Towards a General Method for Logical Rule Extraction from Time Series

Guido Sciavicco[1][0000-0002-9221-879X],
Ionel Eduard Stan[2][0000-0001-9260-102X], and
Alessandro Vaccari[1][0000-0001-9135-9155]

[1] Dept. of Mathematics and Computer Science,
University of Ferrara (Italy)
guido.sciavicco@unife.it|alessandr.vaccari@student.unife.it
[2] Dept. of Mathematics, Computer Science, and Physics
University of Udine (Italy)
stan.ioneleduard@spes.uniud.it

Abstract. Extracting rules from temporal series is a well-established temporal data mining technique. The current literature contains a number of different algorithms and experiments that allow one to abstract temporal series and, later, extract meaningful rules from them. In this paper, we approach this problem in a rather general way, without resorting, as many other methods, to expert knowledge and ad-hoc solutions. Our very simple temporal abstraction method allows us to transform time series into timelines, which can be then used for logical temporal rule extraction using an already existing temporal adaptation of the algorithm APRIORI. We have tested this approach on real data, obtaining promising results.

Keywords: Rule extraction; Time series; Timelines.

1 Introduction

Rule-based methods are a popular class of techniques in machine learning and data mining [8]. They share the goal of finding regularities in data that can be expressed in the form of *if-then* rules. Depending on the type of rules, we can discriminate between *descriptive rules* discovery, which aims at describing significant patterns in the given data set in terms of rules, and *predictive rules* discovery, which is focused on learning a collection of the rules that collectively cover the instance space and can make a prediction for every possible instance. In this paper, we are interested in descriptive (or *association*) rules, for which there are three popular approaches: *inductive logic programming*, which uses logic programming as a uniform representation for examples, background knowledge and hypotheses, and aims at deriving a hypothesised logic program (that is, a set of rules) which entails all the positive and none of the negative examples (see, e.g., [18, 19]); *rule induction via metaheuristics*, typically driven by evolutionary algorithms; and APRIORI [1] and its subsequent developments. These approaches have been extensively compared in the literature (see, e.g., [9] and references therein); apparently, although APRIORI is probably the first technology for rule extraction that gained some acknowledgment in the community, its main ideas are still widely used, since no negative examples are needed (in contrast to inductive logic programming), and since it is considered reliable and fast (in contrast to metaheuristic approaches, which are computationally expensive).

Time series are largely used to describe a wide range of data. Their use is ubiquitous; among others, they are commonly used in *environmental sciences* [22], where data describing the quality of air, water, soil, food most often include multi-variate time series, in *industry* [21], where time series are used to describe the working parameters of machines, in *medical sciences* [14], where complex medical exams are normally described by multi-variate series, and in *smart homes* [13], where sensors that help the intelligent systems in taking decisions usually generate data in form of series. As a consequence, extracting rules from time series can be very important, and the literature on this topic is relatively large, ranging from primitive approaches [6, 7], to generalizations of APRIORI to take into account the temporal component [2], to more general and modern methodologies [20]. A survey on temporal abstraction methods is given by Höppner in [12]. There exist several abstraction methods, which can be roughly separated into *adimensional*, that is, methods that do not consider the temporal dimension, and *dimensional*, which do consider the temporal dimension. Examples of the first category include: *(i) Equal Width Discretization* (EWD), *(ii) Equal Frequency Discretization* (EFD), and *(iii) k-means clustering*. Examples of dimensional methods, include: *(iv) Symbolic Aggregate approXimation* (SAX) [15], which does not explicitly consider the temporal order of the values and *(v) Persist* [16], which maximizes the duration of the resulting time intervals and which explicitly considers the temporal order. Moreover, in [17], a classification-driven discretization method, namely *Temporal Discretization for Classification* (TD4C), is presented. A comparative study on various *pattern languages* (i.e. approaches to represent the temporal interval patterns) is presented in [11], where the authors point out the strengths and weaknesses for each of them, based on four well-known problems in the literature: preservation of qualitative relationship, preservation of quantitative durations, concurrency and robustness; many of such languages are Allen-based, but none of them have a logical approach.

Upon examining the current literature on temporal series extraction, one can see that most of the proposed solutions share common characteristics: *(i)* temporal abstraction, that is, the segmentation and (possibly) aggregation of time series into symbolic time intervals is often based on external knowledge and/or specific to the rule extraction algorithm; *(ii)* rule extraction algorithm are mostly ad-hoc, and *(iii)* the extracted rules are usually existential, often limited to being binary rules, in which the temporal (Allen's) relations between intervals are used in a very limited way. In this paper, we present a general technique, structured into the two, classical steps of temporal abstraction and rule extraction, that addresses the above issues. Our temporal abstraction step is completely general

HS	**Allen's relations**	**Graphical representation**
$\langle A \rangle$	$[x,y]R_A[x',y'] \Leftrightarrow y = x'$	
$\langle L \rangle$	$[x,y]R_L[x',y'] \Leftrightarrow y < x'$	
$\langle B \rangle$	$[x,y]R_B[x',y'] \Leftrightarrow x = x', y' < y$	
$\langle E \rangle$	$[x,y]R_E[x',y'] \Leftrightarrow y = y', x < x'$	
$\langle D \rangle$	$[x,y]R_D[x',y'] \Leftrightarrow x < x', y' < y$	
$\langle O \rangle$	$[x,y]R_O[x',y'] \Leftrightarrow x < x' < y < y'$	

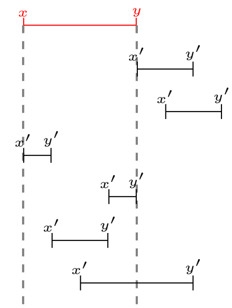

Fig. 1. Allen's interval relations and HS modalities.

and domain-independent, and separated from the rule extraction phase, which, in turn, is based on a general tool known as Temporal APRIORI [3]. The main characteristics of our approach are: *(i)* rules are written in a well-known logical language, that is, Halpern and Shoham's interval-based temporal logic HS [10], *(ii)* every Allen's relation may have a role in the extracted rules, and the (sub)set of relations that are actually used is completely customizable, and *(iii)* rules are logic formulas, that is, they generalize classical, static, APRIORI rules. Temporal APRIORI is a rule extraction algorithm to extract interval temporal logic rules from timelines; the cornerstone of this method is, therefore, consistently transforming multi-variate time series into timelines.

2 Temporal APRIORI

Temporal APRIORI is a rule extraction algorithm that generalizes APRIORI to cope with instances with a temporal component, and it has been presented in [3]. The starting point of Temporal APRIORI is the observation that non-temporal rules can be thought as implications written in propositional logic, even though they are not interpreted as implications in strict logical terms. On the one hand, rules represent positive information only: instances where the implication is trivially satisfied by the absence of the antecedent are not relevant in this setting. On the other hand, rules express a likelihood information, such as *if these items are present, it is very likely that this other item will be present, too*, rather than a deterministic Boolean value. Classical static rules (such as rules extracted from frequent item sets, to use APRIORI terminology) have the form of propositional Horn logic rules:

$$\rho : p_1 \wedge p_2 \wedge \ldots \wedge p_k \Rightarrow p \tag{1}$$

where $p_1, \ldots, p_k, p$ are propositional letters associated to the items of the instances in the data set (*literals*). Since in many application domains temporal information is stored in form of intervals, extracting interval-based temporal rules is the natural generalization of the above idea. The most representative (and general) interval temporal language is Halpern and Shoham's Modal Logic of Time Intervals [10], often referred to as HS, and it is a modal propositional language that features precisely one existential modal operator and one universal modal operator for each basic relation between two intervals. Its sub-Boolean (Horn-like) fragment, originally studied in [5] naturally generalizes the classical propositional Horn logic, and, at the same time, it has some decidable and tractable sub-fragments [4]. The Horn fragment of HS has a simple grammar. First, we define *temporal literals*:

$$\lambda ::= \top \mid \bot \mid p \mid \langle X \rangle \lambda \mid [X]\lambda \mid \langle \overline{X} \rangle \lambda \mid [\overline{X}]\lambda,$$

where $\langle X \rangle$ (resp., $\langle \overline{X} \rangle$) is a modal operator that existentially ranges over the Allen's relation R_X, that is, $X \in \{A, B, E, D, O, L\}$ (resp., the inverse of Allen's relation R_X), $[X]$ (resp., $[\overline{X}]$) is its universal version, and p is a propositional letter, and then, we define *rules*:

$$\varphi ::= \lambda \mid [G](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda) \mid \varphi_1 \wedge \varphi_2.$$

The semantics of Horn HS formulas is given in terms of *interval models* (or *timelines*) of the type $T = \langle D, V \rangle$, where $(D, \leq)$ is a linearly ordered set and $V : \mathcal{AP} \rightarrow 2^{I(D)}$ is a *valuation function* which assigns to each atomic proposition $p \in \mathcal{AP}$ the set of intervals $V(p)$ on which p holds, being $I(D)$ the set of all intervals (that is, pairs of the type $[x, y]$, where $x < y$) that can be formed on D. The *truth* of a formula φ on a given interval $[x, y]$ in a timeline T is defined by structural induction on formulas as follows:

- $T, [x, y] \Vdash \top$ and $T, [x, y] \nVdash \bot$ for every $[x, y] \in I(D)$;
- $T, [x, y] \Vdash p$ if $[x, y] \in V(p)$;
- $T, [x, y] \Vdash \langle X \rangle \psi$ if there is a $[w, z]$ such that $[x, y] R_X [w, z]$ and $T, [w, z] \Vdash \psi$;
- $T, [x, y] \Vdash [X]\psi$ if, for all $[w, z]$ such that $[x, y] R_X [w, z]$, $T, [w, z] \Vdash \psi$;
- $T, [x, y] \Vdash [G](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda)$ if, for all $[w, z]$ such that $T, [w, z] \Vdash \lambda_1 \wedge \cdots \wedge \lambda_k$, $T, [w, z] \Vdash \lambda$;
- $T, [x, y] \Vdash \psi_1 \wedge \psi_2$ if $T, [x, y] \Vdash \psi_1$ and $T, [x, y] \Vdash \psi_2$.

Timelines generalize static instances. Consider, for example, the medical history of a patient. While during a interesting period of observation, we may *statically* describe the set of its symptoms, the (suitably discretized) values of his/her tests, and the therapies to which he/she has undergone, and extract static rules in the form of (1), if we take into account the temporal component, we may, instead, describe the same medical histories by associating every interesting information to the temporal interval in which it holds. The rules that may be extracted take, therefore, the form:

$$\rho : \lambda_1 \wedge \lambda_2 \wedge \ldots \wedge \lambda_k \Rightarrow \lambda, \tag{2}$$

where $\lambda_1, \ldots, \lambda_k, \lambda$ are temporal literal. As we have already recalled, rules are not implications in the strict logical sense. As much as static rules are concerned, the validity of a rule such as (1) in a static data set depends on two parameters, known as support and confidence: the *support* of a rule is the minimum fraction of instances of the data set in which every one satisfies both the antecedent $(p_1 \wedge p_2 \wedge \ldots \wedge p_k)$ and the consequent (p) should hold, and the *confidence* is the minimum fraction of the support of the antecedent only in which both the antecedent and the consequent should hold in order to consider a rule such as (1) as true on a data set. So, for example, we say that $p_1 \wedge p_2 \Rightarrow p$ *holds on a data set with support* 0.75 *and confidence* 0.95 if at least three quarters of the instances satisfy p_1, p_2 and p and, of those instances which satisfy p_1 and p_2, ninety-five percent satisfy also p. Temporal rules require a similar treatment, with the additional problem that instances are not static, and thus evaluating a rule on single instance is not immediate. The concepts of support and confidence naturally generalize along two directions. First, we use the *temporal support* and *temporal confidence* to evaluate a single temporal literal λ on a single timeline T: the former establishes which is the minimal fraction of all intervals of T that must be captured by the relation R_X over the interval $[x, y]$ in order to the temporal literal λ to make sense on it, and the latter establishes the minimal fraction of those that must satisfy the argument of λ in order for λ to be considered true. In this, way, for example, we do not evaluate $[D]p$ on a too short interval, or $[A]p$ on an interval too close to the rightmost point of a timeline, and we do evaluate as true $[L]p$ on an interval $[x, y]$ when the number of intervals of the type $[z, t]$ $(z > y)$ with $\neg p$ are less than a certain fraction. Observe that temporal support and confidence may depend on the specific relation R_X and on the fact that λ is existential or universal. Second, we use the *global support* and *global confidence* to set, respectively, the minimum fractions of intervals of T in which both the antecedent and the consequent of a rule such as (2) should hold and the minimum fraction of the support of the antecedent only in which also the consequent should hold in order to evaluate as true the entire rule. Finally, to evaluate a rule on a temporal data set, one applies a generalized version of the the standard support, defined as the minimum fraction of timelines in which at least one intervals satisfies both the antecedent and the consequent of the body of the rule (so that the rule is significant in the temporal data set).

A prototypal implementation of Temporal APRIORI is described in [3] that implements most of the above concepts, and has been used to run the experiments described in this paper.

3 Temporal Abstraction of Time Series

As we have already recalled, temporal abstraction has been widely studied in the recent literature. Our proposal is driven by two objectives: first, generalizing and systematizing the other approaches with a simple, domain-independent algorithm, and, second, transforming time series into timelines from which rules such as those described in the previous section can be extracted.

Let $\bar{F}(t) = (f_1(t), \ldots, f_m(t))$ be a multi-variate time series. Each $f_j(t)$ is referred to as *variable* (or *attribute*, to use the standard terminology for static data). A data set of multi-variate time series has the form $\bar{F}_1(t), \bar{F}_2(t), \ldots, \bar{F}_n(t)$. Although time series that describe real-life data may have any codomain (typically, the reals), since a data set is always finite and extensively described we can always assume that each $f_j(t)$ is a function of the type:

$$f_j : D \rightarrow \mathbb{N},$$

where D is a finite temporal domain. As for example, in a medical domain temporal data set, each instance $\bar{F}(t) = (f_1(t), f_2(t))$ may be the description of the medical history of a patient that includes his/her *fever* ($f_1(t)$) and his/her level of *blood pressure* ($f_2(t)$). Our purpose is to convert $\bar{F}(t)$ into a timeline T in which every interval is suitably labelled to carry the same information (in abstract form). Time series are often abstracted in different ways with different aims; in some cases, an interval is labelled with the *state* of some variable $f_j(t)$, that is, with the average value of $f_j(t)$ in that intervals; in some other cases, a label represents the *trend* of some variable. In order to generalize and systematize such a labelling process, we introduce the concept of *z-th degree of timeline*, in analogy with the z-th degree of *discrete derivative*. As a matter of fact, states are simply averages of the values of some $f_j(t)$, while trends are averages of the values of $f_j^1(t)$. In general, therefore, one may be interested to abstract a time series at any degree of derivative, to obtain a timeline from which rules can be extracted. In the following, we use the symbol $\bar{F}^z(t)$ for $(f_1^z(t), f_2^z(t), \ldots, f_m^z(t))$.

Since at each degree of derivative the finite domain of the resulting function contains one less point, we denote by D^z the domains obtained from D at the z-th degree of derivative. Fixed a degree z, the abstraction process consists of producing a timeline $T_{\bar{F}^z(t)}$ from $\bar{F}^z(t)$:

$$T_{\bar{F}^z(t)} = \langle D^z, V \rangle,$$

and we have to specify the valuation function V. To this end, we first consider the mean (denoted by μ_j) and the standard deviation (denoted by σ_j) of the j-th component entire series (at the z-th derivative), and, for a specific interval $[x, y] \in I(D)$, we define:

$$\mu_j^{xy} = \frac{\Sigma_{x \leq t \leq y} f_j^z(t)}{y - x},$$

that is, the mean of the values of $f_j^z(t)$ between x and y, and we use them to build a set of propositional letters to define the valuation function V. In classical solutions for temporal abstraction labels are often domain-dependent. In order to avoid the use of domain-related knowledge, we introduce two parameters, that is, $l > 1, l \in \mathbb{N}$ (*number of labels*, assumed to be odd) and $k \in [0, 1] \subset \mathbb{R}$ (*displacement*), and define the set of propositional letters:

$$\{L_p^j \mid 1 \leq p \leq l, 1 \leq j \leq m\},$$

and, finally, define:

```
proc Abstract (F, z, l, k)
  T = ∅
  for (i = 1 to n)
    T_i = Abs(F̄_i(t), z, l, k)
    T = T ∪ {T_i}
  return T
```

Fig. 2. A general, domain-independent, temporal abstraction algorithm.

$$[x,y] \in V(L_p^j) \text{ iff } \begin{cases} \mu_j^{xy} < \mu_j - \lfloor \frac{l}{2} \rfloor k\sigma_j & \text{if } p = 1 \\ \mu_j - (\lceil \frac{l}{2} \rceil - p + 1)k\sigma_j \leq \mu_j^{xy} < (\lceil \frac{l}{2} \rceil - p)k\sigma_j & \text{if } 1 < p < \lceil \frac{l}{2} \rceil \\ \mu_j - k\sigma_j \leq \mu_j^{xy} \leq \mu_j + k\sigma_j & \text{if } p = \lceil \frac{l}{2} \rceil \\ \mu_j - (p - \lceil \frac{l}{2} \rceil)k\sigma_j < \mu_j^{xy} \leq (p - \lceil \frac{l}{2} \rceil + 1)k\sigma_j & \text{if } \lceil \frac{l}{2} \rceil < p < l \\ \mu_j^{xy} > \mu_j + \lfloor \frac{l}{2} \rfloor k\sigma_j & \text{if } p = l \end{cases}$$

So, for example, if $z = 0$, $l = 3$ and $k = 0.5$, then L_1 (resp., L_2, L_3) can be read as *low* (resp., *average*, *high*), and an interval $[x, y]$ is labelled with *low* if its mean value is less than the mean value of the entire series (on the same component) minus half of its standard deviation. As another example, if $z = 1$, $l = 3$, and $k = 0.25$, then an interval $[x, y]$ is labelled with *increasing* (corresponding to L_3 on the first derivative, that is, the series of the trends) if the mean value of the differences in $[x, y]$ exceeds the mean value of all differences plus one fourth of the standard deviations of all differences. In this way, we can temporally abstract any multi-variate time series at any level of derivative, so that rules can be discovered that link the states, or the trends, or the accelerations, and so on, in a consistent, simple, and general way.

Given a time series $\bar{F}(t)$, we say that the *abstracted z-th degree timeline* $T_{\bar{F}^z(t)}$, with l labels and displacement k is:

$$T_{\bar{F}^z(t)} = Abs(\bar{F}(t), z, l, k),$$

where Abs is a procedure that applies the above labelling strategy. Given a set of n time series $\mathcal{F} = (\bar{F}_1(t), \bar{F}_2(t), \ldots, \bar{F}_n(t))$, we convert it into a temporal data set (a set of n timelines) $\mathcal{T} = (T_1, T_2, \ldots, T_n)$ by simply applying the procedure *Abstract* in Fig. 2.

4 Application Example

In this example we use a set of time series that emerges from collecting physical-chemical data from underground water of a very specific area in the North-East of Italy. Such samples were collected as a part of a ongoing investigation commissioned by the local Regional Agency for Environment and Prevention to the University of Ferrara, with the purpose of exploring the causes of a sudden, unexpected spike of certain polluting agents in the underground water. Such

$\langle A \rangle$ $C.E.$ is average $\Rightarrow \langle L \rangle$ Na is high
$\langle A \rangle$ Cl is average $\Rightarrow \langle L \rangle$ Na is high
$\langle A \rangle$ Na is low $\Rightarrow \langle L \rangle$ $C.E.$ is average
$\langle L \rangle$ $C.E.$ is high $\Rightarrow \langle L \rangle$ Na is high
$C.E.$ is average $\wedge$ $HCO3$ is average $\Rightarrow \langle L \rangle Na$ is high

Table 1. Examples of rules extracted from the temporal data set at the 0-th degree of derivative.

$\langle A \rangle$ Cl decreaes $\Rightarrow \langle A \rangle$ $C.E.$ is stable
$\langle A \rangle$ Cl decreaes $\Rightarrow \langle A \rangle$ I is stable
$\langle D \rangle$ $C.E.$ is stable $\wedge \langle A \rangle$ I is stable $\Rightarrow [O]$ Cl does not increase
$\langle D \rangle$ $C.E.$ is stable $\wedge [A]$ I is not stable $\Rightarrow [O]$ Cl does not decrease

Table 2. Examples of rules extracted from the temporal data set at the 1-th degree of derivative.

data are being used to perform several physical-chemical researches; since they have the form of multi-variate time series, we can also use them to test our rule extraction method.

In the relevant area, 92 sampling points (underground water wells) were chosen for this analysis. Samples have been collected from 2012 to 2018, in a periodic way, on most of such points. Each sample has been analyzed from the physical-chemical point of view, and several indicators have been registered: Br (*Bromine*), Ca (*Calcium*), Cl (*Chlorine*), Fe (*Iron*), $HCO3$ (*Bicarbonate*), I (*Iodine*), K (*Potassium*), Mg (*Magnesium*), $NH4$ (*Ammonium cation*), $NO3$ (*Nitrate*), Na (*Sodium*), $SO4$ (*Sulfate*), Hg (*Mercury*), T (*Temperature*), Eh (*Reduction potential*), DO (*Chemical oxygen demand*), and $C.E.$ (*Electric conductivity*). After re-normalization, such data have been temporally ordered, obtaining 92 17-variate time series. These have been abstracted using the algorithm explained in the previous section, at the 0-th, and the 1-st degree, with 3 labels (per variable), and $k = 0.5$, to obtain two temporal data sets. Of the 92 series, only 43 meaningful timelines could be extracted: the remaining ones where too short (they have less than 3 observations). Moreover, for this particular exercise, the function *Abs* has been implemented in such a way that intervals that contain too long gaps (more than 150 consecutive days without observations) have not been labelled. The following parameters have been set for both experiments: minimum support 0.8, minimum confidence 0.85, minimum global confidence 0.85. For a better understanding of the underlying problem, the three labels have been paired with the labels corresponding to their negation, during the abstraction process. So, for example, for the 1-st degree of derivative and the value of Magnesium, we have used the letters *decreasing, stable, increasing* (corresponding to the three possible values of the derivative) and *not decreasing, not stable, not increasing.*

As it happens with classical, static APRIORI, a rule extraction generally produces many results. We have limited ourselves to analyze rules with unary or binary antecedent, and modal depth 1. Also, in this particular context, at the 0-th derivative degree rules that relate in time average situations with average situation are probably meaningless, as well as, at the 1-th degree, are rules that relate stable situations with stable situations. Examples of extracted rules are in Tab. 1 and Tab. 2. The first two rules predict a future period of high Sodium in the sample, provided that the observation is made immediately before a period in which conductivity or Chlorine is stable. By means of the the fourth rule we are able to foresee that a future period of high conductivity will be associated with a future period of high Sodium. The last two rules of the second group, corresponding to the 1-st degree of derivative, are particularly interesting. The first one allows us to say that if we are in a period in which, at some point, the conductivity is stable, and right before a period in which the Iodine level is also stable, then, the level of Chlorine will be stable or it will decrease for a while. But the second one says that if we are in a period in which, at some point, the conductivity is stable (as before), and never in future the Iodine level will be stable, then, the level of Chlorine will be stable or it will increase for a while.

5 Conclusions

It is well-recognized that extracting rules from time series is an important task. In this paper we approached the problem in a general way, and with a novel technique. We made use of a temporal logic language with a very high expressive power (at least at the qualitative level), and we have designed a temporal abstraction algorithm that transforms time series into timelines, so that temporal rules can be extracted with an already existing temporal generalization of APRIORI.

Acknowledgements. G. Sciavicco acknowledges the partial support by the Italian INDAM GNCS project *Formal methods for techniques for combined verification.*

References

1. Agrawal, R., Imieliński, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Proc. of the 1993 ACM SIGMOD International Conference on Management of Data. pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11th International Conference on Data Engineering. pp. 3–14 (1995)
3. Bresolin, D., Cominato, E., Gnani, S., Muñoz-Velasco, E., Sciavicco, G.: Extracting interval temporal logic rules: A first approach. In: Proc. of the 25th International Symposium on Temporal Representation and Reasoning. Leibniz International Proceedings in Informatics, vol. 120, pp. 7:1–7:15 (2018)
4. Bresolin, D., Kurucz, A., Muñoz-Velasco, E., Ryzhikov, V., Sciavicco, G., Zakharyaschev, M.: Horn fragments of the Halpern-Shoham interval temporal logic. ACM Transactions on Computational Logic **18**(3), 22:1–22:39 (2017)

5. Bresolin, D., Muñoz-Velasco, E., Sciavicco, G.: Sub-propositional fragments of the interval temporal logic of Allen's relations. In: Proc. of the 14th European Conference on Logics in Artificial Intelligence. Lecture Notes in Computer Science, vol. 8761, pp. 122–136 (2014)
6. Cotofrei, P., Stoffel, K.: Rule extraction from time series databases using classification trees. In: Proc. of the IASTED International Conference on Applied Informatics. pp. 327–332 (2002)
7. Das, G., Lin, K., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: Proc. of the 4th International Conference on Knowledge Discovery and Data Mining. pp. 16–22 (1998)
8. Fürnkranz, J., Gamberger, D., Lavrac, N.: Foundations of Rule Learning. Cognitive Technologies, Springer (2012)
9. Fürnkranz, J., Kliegr, T.: A brief overview of rule learning. In: Proc. of the 9th International Symposium Rule Technologies: Foundations, Tools, and Applications. pp. 54–69 (2015)
10. Halpern, J., Shoham, Y.: A propositional modal logic of time intervals. Journal of the ACM **38**(4), 935–962 (1991)
11. Höppner, F., Peter, S.: Temporal interval pattern languages to characterize time flow. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **4**(3), 196–212 (2014)
12. Höppner, F.: Time series abstraction methods - A survey. In: Proc. of the 32nd Annual Meeting of the Society for Informatics. pp. 777–786 (2002)
13. Jakkula, V., Cook, D., Jain, G.: Prediction models for a smart home based health care system. In: Proc. of the 21st International Conference on Advanced Information Networking and Applications. pp. 761–765 (2007)
14. Kurbalija, V., Radovanović, M., Ivanović, M., Schmidt, D., von Trzebiatowski, G.L., Burkhard, H., Hinrichs, C.: Time-series analysis in the medical domain: A study of tacrolimus administration and influence on kidney graft function. Computers in Biology and Medicine **50**, 19 – 31 (2014)
15. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Mining and Knowledge Discovery **15**(2), 107–144 (2007)
16. Mörchen, F., Ultsch, A.: Optimizing time series discretization for knowledge discovery. In: Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 660–665. ACM (2005)
17. Moskovitch, R., Shahar, Y.: Classification-driven temporal discretization of multivariate time series. Data Mining and Knowledge. Discovery **29**(4), 871–913 (2015)
18. Muggleton, S.: Inductive logic programming: Issues, results and the challenge of learning language in logic. Artificial Intelligence **114**(1-2), 283–296 (1999)
19. Raedt, L.D.: Logical and relational learning. Cognitive Technologies, Springer (2008)
20. Sacchi, L., Larizza, C., Combi, C., Bellazzi, R.: Data mining with temporal abstractions: learning rules from time series. Data Mining and Knowledge Discovery **15**(2), 217–247 (2007)
21. Sheremetov, L., González-Sánchez, A., López-Yáñez, I., Ponomarev, A.: Time series forecasting: Applications to the upstream oil and gas supply chain. In: Proc. of the 7th IFAC Conference on Manufacturing Modelling, Management, and Control. IFAC Proceedings Volumes, vol. 46, pp. 957 – 962 (2013)
22. Vito, S.D., Piga, M., Martinotto, L., Francia, G.D.: CO, NO2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization. Sensors and Actuators B: Chemical **143**, 182–191 (01 2009)