# Higher Order Temporal Rules*

Paul Cotofrei and Kilian Stoffel

University of Neuchatel,
Pierre-a-Mazel 7, 2000 Neuchatel, Switzerland
{paul.cotofrei,kilian.stoffel}@unine.ch

**Abstract.** The theoretical framework we proposed, based on first-order temporal logic, permits to define the main notions used in temporal data mining (event, temporal rule) in a formal way. The concept of *consistent linear time structure* allows us to introduce the notions of *general interpretation* and of *confidence*. These notions open the possibility to use statistical approaches in the design of algorithms for inferring higher order temporal rules, denoted temporal meta-rules.

## 1 Introduction

The domain of temporal data mining focuses on the discovery of causal relationships among events that may be ordered in time and may be causally related. The contributions in this domain encompass the discovery of temporal rule, of sequences and of patterns. However, in many respects this is just a terminological heterogeneity among researchers that are, nevertheless, addressing the same problem, albeit from different starting points and domains.

Although there is a rich bibliography concerning formalism for temporal databases, there are very few articles on this topic for temporal data mining. In [1,2,3] general frameworks for temporal mining are proposed, but usually the researches on causal and temporal rules are more concentrated on the methodological or algorithmic aspect, and less on the theoretical aspect. In this article, we start with an innovative formalism based on first-order temporal logic, which permits an abstract view on temporal rules. This formalism allows the application of an inference phase in which higher order temporal rules (denoted temporal meta-rules) are inferred from local temporal rules, the lasts being extracted from different sequences of data. Using this strategy, known in the literature as higher order mining [4], we can guarantee the scalability of our system (the capacity to handle huge databases), by applying standard statistical and machine learning tools. In the same time, the analysis of higher order temporal rules may put in evidence changes in extracted rules over time, that is, changes in the model of the data. The algorithms we proposed for extracting temporal meta-rules are not dependent on a particular knowledge discovery methodology as long as the local temporal rules, generated by this methodology, may be expressed in our formalism.

* Supported by the Swiss National Science Foundation (grant N ° 2100-063 730.

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 323–332, 2003.

The rest of the paper is structured as follows. In the next section, the first-order temporal logic formalism is extensively presented (definitions of the main terms – event, temporal rules, confidence – and concepts – consistent linear time structure, general interpretation). The notion of temporal meta-rules and the algorithms for inferring such high order rules are described in Section 3. Finally, the last section summarizes our work and lists some possible future directions.

## 2 The Formalism of Temporal Rules

Time is ubiquitous in information systems, but the mode of representation/perception varies in function of the purpose of the analysis [5,6]. Firstly, there is a choice of a *temporal ontology*, which can be based either on *time points* (instants) or on *intervals* (periods). Secondly, time may have a *discrete* or a *continuous* structure. Finally, there is a choice of *linear* vs. *nonlinear* time (e.g. acyclic graph). For our methodology, we chose a temporal domain represented by linearly ordered discrete instants.

**Definition 1.** *A single-dimensional linearly ordered temporal domain is a structure $T_P = (T, <)$, where $T$ is a set of time instants and "$<$" a linear order on $T$.*

A *first-order temporal language* L is constructed over an alphabet containing function symbols (including constants), predicate symbols, variables, logical connectives, temporal connectives and qualifier symbols. A *function symbol* (*predicate symbol)* is a lower (upper) case letter followed by a string of lower case letters and/or digits. A *constant* is a zero-ary function symbol and a zero-ary predicate is a *proposition symbol.* An upper case letter represents a variable. There are several special binary predicate symbols $\{=, <, \leq, >, \geq\}$ known as *relational symbols.* The basic set of logical connectives is $\{\wedge, \neg\}$ from which one may express $\vee$, $\rightarrow$ and $\leftrightarrow$. The basic temporal connectives are X (*next time*) and U (*until*) from which we may derive F (*sometime*) and G (*always*).

Consider now a restricted first-order temporal language L which contains only n-ary function symbols ($n \geq 0$), n-ary predicate symbols ($n > 1$, so no proposition symbols), the set of relational symbols $\{=, <, \leq, >, \geq\}$, a single logical connective $\{\wedge\}$ and a temporal connective of the form $X_k,\ k \in \mathbf{Z}$, where $k$ strictly positive means *next k times*, $k$ strictly negative means *last k times* and $k = 0$ means *now.*

The syntax of L defines terms, atomic formulae and compound formulae, which are defined inductively by the usual rules. A Horn clause is a formula of the form $B_1 \wedge \cdots \wedge B_m \rightarrow B_{m+1}$ where each $B_i$ is a positive (non-negated) atom. The atoms $B_i,\ i = 1, \ldots, m$ are called implication clauses, whereas $B_{m+1}$ is known as the implicated clause. Syntactically, we cannot express Horn clauses in our language L because the logical connective $\rightarrow$ is not defined. However, to allow the description of rules, which formally look like a Horn clause, we introduce a new logical connective, $\mapsto$, which practically will represent a rewrite of the connective $\wedge$. Therefore, a formula in L of the form $p \mapsto q$ is syntactically equivalent with the formula $p \wedge q$. When and under what conditions we may use the new connective, one precise in the next definitions.

**Definition 2.** *An event (or temporal atom) is an atom formed by the predicate symbol E followed by a bracketed n-tuple of terms* $(n \geq 1)$ $E(t_1, t_2, \ldots, t_n)$*. The first term of the tuple,* $t_1$*, is a constant representing the name of the event and all others terms are function symbols. A short temporal atom (or the event's head) is the atom* $E(t_1)$*.*

**Definition 3.** *A constraint formula for the event* $E(t_1, t_2, \ldots t_n)$ *is a conjunctive compound formula,* $C_1 \wedge C_2 \wedge \cdots \wedge C_k$*, where each* $C_j$ *is a relation implying one of the terms* $t_i$*.*

For a short temporal atom $E(t_1)$, the only constraint formula that is permitted, denoted *short constraint formula*, is $t_1 = c$, where $c$ is a constant.

**Definition 4.** *A temporal rule is a formula of the form* $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$*, where* $H_{m+1}$ *is a short constraint formula and the* $H_i$ *are constraint formulae, prefixed by the temporal connectives* $X_{-k}$*,* $k \geq 0$*. The maximum value of the index k is called the time window of the temporal rule.*

*Remark.* The reason for which we did not permit the expression of the implication connective in our language is related on the truth table for a formula $p \rightarrow q$: even if p is false, the formula is still true, which is unacceptable for a temporal rationing of the form *cause*$\rightarrow$ *effect*.

Practically, the only atoms constructed in L are temporal atoms and the only formulae constructed in L are constraint formulae and temporal rules. As a consequence of the Definition 4, a conjunction of relations $C_1 \wedge C_2 \wedge \cdots \wedge C_n$, each relation prefixed by temporal connectives $X_{-k}$, $k \geq 0$, may be rewritten as $C_{\sigma(1)} \wedge \cdots \wedge C_{\sigma(n-1)} \mapsto C_{\sigma(n)}$, – $\sigma$ being a permutation of $\{1..n\}$ – only if there is a short constraint formula $C_{\sigma(n)}$ prefixed by $X_0$.

The semantics of L is provided by an interpretation I over a domain D. The interpretation assigns an appropriate meaning over D to the (non-logical) symbols of L. Usually, the domain D is imposed during the discretisation phase, which is a pre-processing phase used in almost all knowledge extraction methodologies. Based on Definition 2, an event can be seen as a labelled (constant symbol $t_1$) sequence of points extracted from raw data and characterized by a finite set of features (function symbols $t_2, \cdots, t_n$). Consequently, the domain D is the union $D_e \cup D_f$, where the set $D_e$ contains all the strings used as event names and the set $D_f$ represents the union of all domains corresponding to chosen features.

To define a first-order linear temporal logic based on L, we need a structure having a temporal dimension and capable to capture the relationship between a time moment and the interpretation I at this moment.

**Definition 5.** *Given L and a domain D, a (first order) linear time structure is a triple* $M = (S, x, \Im)$*, where S is a set of states,* $x : \mathbf{N} \rightarrow S$ *is an infinite sequence of states* $(s_0, s_1, \ldots, s_n, \ldots)$ *and* $\Im$ *is a function that associates to each state s an interpretation* $\Im(s)$ *of all symbols defined at s.*

In the framework of temporal data mining, the function $\Im$ is a constant and it is equal to the interpretation I. In fact, the meaning of the events, constraint formulae and temporal rules is not changing over time. What is changing over time is the value of the meaning. Given a first order time structure M, we denote the instant $i$ (or equivalently, the state $s_i$) for which $I(P) = true$ by $i \Rightarrow P$, i.e. at time instant $i$ the formula P is true. Therefore, $i \Rightarrow E(t_1, \ldots, t_n)$ means that at time $i$ an event with the name $t_1$ and characterized by the global features $t_2, \ldots, t_n$ started. A constraint formula is true at time $i$ if and only if all relations are true at time $i$. A temporal rule is true at time $i$ if and only if $i \Rightarrow H_{m+1}$ and $i \Rightarrow (H_1 \wedge \cdots \wedge H_m)$. (*Remark*: $i \Rightarrow P \wedge Q$ if and only if $i \Rightarrow P$ and $i \Rightarrow Q$; $i \Rightarrow X_k P$ if and only if $i + k \Rightarrow P$).

Now suppose that the following assumptions are true:

A. For each formula $P$ in L, there is an algorithm that calculates the value of the interpretation $I(P)$ in a finite number of steps.
B. There are states (called incomplete states) that do not contain enough information to calculate the interpretation for all formulae defined at these states.
C. It is possible to establish a measure, (called *general interpretation*) about the degree of truth of a compound formula along the entire sequence of states $(s_0, s_1, \ldots, s_n, \ldots)$.

The first assumption express the calculability of the interpretation I. The second assumption express the situation when only the body of a temporal rule can be evaluated at time moment i, but not the head of the rule. Therefore, for the state $s_i$, we cannot calculate the interpretation of the temporal rule and the only solution is to estimate it using a general interpretation. This solution is expressed by the third assumption. (*Remark:* The second assumption violates the condition about the existence of an interpretation in each state $s_i$, from Definition 5. But it is well known that in data mining sometimes data are incomplete or are missing. Therefore, we must modify this condition as "*$\Im$ is a function that associates to* `almost` *each state s an interpretation $\Im(s)$ of all symbols defined at s*").

However, to ensure that this general interpretation is well defined, the linear time structure must present some property of consistency. Practically, this means that if we take any sufficiently large subset of time instants, the conclusions we may infer from this subset are sufficiently close from those inferred from the entire set of time instants. Therefore,

**Definition 6.** *Given L and a linear time structure M, we say that M is a consistent time structure for L if, for every n-ary predicate symbol P, the limit* $co(P) = \lim_{n \to \infty} \frac{\#A}{n}$ *exists, where* $A = \{i \in \{0, \ldots, n\} | i \Rightarrow P\}$ *and # means "cardinality". The notation co(P) denotes the confidence of P*

Now we define the general interpretation for an n-ary predicate symbol P as:

**Definition 7.** *Given L and a consistent linear time structure M for L, the general interpretation* $I_G$ *for an n-ary predicate P is a function* $D^n \to true \times [0, 1]$, $I_G(P) = (true, co(P))$.

The general interpretation is naturally extended to constraint formulae, prefixed or not by temporal connectives. There is only one exception: for temporal rules the confidence is calculated as a limit ratio between the number of certain applications (time instants where both the body and the head of the rule are true) and the number of potential applications (time instants where only the body of the rule is true). The reason for this choice is related to the presence of incomplete states, where the interpretation for the implicated clause cannot be calculated. A useful temporal rule is a rule with a confidence greater than 0.5.

**Definition 8.** *The confidence of a temporal rule $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ is the limit $\lim_{n\to\infty} \frac{\#A}{\#B}$, where $A = \{i \in \{0, \ldots, n\} | i \Rightarrow H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}$ and $B = \{i \in \{0, \ldots, n\} | i \Rightarrow H_1 \wedge \cdots \wedge H_m\}$.*

For different reasons, (the user has not access to the entire sequence of states, or the states he has access to are incomplete), the general interpretation cannot be calculated. A solution is to estimate $I_G$ using a finite linear time structure, i.e. a model.

**Definition 9.** *Given L and a consistent time structure $M = (S, x, \Im)$, a model for M is a structure $\tilde{M} = (\tilde{T}, \tilde{x})$ where $\tilde{T}$ is a finite temporal domain $\{i_1, \ldots, i_n\}$, $\tilde{x}$ is the subsequence of states $\{x_{i_1}, \ldots, x_{i_n}\}$ (the restriction of x to the temporal domain $\tilde{T}$) and for each $i_j, j = 1, \ldots, n$, the state $x_{i_j}$ is a complete state.*

Now we may define the estimations for the general interpretation and for the confidence of a temporal rule, giving a model :

**Definition 10.** *Given L and a model $\tilde{M}$ for M, an estimator of the general interpretation for an n-ary predicate P, $\tilde{I}_G(P)$, is a function $D^n \to true \times [0, 1]$, assigning to P the value* true *with a confidence equal to the ratio $\frac{\#A}{\#\tilde{T}}$, where $A = \{i \in \tilde{T} | i \Rightarrow P\}$. The notation $co(P, \tilde{M})$ will denote the estimated confidence of P, given $\tilde{M}$.*

**Definition 11.** *Given a model $\tilde{M} = (\tilde{T}, \tilde{x})$ for M, the estimation of the confidence of the temporal rule $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ is the ratio $\frac{\#A}{\#B}$, where $A = \{i \in \tilde{T} | i \Rightarrow H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}$ and $B = \{i \in \tilde{T} | i \Rightarrow H_1 \wedge \cdots \wedge H_m\}$.*

## 2.1 A General Methodology

A general methodology for temporal rules extraction may be structured in two phases. The first, called discretisation phase, transforms sequential raw data into sequences of events and establishes the set of temporal atoms that can be defined syntactically in L. In addition, during this phase, a linear time structure is defined: at each time moment i, the state contains as information the set of events started at i. The second phase, called inference phase, extract temporal rules from the set of all events. To guarantee the scalability of the methodology, this phase is divided in two steps:

A. application of a first induction process, using different models $\tilde{M}$ for $M$, to obtain different sets of temporal rules, and
B. application of a second inference process, using the previously inferred temporal rules, to obtain the final set of temporal meta-rules.

Among different approaches that can be applied to extract rules from a set of events - Association Rules[7], Inductive Logic Programming[8], Classification Trees[9] - we proposed (see [10,11]) the classification tree approach. Consequently, the first induction process consists in creating multiple classification trees, each based on a different training set. Choosing a training set is equivalent to choose a model. All the states from these models are complete states, because the algorithm that construct the tree must know, for each time moment, the set of predictor events and the corresponding dependent event.

Once the classification tree constructed, the outcome of the test contained in each node becomes a relation and the set of all relations situated on a path from root to a leaf becomes a constraint formula. This constraint formula becomes a temporal rule by adding temporal connectives . The confidence of temporal rule is calculated according to the Definition 11. (*Remark*: the classification tree approach guarantees the extraction of useful temporal rules from a given model, but do not guarantee the extraction of all useful temporal rules from this model). The second inference process is designed to obtain temporal meta-rules, which are temporal rules in accordance with the Definition 4, but supposed to have a small variability of the estimated confidence among different models. Therefore, a temporal meta-rule may be applied with the same confidence in any state, complete or incomplete. The process of inferring temporal meta-rules is related to a new approach in data mining, called the higher order mining, i.e. mining from the results of previous mining runs. According to this approach, the rules generated by the first induction process are first order rules and those generated by the second inference process (i.e. temporal meta-rules) are higher order rules. The formalism we proposed does not impose what methodology to use to discover first order temporal rules. As long as these rules may be expressed according to the Definition 4, the strategy (including algorithms, criterions, statistical methods) developed to infer temporal meta-rules might be applied.

## 3 Temporal Meta-rules

Suppose that for a given model $\tilde{M}$ we dispose of a set of temporal rules, extracted from the corresponding classification tree. It is very likely that some temporal rules contain constraint formulae that are irrelevant, i.e. by deleting these relations, the general interpretation of the rules remain unchanged. In the frame of a consistent time structure $M$, it is obviously that we cannot delete a relation from a temporal rule (noted $TR$) if the resulting temporal rule (noted $TR^-$) has a general interpretation with a lower confidence. But for a given model $\tilde{M}$, we obtain an estimate of $co(TR)$, which is $co(TR, \tilde{M})$. This estimator having a binomial distribution, we can calculate a confidence interval for $co(TR)$ and, consequently, we accept to delete a relation from $TR$ if and only if the lower confidence limit of $co(TR^-, \tilde{M})$ is greater than the lower confidence limit of $co(TR, \tilde{M})$.

The estimator $co(TR, \tilde{M})$ being a ratio, $\#A/\#B$, a confidence interval for this value is constructed using a normal distribution depending on $\#A$ and $\#B$ (more precisely, the normal distribution has mean $\pi = \#A/\#B$ and variance $\sigma^2 = \pi(1-\pi)/\#B$). The lower limit of the interval is $L_\alpha(A, B) = \pi - z_\alpha \sigma$, where $z_\alpha$ is a quantile of the normal distribution for a given confidence level $\alpha$. The algorithm which generalize a single temporal rule TR, by deleting one relation, is presented in the following:

**Algorithm 1** ***Generalization 1-delete***

**Step 1.** *Let* $TR = H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$. *Let* $\aleph = \cup C_j$, *where* $C_j$ *are all relations that appear in the constraint formulae of the implication clauses. Rewrite TR, by an abuse of notation, as* $\aleph \mapsto H_{m+1}$. *If* $n = \#\aleph$, *denote by* $C_1, \ldots, C_n$ *the list of all relations from* $\aleph$.

**Step 2.** *For each* $i = 1, \ldots, n$ *do*

$\aleph^- = \aleph - C_i, \quad TR_i^- = \aleph^- \mapsto H_{m+1}$
$A = \{i \in \tilde{T} | i \Rightarrow \aleph \wedge H_{m+1}\}, B = \{i \in \tilde{T} | i \Rightarrow \aleph\}$
$A^- = \{i \in \tilde{T} | i \Rightarrow \aleph^- \wedge H_{m+1}\}, B^- = \{i \in \tilde{T} | i \Rightarrow \aleph^-\}$
$co(TR, \tilde{M}) = \#A/\#B, \ co(TR_i^-, \tilde{M}) = \#A^-/\#B^-$
*If* $L_\alpha(A, B) \leq L_\alpha(A^-, B^-)$ *then store* $TR_i^-$

**Step 3.** *Keep only the generalized temporal rule* $TR_i^-$ *for which* $L_\alpha(A^-, B^-)$ *is minimal.*

The core of the algorithm is the **Step 2**, where the sets used to estimate the confidence of the initial temporal rule, $TR$, and of the generalized temporal rule, $TR^-$, i.e. $A, B, A^-$ and $B^-$, are calculated. The complexity of this algorithm is linear in $n$. Using the criterion of lower confidence limit, (or LCL), we define the temporal meta-rule inferred from $TR$ as the temporal rule with a maximum set of relations deleted from $\aleph$ and having the minimum lower confidence limit greater than $L_\alpha(A, B)$. An algorithm designed to find the largest subset of relation that can be deleted will have an exponential complexity. A possible solution is to use the Algorithm 1 in successive steps until no more deletion is possible, but without having the guarantee that we will get the global minimum.

Suppose now that we dispose of two models, $\tilde{M}_1 = (\tilde{T}_1, \tilde{x}_1)$ and $\tilde{M}_2 = (\tilde{T}_2, \tilde{x}_2)$, and for each model we have a set of temporal rules with the same implicated clause $H$ (sets denoted $S_1$, respectively $S_2$). Let S be a subset of the reunion $S_1 \cup S_2$. If $TR_j \in S$, $j = 1, \ldots, n$, $TR_j = H_1 \wedge \cdots \wedge H_{m_j} \mapsto H$, then denote

$$A_j = \{i \in \tilde{T}_1 \cup \tilde{T}_2 | i \Rightarrow H_1 \wedge \ldots \wedge H_{m_j} \wedge H\}, \mathbf{A} = \cup A_j,$$
$$B_j = \{i \in \tilde{T}_1 \cup \tilde{T}_2 | i \Rightarrow H_1 \wedge \ldots \wedge H_{m_j}\}, \mathbf{B} = \cup B_j,$$
$$\mathbf{C} = \{i \in \tilde{T}_1 \cup \tilde{T}_2 | i \Rightarrow H\}.$$

The performance of the subset S can be summarized by the number of *false positives* (time instants where the implication clauses of each temporal rule from S are true, but not the clause H) and the number of *false negatives* (time instants where the clause H is true, but not at least one of the implication clauses of the temporal rules from S). Practically, the number of *false positives* is $fp = \#(\mathbf{B} - \mathbf{A})$ and the number of *false negatives* is $fn = \#(\mathbf{C} - \mathbf{B})$. The worth of the subset S of temporal rules is assessed using the Minimum Description Length Principle (MDLP). This provides a basis for offsetting the accuracy of a theory (here, a subset of temporal rules) against its complexity. The principle is simple: a Sender and a Receiver have both the same models $\tilde{M}_1$ and $\tilde{M}_2$, but the states from the models of the Receiver are incomplete states (the interpretation of the implicated clause cannot be calculated). The sender must communicate the missing information to the Receiver by transmitting a theory together with the exceptions to this theory. He may choose either a simple theory with a great number of exceptions or a complex theory with fewer exceptions. The MLDP states that the best theory will minimize the number of bits required to encode the total message consisting of the theory together with its associated exceptions.

To encode a temporal rule from S, we must specify its implication clauses (the implicated clause being the same for all rules, there is not need to encoded it). Because the order of the implication clauses is not important, the number of required bits is divided by $\kappa \log_2(m!)$, where $m$ is the number of implication clauses and $\kappa$ is a constant depending on encoding procedure. The number of bits required to encode the set S is the sum of encoding length for each temporal rule from S divided by $\kappa \log_2(n!)$ (the order of the $n$ temporal rules from S is not important). The exceptions are encoded by indicating the sets *false positive* and *false negative*. If $b = \#\mathbf{B}$ and $N = \#(\tilde{T}_1 + \tilde{T}_2)$ then the number of bits required is $\kappa \log_2\left(\binom{b}{fp}\right) + \kappa \log_2\left(\binom{N-b}{fn}\right)$, because we have $\binom{b}{fp}$ possibilities to choose the *false positives* among the cases covered by the rules and $\binom{N-b}{fn}$ possibilities to indicate the *false negatives* among the uncovered cases. The total number of bits required to encode the message is then equal to *theory bits + exceptions bits.*

Using the criterion of MDLP, we define as temporal meta-rules inferred from a set of temporal rules (implying the same clause and extracted from at least two different models), the subset S that minimizes the total encoding length. The algorithm that find this subset S has the same complexity as the algorithm which find the largest subset of relations to be deleted, (so exponential), but in practice we may use different non-optimal strategies (hill-climbing, genetic algorithms, simulated annealing), having a polynomial complexity.

Because the two definitions of temporal meta-rules differ not only in criterion (LCL, respectively MLDP), but also in the number of initial models (one, respectively at least two), the second inference process is applied in two steps. During the first step, temporal meta-rules are inferred from each set of temporal rules based on a single model. During the second step, temporal meta-rules are inferred from each set of temporal rules created during the step one and having the same implicated clause (see Fig. 1). There is another reason to apply firstly the LCL criterion: the resulted temporal meta-rules are less redundant concerning the set of implication clauses and so the encoding procedures, used by MLDP criterion, don't need an adjustment against this effect.

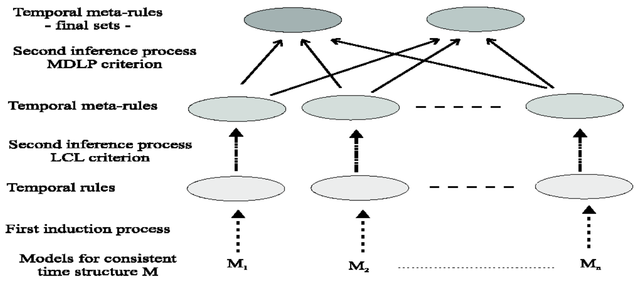


**Fig. 1.** Graphical representation of the second inference process

## 4 Conclusions

The theoretical framework we proposed, based on first-order temporal logic, permits to define the main notions (event, temporal rule, constraint) in a formal way. The notion of the consistent linear time structure allows us to introduce the notions of general interpretation and of confidence. These notions open the possibility to use probabilistic concepts and allow, at the same time, to formalize an inference process in which temporal meta-rules are derived from locally temporal rules. This process is related to a new research area for data mining, the higher order mining, which opens new perspectives on the analysis of mining results and their evolution. The algorithms we proposed for inferring higher order temporal rules are based on two different criterion, Lower Confidence Limit and Minimum Description Length Principle. In both cases, the complexity of the optimal solution is exponential and so,in practice, it is recommended to use non-optimal, but polynomial, strategies.

It is important to mention that the condition of the existence of the limit, in the definition of consistent linear time structure, is a fundamental one: it express the fact that the linear time structure M represents a homogenous model and therefore the conclusions (or inferences) based on a finite model for M are consistent. However, at this moment, we do not know methods which may certified that a given temporal structure is consistent. In our opinion, the only feasible approach to this problem is the development of methods and procedure for detecting the change points in the model, and, in this direction, the analysis of temporal meta-rules seems a very promising starting point.

## References

1. Al-Naemi, S.: A Theoretical Framework for Temporal Knowledge Discovery. In: Proceedings of International Workshop on Spatio-Temporal Databases, Spain (1994) 23–33
2. Chen, X., Petrounias, I.: A Framework for Temporal Data Mining. In: Proceedings of the 9th International Conference on Database and Expert System Applications, LNCS 1460 (1998) 796–805
3. Malerba, D., Esposito, F., Lisi, F.: A Logical Framework for Frequent Pattern Discovery in Spatial Data. In: Proceedings of the 5th International Conference KDD, San Diego, USA (2001) 53–62
4. Spiliopoulou, M., Roddick, J.: Higher Order Mining: Modeling and Mining the Results of Knowledge Discovery. In: Proceedings of the 2nd International Conference on Data Mining Methods and Databases, UK (2000) 309–320
5. Chomicki, J., Toman, D.: Temporal Logic in Information Systems. BRICS Lecture Series (1997)
6. Emerson, E.A.: Temporal and Modal Logic. Handbook of Theoretical Computer Science (1990) 995–1072
7. Chen, X., Petrounias, I.: Discovering Temporal Association Rules: Algorithms, Language and System. In: Proceedings of the 6th International Conference on Data Engineering, San Diego, USA (2000) 306
8. Rodriguez, J., Alonso, C., Boström, H.: Learning First Order Logic Time Series Classi.ers: Rules and Boosting. In: Proceedings of the $10^t h$ International Conference on Inductive Logic Programming. (2000) 260–275
9. Karimi, K., Hamilton, H.: Finding Temporal Relations: Causal Bayesian Networks vs. C4.5. In: Proceedings of the 12th International Symposium on Methodologies for Intelligent Systems, Charlotte, USA (2000)
10. Cotofrei, P., Stoffel, K.: Classi.cation Rules + Times = Temporal Rules. In: Proceedings of the 2nd International Conference on Computational Science, Amsterdam. LNCS 2329 (2002) 572–581
11. Cotofrei, P., Stoffel, K.: A formalism for Temporal Rules. In: Proceedings of the ACM/SIGKDD Workshop on Temporal Data Mining, Edmonton, Canada (2002) 25–37