}\}$ with $|\{e_1, e_2, \ldots, e_{t'}\}| > |\{e_1, e_2, \ldots, e_t\}|$. The *support* of the event e_u is computed as follows: let $\{e_1, e_2, \ldots, e_z\}$ be the set of candidates such that the time interval associated to each of them contains that of e_u and $\{e_1, e_2, \ldots, e_t\}$ be the set of candidates as described at the point 1), then the support of e_u is $supp(e_u) = (t + 1)/z$. The sequence of the most supported events for each pair of disease stages $(S_j, S_{j+1}) \in R$ forms the set ES of sequences of events.

3.4 Discovery of Temporal Patterns

Discovery of temporal patterns from ES is performed by resorting to the ILP method for frequent patterns mining implemented in SPADA [7]. The sequences generated in the section 3.3 are modeled with the logic predicates introduced in the section 2 and stored as sets of ground atoms in the extensional part D_E of a deductive database D [2]. The intensional part D_I of the database D is defined with the predicates based on Allen temporal logic [1]: D_I represents background knowledge on the problem (e.g., precedence relationships between two events through the predicate $before()$) and allows to entail additional atoms by applying these predicates to the extensional part. For example, give two sample sequences $seq1 : \langle e_1, e_2 \rangle$, $seq2 : \langle e_3, e_4 \rangle$ the extensional part D_E of D would include the following ground atoms: *sequence(seq1). sequence(seq2).*

event(seq1,e_1). event(seq1,e_2).event(seq2,e_3).

event(seq2,e_4). time_tF(e_1,10). time_tL(e_1,15). time_tF(e_2,22). time_tL(e_2,25).

time_tF(e_3,90). time_tL(e_3,110). time_tF(e_4,170). time_tL(e_4,190).

where the constants $seq1$ and $seq2$ denote two distinct sequences, while the constants e_1, e_2, e_3, e_4 identify four events. The intensional part D_I is formulated as the logic program:

before(E1, E2) ← event(S, E1),event(S, E2), E1 ≠ E2, time_tL(E1,T1), time_tF(E2,T2), T1<T2, not(event(S, E3), E3≠ E1, E3≠ E2, time_tF(E3,T3F), time_tL(E3,T3L), T1<T3F, T3L<T2)

by considering the atoms in D_E the ground atoms *before(e1, e2), before(e3, e4)* are entailed and added to D_E.

By following the level-wise method integrated in SPADA, the process of temporal patterns discovery performs a search in the space of patterns and finds out patterns whose support is greater than the user-defined threshold *minF* (frequent patterns) while it prunes those with support less than *minF* (infrequent patterns). The support of a pattern P is the percentage of sequences in D which covers the pattern P. The implementation of the anti-monotonicity of the support in the system guarantees the effectiveness of the level-wise method.

4 Application to Biomedical Data

In this section we explore the applicability of the proposed framework to a scenario of biomedicine. In particular, we focus on the analysis of data observed during a polysomnography, namely longitudinal data which describe the dynamic process of the human sleep, in order to investigate sleep disorders. Sleep disorders represent an issue of great importance and widely investigated in medicine because some serious diseases are accompanied by typical sleep disturbances. This attracts the interest of several scientific communities and, in this work, it is studied to discover patterns of events, in terms of breathing and cardiovascular system time-annotated disorders, which may trigger particular stages of the human central nervous system during sleep.

Dataset Description. The dataset[4] has been created by sampling measurements at 1 second of a patient from 21.30 p.m. to 6.30 a.m. Physiological parameters are *eeg* (electroencephalogram), *leog*, *reog* (electrooculograms), *emg* (electromyogram), *ecg* (electrocardiogram), *airflow*, (nasal respiration), *thorex* (thoracic excursion), *abdoex* (abdominal excursion), *pr* (heart rate) and *saO2* (arterial oxygen saturation). Where, *ecg, airflow, thorex, abdoex, pr, saO2* describe the cardiovascular and respiratory systems, while *eeg, leog, reog, emg* describe the central nervous system.

Results. Different sets S of disease stages are obtained by tuning *minSD* [5]. For each S, several collections R are created by setting CS to 60, 70, 80. Pattern are discovered from these collections by setting the threshold *minF* to 5% (Table 1). As we can see the number of discovered patterns (#patterns) is strongly dependent on the minimum duration of the stages. Indeed, the greater the stages, the higher the dissimilarity between the stages and the lower the number of similar pairwise stages (cardinality of R). This can be due to the fact that the fluents of stages with longer duration characterize and discriminate an higher number of physiological measurements. Therefore, they tend to be too specific for the set of data to characterize and very dissimilar from other fluents. In these cases, the cardinality of R is lower and this produces a set ES with a small number of sequences where it could be difficult to discover frequent patterns.

A first interesting result is produced when the minimum duration is set to 60 secs and CS to 60. In this case a set ES of nine sequences (as many the pairs of stages) of complex events is identified, while 579 frequent patterns are discovered. Among them, the most frequent one, which can trigger the transition depicted by the 9 pairs of stages, is so described:

[4] Accessible at `http://www.physionet.org/physiobank/`

Table 1. Patterns and stages discovered by tuning the minimum duration and CS

minimal duration (secs)	$\|S\|$	CS	$\|R\|$	#pattern
60	139	60	9 pairwise stages	579
		70	3 pairwise stages	112
		80	0	0
120	126	60	6 pairwise stages	63
		70	3 pairwise stages	34
		80	0	0
300	31	60	3 pairwise stages	7
		70	1 pairwise stages	4
		80	0	0

$$\begin{array}{l}
\quad \mathit{sequence}(S), \mathit{event}(E1, S), \mathit{event}(E2, S), \mathit{event}(E3, S), \mathit{before}(E1, E2), \mathit{before}(E2, E3),\\
\mathit{parameter\_of}(E1, P1), \mathit{is\_a}(P1, \mathit{abdoex}), \mathit{value\_interval}(P1,' [-1.412, 0.722]'), \mathit{symbolic\_value}(P1,\\
'\mathit{STRONG\_INCREASE}'), \mathit{parameter\_of}(E2, P2), \mathit{is\_a}(P2, \mathit{airflow}), \mathit{value\_interval}(P2,\\
'[-2.322, 3.482]'), \mathit{symbolic\_value}(P2,' \mathit{STRONG\_DECREASE}'), \mathit{parameter\_of}(E3, P3), \mathit{is\_a}(P3,\\
\mathit{saO2}), \mathit{value\_interval}(P3,' [94.013, 95.012]'), \mathit{symbolic\_value}(P3,' \mathit{DECREASE}') \qquad [\mathit{support} = 21.4\%]
\end{array}$$

This pattern involves both temporal predicates ($\mathit{before}()$), structural predicates (e.g., $\mathit{parameter\_of}()$) and properties (e.g., $\mathit{symbolic\_value}()$) and it is supported by a percentage of 21.4% of the total sequences.

Patterns with more predicates but with lower support are rather discovered at higher values of the minimal duration. For instance, one pattern mined when the minimal duration is 120 secs and CS=60 is the following:

$$\begin{array}{l}
\quad \mathit{sequence}(S), \mathit{event}(E1, S), \mathit{event}(E2, S), \mathit{event}(E3, S), \mathit{before}(E1,\ E2), \mathit{before}(E2, E3),\\
\mathit{before}(E3, E4), \mathit{parameter\_of}(E1, P11), \mathit{is\_a}(P11, \mathit{thorex}), \mathit{value\_interval}(P11,' [-3.984, 3.984]'),\\
\mathit{symbolic\_value}(P11,' \mathit{INCREASE}'), \mathit{parameter\_of}(E2, P21), \mathit{is\_a}(P21, \mathit{abdoex}), \mathit{value\_interval}(P21,\\
'[-1.757, 1.82]'), \mathit{symbolic\_value}(P21,' \mathit{STRONG\_INCREASE}'), \mathit{parameter\_of}(E2, P22), \mathit{is\_a}(P22,\\
\mathit{thorex}), \mathit{value\_interval}(P22,' [-0.91, 2.071]'), \mathit{symbolic\_value}(P22,' \mathit{STRONG\_INCREASE}'),\\
\mathit{parameter\_of}(E3,\ P3), \mathit{is\_a}(P3, \mathit{saO2}), \mathit{value\_interval}(P3,' [97.010, 98.009]'), \mathit{symbolic\_value}(P3,\\
'\mathit{DECREASE}'), \mathit{parameter\_of}(E4, P4), \mathit{is\_a}(P3, \mathit{abdoex}), \mathit{value\_interval}(P3,' [-1.663, 1.443]'),\\
\mathit{symbolic\_value}(P3,' \mathit{STEADY}') \qquad [\mathit{support} = 7.14\%]
\end{array}$$

This pattern demonstrates empirically that when the stage duration is higher, then the frequency of temporal pattern is lower. Indeed, a larger value of the minimal duration leads to the generation of wider time-windows and a numerous set of complex events, many of which are so different to reduce the frequency of patterns of events. This observation is also confirmed by the accuracy of the results (Table 2) of the method of event detection (subsection 3.3). Indeed, when the minimal duration is 120 secs (Table 2 right) the number of true positive events (sensitivity) decreases while the number of false positive events increases, and this leads to avoid that the true positive events contribute to form the final set of frequent patterns. True positive events are defined by asking domain experts to manually identify physiological parameters expected to be involved in known events.

Table 2. Accuracy of the event detection for minimal duration set to 60 secs (left) and 120 secs (right) and CS to 60

$[t_F..t_L]$width	sensitivity (%)	specificity (%)
10	71	44
15	68	46
20	64	43
25	70	48
30	71	48

$[t_F..t_L]$width	sensitivity (%)	specificity (%)
20	67	39
30	62	42
40	59	40
50	64	36
60	66	41

5 Conclusions

We investigated some issues raising when analyzing longitudinal data and proposed a combined approach driven by only data which does not (necessarily) rely on domain knowledge. Given the characteristic of longitudinal data to represent a dynamic process, the approach can have particular usefulness in the initial or preliminary investigations of the processes, as the experiments empirically prove. As future work we plan to explore the possibilities to integrate other forms of temporal data describing the same process into the several tasks of the framework.

Acknowledgment. This work is in partial fulfillment of the research objectives of the project ATENEO-2010: "Modelli e Metodi Computazionali per la Scoperta di Conoscenza in Dati Spazio-Temporali".

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM 26(11), 832–843 (1983)
2. Ceri, S., Gottlob, G., Tanca, L.: Logic Programming and Databases. Springer, Heidelberg (1990)
3. Chen, X., Petrounias, I.: A framework for temporal data mining. In: Quirchmayr, G., Bench-Capon, T.J.M., Schweighofer, E. (eds.) DEXA 1998. LNCS, vol. 1460, pp. 796–805. Springer, Heidelberg (1998)
4. Diday, E., Esposito, F.: An introduction to symbolic data analysis and the sodas software. Intell. Data Anal. 7(6), 583–601 (2003)
5. Loglisci, C., Berardi, M.: Segmentation of evolving complex data and generation of models. In: ICDM Workshops, pp. 269–273. IEEE Computer Society, Los Alamitos (2006)
6. Loglisci, C., Malerba, D.: Discovering triggering events from longitudinal data. In: ICDM Workshops, pp. 248–256. IEEE Computer Society Press, Los Alamitos (2008)
7. Malerba, D., Lisi, F.A.: An ILP method for spatial association rule mining. In: First Workshop on Multi-Relational Data Mining, pp. 18–29 (2001)
8. Mörchen, F.: Unsupervised pattern mining from symbolic temporal data. SIGKDD Explorations 9(1), 41–55 (2007)
9. Muggleton, S.: Inductive Logic Programming. Academic Press, London (1992)
10. Singer, J.D., Willet, J.B.: Applied longitudinal data analysis. Modelling change and event occurrence. Oxford University Press, Inc., Oxford (2003)