New Approach on Temporal Data Mining for Symbolic Time Sequences: Temporal Tree Associate Rules

Mathieu Guillame-Bert
INRIA Rhône-Alpes Research Center
655 Av. de l'Europe, 38330 Montbonnot-St.-Martin
France
Email: mathieu.guillame-bert@inrialpes.fr

James L. Crowley
INRIA Rhône-Alpes Research Center
655 Av. de l'Europe, 38330 Montbonnot-St.-Martin
France
Email: james.crowley@inrialpes.fr

***Abstract*—We introduce a temporal pattern model called Temporal Tree Associative Rule (TTA rule). This pattern model can be used to express both uncertainty and temporal inaccuracy of temporal events expressed as Symbolic Time Sequences. Among other things, TTA rules can express the usual time point operators, synchronicity, order, chaining, as well as temporal negation. TTA rules is designed to allows predictions with optimum temporal precision. Using this representation, we present an algorithm that can be used to extract Temporal Tree Associative rules from large data sets of symbolic time sequences. This algorithm is a mining heuristic based on entropy maximisation and statistical independence analysis. We discuss the evaluation of probabilistic temporal rules, evaluate our technique with an experiment and discuss the results.**

***Keywords*-data mining; temporal patterns; knowledge discovery; augmented environments;**

I. Introduction

Temporal data mining is concerned with the extraction of temporal patterns from large data sets. The usefulness of temporal data mining continues to grow as increasing amounts of temporal data about everyday activities become available. Three classes of temporal patterns exist:

- Temporal Association Patterns represent associative relations between time sampled observations. For example, '*if the door bell rings, somebody will probably enter the house in the following minutes*'.
- Temporally Constrained (temporal or not temporal) Association Patterns represent the evolution of a pattern over time. For example, '*Between 7pm and 10pm, people who buy diapers in super-markets, also buy beer with a 50% chance*'.
- Temporally Evolving Association Patterns represent the evolution of a pattern over time. For example, '*If I leave my office now, I will be home in t minutes, with t depending of the time of the day (because of traffic)*'.

Temporal data mining of Temporal Association Patterns has been successfully applied in numerous domains including medical, trading, robotics, social analysis, fraud detection, marketing and assisted design [1]–[4]. Because of the variety of different domains and types of applications, several temporal data models have proven to be useful (numeric time series, symbolic time series, symbolic time sequences, symbolic interval series, item set sequences, etc.) [5]–[8]. Similarly, different types of pattern models have been studied (sequences of events, sequences rules, temporal rules, first order logic based rules, chronicles, etc.) [5]–[9]. For each of these data and pattern models, efficient algorithms have been proposed. The suitability of an algorithm depends on the domain and type of application.

In this paper we focus on symbolic time sequences (sets of time sampled symbolic events). We present a temporal rule model called Temporal Tree Associative Rule (TTA rules). This rule model can express both uncertainty and temporal inaccuracy of temporal events. TTA rules can express the usual time point operators, synchronicity, order, chaining and temporal negation. We give several examples of TTA rules to demonstrate the expressive power of this representation.

We present an algorithm able to extract Temporal Tree Associative rules from symbolic time series. The selection of a good time intervals is a recurrent problem across the different techniques of symbolic time sequence mining. The algorithm is based on a statistical independence analysis heuristic to deal with this problem. It is able to deal with noisy and dense data.

The underlying idea of the algorithm is the following one: The algorithm begins by mining simple and trivial TTA rules. Next, it enriches rules with new temporal conditions according to an entropy minimization policy.

The algorithm is evaluated on a real world dataset. Real world datasets evaluation supports the concrete usefulness of the technique.

Evaluation on large computer generated dataset has also been done but the results are not presented in this paper. It allows precise understanding of the algorithm's heuristic. The computer generated dataset and evaluating tool are available online [10].

The next section compares our technique to related work. The third section presents the notation used in this article. The fourth section defines the TTA rule representation. Simple examples are presented. The fifth section gives and explains the algorithm. The sixth section shows and discusses the results of the application of the algorithm on a test case. The last section discusses several aspects of the algorithms.

II. Related work

Algorithms have been developed to extract a variety of different temporal models [11]–[16]. Mined temporal patterns are generally used for prediction (guessing future or non observable events) or classification (for verification or discriminative pattern). In the first case, the precision of the time of occurrence of a prediction's occurrence is an important criterion.

In the current literature, most temporal patterns dealing with symbolic time sequences generally give predictions with non optimum precision: Episode patterns [17] produce predictions with a precision, equal or lower to the size of the time windows used for time discretization. Partial order patterns [18] propose a partial improvement by adding an information or order in the pattern. Ghallab developed the idea of temporal constraint networks called Chronicles [9]. Dousson et al. [15] developed an on-line algorithm able to extract chronicles. This algorithm, called FACE, is based on an heuristic for the determination of time constraints. As far as we know, Chronicles are the only patterns with the ability to express a simple rule like 'if A appends at time t, B will appends at time t + 1 hour $\pm$ 2 minutes'. We believe such rules are relevant for numerous domains such as human activity analysis or market analysis. TTA rules can express this kind of temporal relation and add an exact probability distribution information.By opposition to related approaches [8], [16], the algorithm can deal with patterns with low confidence, low support, and with noisy and dense time sequences.

TTA rules can express the usual time point relation 'before/after', the notion of synchronicity, order and chaining, with different level of flexibility. TTA rules can also express Negation such as e.g. 'there are not occurrences of events of type A during a given interval' or 'there are not chains of A followed by B followed by C'.

III. Notations and events representation

Definition 1: A *probability distribution* describes the probability of each value (or interval of values) of a random variable.

Definition 2: A *(temporal) event* e is a symbol (called type and noted symbol_e) and a time of occurrence (time_e). The writing convention is $e := \mathrm{symbol}_e\,[\mathrm{time}_e]$. Literally an event e expresses that an event of symbol symbol_e appends at a time time_e.

Example 1: Suppose $e := A\,[7.5]$ to be an event. Literally e_1 means that an event of symbol A occurs at time 7.5.

Definition 3: A *state* s is a function $\mathbb{R} \to \{0,1\}$ that maps a value for every time location (i.e. real number). If $s(t) = 1$, s is said to be true at time t. Otherwise, s is said to be false at time t.

IV. Temporal Tree Associative Rules

We are now defining a Temporal Tree Associative Rule. Several graphical examples of rules are given at the end of this section.

Definition 4: A *time mask* m is a function $\mathbb{R} \to \{0,1\}$;If $m(t) = 1$, m is said to be true at time t. Otherwise, m is said to be false at time t. Given two time masks m and m', the relation $m \leq m'$ is true if and only if there is no $x \in \mathbb{R}$ such as $m(x) = 1$ and $m'(x) = 0$. We define the time mask $T_{a,b}$ such as:

$$T_{a,b} : x \mapsto \begin{cases} 1 & \text{if } x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$$

Definition 5: A *type 1 condition* c is a symbol (symbol_c) and a set of type 2 conditions (conds_c). The writing convention is $c := \langle \mathrm{symbol}_c, \mathrm{conds}_c \rangle$.

For a set of events E, a temporal condition c is true at time t if:

- E contains an event e of symbol symbol_c and time t i.e. $\mathrm{symbol}_e = \mathrm{symbol}_c$ and $\mathrm{time}_e = t$.
- All type 2 conditions $c' \in \mathrm{conds}_c$ are true at time t (see definition bellow).

Definition 6: A *type 2 condition* c is either:

- The negation of a type 2 condition c_2 (written $c := \mathrm{not}\, c_2$). Here, c is true at time t if and only if c_2 is false at time t.
- The reading of a state s (written $c := s$). Here, c is true at time t if and only if s is true at time t i.e. $s(t) = 1$.
- An association between a time mask m and a type 1 condition c_3 (written $c := [m, c_3]$). Here, c is true at time t if and only if $\exists t'$ with $m(t' - t) = 1$ and c_3 is true at time t'.

Definition 7: A *Temporal Tree Associative Rule* (TTA rule) r is a symbol (symbol_r), a confidence (conf_r), a non null probability distribution (dist_r) and a type 1 condition (cond_r). $\mathrm{dist}_r(t - t')$ is the probability density of having an event of symbol symbol_r at time t while the condition cond_r being true at time t'. The writing convention is $r := \mathrm{cond}_r \Rightarrow \mathrm{head}_r\, \langle \mathrm{conf}_r, \mathrm{dist}_r \rangle$.

When the condition cond_r of a rule r is true at time t, r is said to do a prediction of an event of symbol head_r with a probability of conf_r and with a temporal location of $t + \mathrm{dist}_r$. An event e of symbol $\mathrm{symbol}_e = \mathrm{head}_r$ is said to verify such prediction if the density of the prediction is not equal to zero at time time_e i.e. $f'(\mathrm{time}_e) > 0$ with $f' := t + \mathrm{dist}_r$.

Definition 8: The *support* of the rule r is the percentage of events with type head_r explained by the rule.

Definition 9: The *standard deviation* std_r of a rule r is the standard deviation of its probability distribution dist_r. The *precision* of a rule r is defined as $\frac{1}{\mathrm{std}_r}$.

Definition 10: A *unit trivial rule* is a rule with the pattern $\langle x, \emptyset \rangle \Rightarrow y \langle 100\%, \mathbb{U}_{-\infty,+\infty} \rangle$. This type of rule is called trivial because as soon as there is at lease one occurrence of x and one occurrence of y, this rule's confidence and support are 100%.

We are now presenting four examples of TTA rules in order to show their power of expression. We specify a graphical representation of the rule in order to help the understanding of the rules in fig. 1.

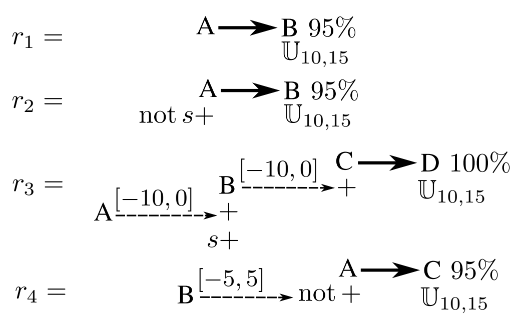


Figure 1. Graphical representation of rules

The following examples shows a unit rule, a state negation rule, a chain of condition rule and a negation of event rule.

Example 2: Suppose the rule $r_1 := \langle A, \emptyset \rangle \Rightarrow B \langle 95\%, \mathbb{U}_{10,15} \rangle$. r_1 expresses that if an event of type A occurs at time t, then, an event of type B will occur between $t + 10$ and $t + 15$ with 95% chance.

Example 3: Suppose the rule $r_2 := \langle A, \{\mathrm{not}\, s\} \rangle \Rightarrow B \langle 95\%, \mathbb{U}_{10,15} \rangle$. r_2 expresses that if an event of type A occurs at time t and the state s is not true at time t, then, an event of type B will occur between $t + 10$ and $t + 15$ with 95% chance.

Example 4: Suppose the rule $r_3 := \langle C, \{[T_{-10,0}, \langle B, \{s, [T_{-10,0}, \{\langle A, \emptyset \rangle\}]\} \rangle]\} \rangle \Rightarrow D \langle 100\%, \mathbb{U}_{10,15} \rangle$. r_3 expresses that if an event of type C occurs at time t_c followed by an event of type B at time t_b (with a maximum interval of 10 seconds i.e $t_b - 10 \leq t_c \leq t_b - 0$) followed by an event of type A at time t_a (with a maximum interval of 10 seconds i.e $t_a - 10 \leq t_b \leq t_a - 0$) and s is true at time t_b, then, an event of type D will occur between $t_a + 10$ and $t_a + 15$ with 100% chance.

Example 5: Suppose $r_4 := \langle A, \{\mathrm{not}\, [T_{-5,5}, \langle B, \emptyset \rangle]\} \rangle \Rightarrow C \langle 95\%, \mathbb{U}_{10,15} \rangle$. Literally, r_4 expresses that if an event of type A occurs at time t and no events of type B occur between $t - 5$ and $t + 5$ i.e. there are no events of type B around the event of type A, then, an event of type C will occur between $t + 10$ and $t + 15$ with 95% chance.

V. Algorithms

In this section we present the algorithm. It relies on three simple operations:

1) **Creation of trivial unit rules**
Given two symbols A and B, a trivial unit rules is $\langle A, \emptyset \rangle \Rightarrow B \langle 100\%, \mathbb{U}_{-\infty,+\infty} \rangle$.
2) **Addition of a condition to a rule**
We present this operation through an example. Due to limitations of space the formal definition and properties are no given. Fig. 2 represents the addition of the condition $[T_{-10,0}, \langle C, \emptyset \rangle]$ to the rule $r_5 := \langle A, \emptyset \rangle \Rightarrow B \langle 95\%, \mathbb{U}_{10,15} \rangle$ to a given location. The result is $r_6 := \langle A, \{[T_{-10,0}, \langle C, \emptyset \rangle]\} \rangle \Rightarrow B \langle 95\%, \mathbb{U}_{10,15} \rangle$.

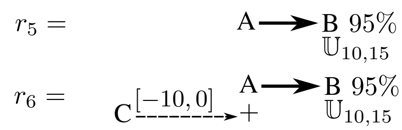


Figure 2. Addition of condition

3) **Division of a rule**
We present this operation through an example. Due to limitations of space the formal definition and properties are no given. Fig. 3 represents a division of r_7 into $\{r_8, r_9\}$ with around $t = 5$.

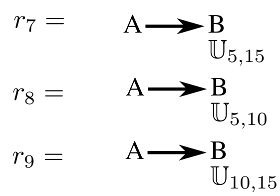


Figure 3. Division of a rule

A. The algorithm

Parameters of the algorithm are: **minConfidence** as the minimum confidence of the rules to generate. **minSupport** as the minimum support of the rules to generate. **maxLoop** as the maximum number of loops in the first step of the algorithm. Other criteria have been tried (maximum number of rules, maximum computation times, etc.). For space reason, these criteria are not described in this paper. **minEntropyGain** as the minimum gain of entropy when adding a condition in a rule. This parameter is used on the heuristic of condition adding. **minProbabilitiesDependency** is a real. It expresses a minimum probability of dependency i.e. co-occurrence. This parameter is used on the heuristic of rule division.

1) Heuristic 1: Given a set of rule R, the heuristic 1 selects a rule $r \in R$ and a condition to add to r in order to improve it. The value of improvement is given by the entropy gain of addition of condition. 90% of the time, the heuristic selects the rule and condition that maximise the entropy gain.

Algorithm 1: E is the set of events, S is the set of states

begin
- ▷ initialisation
- **compute** Sym the set of all symbols
- **compute** R the set of all trivial unit rules with support greater than **minSupport**
- i.e. $R :=$ $\{\langle w, \emptyset\rangle \Rightarrow x \langle 100\%, \mathbb{U}_{-\infty,+\infty}\rangle \,|\forall w, x \in Sym\}$
- ▷ addition of condition
- **while** *stopping criteria is not meet* **do**
  - ▷ addition of condition
  - **select** a rule r, a type 1 condition c_1 of r and a condition c_2 with the **heuristic 1** (see details bellow)
  - **add** the condition c_2 to r
  - i.e. replace c_1 of r by $\langle \mathrm{symbol}_{c_1}, \mathrm{conds}_{c_1} \cup \{c_2\}\rangle$
  - **if** $\mathrm{supp}\, r \geq$ ***minSupport*** **then**
    - **evaluate** the density histogram of r
    - **put** r in R
    - ▷ division
    - **generate** with the **heuristic 2** a division function d for r
    - **divide** r according to d and store the result in $\{r_i\}$
    - **for** $r' \in \{r_i\}$ **do**
      - **if** $\mathrm{supp}\, r' \geq$ ***minSupport*** **then**
        - **put** r' in R
- **return** R

10% of the time, the heuristic selects a random rule and a random condition. This policy prevents the algorithm to fall in local minima.

2) Heuristic 2: The selection of a good division of rule is more complex than the addition of condition. The goal of a rule division is to produce more temporally precise rules (decrees of rule standard deviation) while loosing as less as possible confidence and support. Knowing that a rule's prediction can match several events, the heuristic 2 divides a rule into a sets of independent rules based on graphs colorations techniques. Due to limitation of space, the idea behind this heuristic is given through an example.

Example 6: Suppose two events symbols with the following property: If there is an event A at time t, then there is an event B at time $t + 5$ and an event B at time $t + 15$. Additionally, other events B are uniformly present in the dataset. Suppose the rule $r_1 := \langle A, \emptyset\rangle \Rightarrow B \langle 100\%, \mathbb{U}_{0,20}\rangle$ Fig. 4 represents the histogram of the distribution $P(t' - t''|A[t] \text{ and } B[t])$, the co-occurrence matrix of this distribution, the graph with the vertices coloration, and the colored distribution. Fig. 5 represents the rules r_1, and the divided rules r_2 and r_3.

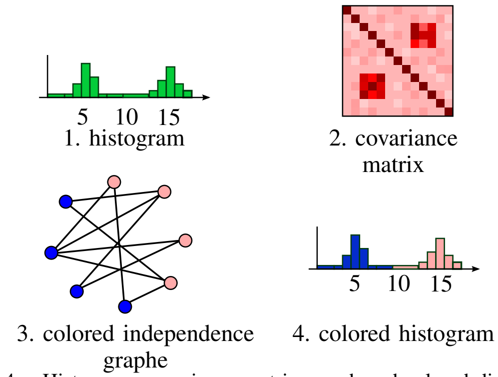


Figure 4. Histogram, covariance matrix, graph and colored distribution

$$r_1 = \mathrm{A} \xrightarrow[\mathbb{U}_{0,20}]{} \mathrm{B}$$

$$r_2 = \mathrm{A} \xrightarrow[\mathbb{U}_{0,10}]{} \mathrm{B}$$

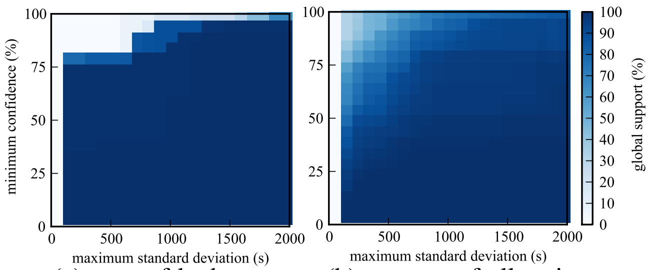
$$r_3 = \mathrm{A} \xrightarrow[\mathbb{U}_{10,20}]{} \mathrm{B}$$

Figure 5. Result of the heuristic 2

VI. Results

We present an evaluation of our algorithm on a real world data set. The 'Home activities dataset' created by Tim van Kasteren et al. [19] is a record of 28 days of sensors data and activity annotations about one person performing activities at home. Activities of the person are annotated (prepare breakfast, dinner, having a drink, toileting, sleeping, leaving the house, etc.). The events data set is divided into two categories: sensors events (start_sensor_fridge, end_sensor_fridge, start_sensor_frontdoor, etc.) and change of activities (start_action_get_drink, end_action_get_drink, start_action_prepare_dinner, etc.). S is composed of 24 states describing the time of the day (it_is_1am, it_is_2am, it_is_3am, etc.). In this experiment, the algorithm is applied in order to predict activity change events according to sensors events and states describing the time. E contains 42 types of events and 2904 occurrences of events. The algorithm is executed for 60 seconds. The confidence, support and standard deviation of rules depend of the kind of activity to predict. While actions sleeping or using bathroom are fairly well explained (high confidence and high support of rules), actions such as getting drink or leaving the house can't be explained with high confidence, high support and low standard deviation at the same time. Fig. VI shows the percentage of event explained (global support) according to minimum confidence and maximum temporal precision.

As an example, fig. VI tells that we can predict all the uses of bathroom with at least 75% confidence and a temporal precision of less than 100 seconds, or with 100% confidence and a temporal precision of less that 800 seconds.



(a) uses of bathroom (b) average of all actions

The algorithm extracts rules of different complexity. Direct correlations between events (unit rules) often have good support but average confidence. For example, the direct implication between the use of the toilet flush and the action of using toilets (confidence : 87%, support : 86% standard deviation:227 seconds), or the use of the front door and the action of leaving the flat (confidence:51%, support:100% standard deviation:194 seconds).

More complex rules are based on several conditions. As an example, the action of leaving bed is predicted with a 91% confidence 100% of the time by the rule represented on fig. 6. This rule is, informally, that if the hall bedroom door is opened without been or having being closed in a range of $\pm$ 17 minutes, there is 91% chance that the person is waking up. The prediction is done with a standard deviation of 25 seconds.

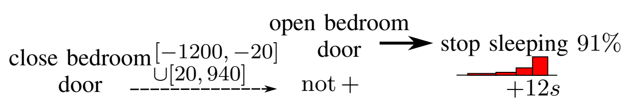


Figure 6. rule of wake up prediction

Finally, rules composed of "chain" of actions often have very high confidence but low support. For example, a chain of events including the use of the toilets, the freezer and the microwave indicates the action of preparing the breakfast with a very high confidence (100%), a low standard deviation (98 seconds), but a low support (21%).

VII. Discussion

The number of Temporal Tree Associative Rules grows exponentially with the depth of rules and the number of histogram categories. In usual data sets, the number of good rules (rules that satisfy the confidence, support and standard deviation requirements) is often also very high. The algorithm we developed is a greedy best-first search heuristic that only generates the most interesting rules (rules with the best confidence, support and standard deviation). At every step, the algorithm picks up the rule with the best immediate improvement with a given percentage of chance. This heuristic does not provide guaranty of finding the best rules but gives good results in usual data sets. The selection of a good time intervals is a recurrent problem across the different techniques of symbolic time sequence mining. We propose a solution based on a statistical independence analysis able to deal with noisy and dense data.

References

[1] G. Das, K. ip Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series," 1998.

[2] G. Guimares, "The induction of temporal grammatical rules from multivariate time series," 2000.

[3] E. J. Keogh and P. Smyth, "A probabilistic approach to fast pattern matching in time series databases," 1997.

[4] H. Mannila and H. Toivonen, "Discovering generalized episodes using minimal occurrences," 1996.

[5] J. Roddick and M. Spiliopoulou, "A survey of temporal knowledge discovery paradigms and methods," 2002.

[6] F. Mörchen, "Unsupervised pattern mining from symbolic temporal data," 2007.

[7] C. M. Antunes and A. L. Oliveira, "Temporal data mining: An overview."

[8] F. Fessant, F. Clrot, and C. Dousson, "Mining of an alarm log to improve the discovery of frequent patterns." pp. 144–152, 2004.

[9] M. Ghallab, "On chronicles: Representation, on-line recognition and learning." 1996.

[10] G.-B. Mathieu, "Symbolic time sequence dataset," 2011, "available at http://www-prima.imag.fr/guillame-bert/?page=database".

[11] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, "Sequential pattern mining using a bitmap representation," 2002.

[12] M. J. Zaki, "Efficient enumeration of frequent sequences," 1998.

[13] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," 1996.

[14] J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M. chun Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," 2001.

[15] C. Dousson and T. V. Duong, "Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems," 1999.

[16] H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," 1997.

[17] R. Agrawal and R. Srikant, "Mining sequential patterns," 1995.

[18] G. Casas-Garriga, "Summarizing sequential data with closed partial orders," 2005.

[19] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," 2008.