Extracting Interval Temporal Logic Rules: A First Approach

Davide Bresolin
Department of Mathematics
University of Padova, Italy
davide.bresolin@unipd.it
https://orcid.org/0000-0003-2253-9878

Enrico Cominato
Department of Mathematics, Computer Science and Physics
University of Udine, Italy
enrico.cominato@gmail.com

Simone Gnani
Department of Mathematics and Computer Science
University of Ferrara, Italy
simone.gnani@student.unife.it

Emilio Muñoz-Velasco
Department of Applied Mathematics
University of Malaga, Spain
ejmunoz@uma.es
https://orcid.org/0000-0002-0117-4219

Guido Sciavicco
Department of Mathematics and Computer Science
University of Ferrara, Italy
guido.sciavicco@unife.it
https://orcid.org/0000-0002-9221-879X

Abstract

Discovering association rules is a classical data mining task with a wide range of applications that include the medical, the financial, and the planning domains, among others. Modern rule extraction algorithms focus on *static* rules, typically expressed in the language of Horn propositional logic, as opposed to *temporal* ones, which have received less attention in the literature. Since in many application domains temporal information is stored in form of intervals, extracting interval-based temporal rules seems the natural choice. In this paper we extend the well-known algorithm APRIORI for rule extraction to discover interval temporal rules written in the Horn fragment of Halpern and Shoham's interval temporal logic.

2012 ACM Subject Classification Information systems → Clustering and classification

Keywords and phrases Interval temporal logic, Horn fragment, Rule extraction

Digital Object Identifier 10.4230/LIPIcs.TIME.2018.7

Acknowledgements D. Bresolin and G. Sciavicco acknowledge partial support by the Italian INDAM GNCS project *Formal methods for verification and synthesis of discrete and hybrid systems* (2018).

25th International Symposium on Temporal Representation and Reasoning (TIME 2018).
Editors: Natasha Alechina, Kjetil Nørvåg, and Wojciech Penczek; Article No. 7; pp. 7:1–7:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Rule-based methods are a popular class of techniques in machine learning and data mining [16]. 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. The first one is based on *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., [29, 32]). A second typical approach is based on *rule induction via metaheuristics*, typically driven by evolutionary algorithms (see [24] for an example); in this case, not only static propositional rules may be derived, but also, and even more often, *fuzzy* rules (see [21], and references within). The third classical rule extraction approach is based on APRIORI [2] and its subsequent developments. These approaches have been extensively compared in the literature (see, e.g., [17] 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).

If-then rules are extracted from an abstract data base, in which every transaction (or instance), is characterized by a set of items or features that represents the attributes of the instance. If-then rules are usually represented as logical rules, 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, as in APRIORI) are *propositional* Horn logic rules:

$$\rho : p_1 \wedge p_2 \wedge \ldots \wedge p_k \Rightarrow p$$

where $p_1, \ldots, p_k, p$ are propositional letters associated to the items of the instances in the data set. Other approaches make use of more complex languages. In [14, 15] first-order point-based *temporal* logics are used to describe temporal classification rules. In [3] a methodology to mine temporal sequential patterns is proposed. In this work patterns are based on instantaneous events and are purely existential; moreover, the algorithm can detect only one temporal relation, that is, *sometime in the future* (F, in the point-based temporal logic notation). Since in many application domains temporal information is stored in form of intervals, extracting interval-based temporal rules seems the natural step forward. The most representative interval temporal language is probably Halpern and Shoham's Modal Logic of Time Intervals [19], often referred to as HS, which is a modal propositional language that features precisely one existential modal operator and one universal modal operator for each basic relation between two intervals. Motivated by the search of computationally affordable versions of HS, several fragments have been explored, that include fragments with restricted sets of modal operators [28, 10, 1], fragments with softer (reflexive) semantics [27], fragments

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

Graphical representation

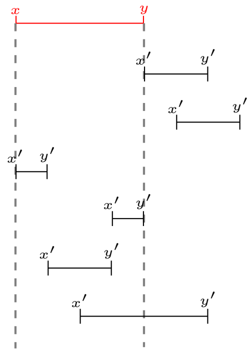

Figure 1 Allen's interval relations and HS modalities.

in which the nesting of modal operators is limited [8], and sub-propositional (Horn-like) fragments [9]. In particular, the Horn fragment of HS presents some advantages that are relevant to this study:

(i) it naturally generalizes the classical propositional Horn logic, and
(ii) some of its interesting sub-fragments are decidable and tractable [9, 11].

In this paper, we start from a set of finite *timelines* (that is, a *temporal data set*) $\mathcal{T} = \{T_1, \ldots, T_n\}$, where each timeline is a model for HS and can be interpreted as a temporal *history*. For example, in the medical domain, a temporal history T_i may be the set of all relevant events for a certain patient i (e.g., patient i has undergone a certain therapy, or has shown a certain combination of symptoms, in the interval $[x, y]$). As another example, in text mining, a temporal history T_i may represent the sequence of contexts of the conversation i. We propose a temporal extension of the algorithm APRIORI that extracts meaningful association rules from a temporal data set, written in the language of Horn HS [9, 11]. We use classical and well-recognized metrics for static rule evaluation (that is, support and confidence) in association with suitable new metrics for temporal rule evaluation, and we test our algorithm against a synthetic temporal data set.

This paper is organized as follows. In the next section we give the necessary preliminaries, while in Section 3 we briefly discuss some possible examples of applications. In Section 4 we discuss the problem of evaluating a rule and describe our algorithm. Then, in Section 5 we consider a synthetic temporal data set and show how our algorithm extracts useful rules from it, before concluding.

2 Preliminaries

Let $(D, \leq)$ be a linearly ordered set. A (strict) *interval* over D is an ordered pair $[x, y]$, where $x, y \in D$ and $x < y$. We denote by $I(D)$ the set of all (strict) intervals over D. If we exclude the identity relation, there are 12 different relations between two intervals in a linearly ordered set, often called *Allen's relations* [4]: the six relations R_A (adjacent to), R_L (later than), R_B (begins), R_E (ends), R_D (during), and R_O (overlaps), depicted in Fig. 1, and their inverses, that is, $R_{\overline{X}} = (R_X)^{-1}$, for each $X \in \mathcal{A} = \{A, L, B, E, D, O\}$.

Halpern and Shoham's Modal Logic of Time Intervals [19] (HS) is a modal logic that features a universal modality $[X]$ and an existential modalitiy $\langle X \rangle$ for each Allen relation R_X. For each $X \in \mathcal{A}$, the *transposes* of the modalities $[X]$ and $\langle X \rangle$ are respectively the modalities $[\overline{X}]$ and $\langle\overline{X}\rangle$, corresponding to the inverse relation $R_{\overline{X}}$ of R_X, and vice versa. Motivated by the search of computationally well-behaved versions and fragments of HS, several sub-propositional fragments of HS have been introduced and studied (see [9], and references therein). In this paper we focus our attention on the *Horn fragment* of HS. The basic blocks of the language are (*positive temporal*) *literals* λ, defined by the following grammar:

$$\lambda \ ::= \ \top \ \mid \ \perp \ \mid \ p \ \mid \ \langle X\rangle\lambda \ \mid \ [X]\lambda, \tag{1}$$

where R_X is one of the interval relations and p is a *propositional letter* from a finite, non-empty set $\mathcal{AP}$. Literals of the type $\langle X\rangle p$ (resp., $[X]p$) are called *existential* (resp., *universal*) literals. Formulas of the Horn fragment are in *clausal* form:

$$\varphi \ ::= \ \lambda \ \mid \ [G](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda) \ \mid \ \varphi_1 \wedge \varphi_2, \tag{2}$$

where $[G]$ is the *global operator* (which imposes that something is true everywhere in the model, and can be defined within the language HS), and each λ is obtained from (1). The conjuncts of the form λ are called the *initial conditions* of φ, and those of the form $[G](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda)$ the *clauses* or *global rules* of φ. The formula $\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda$, under the scope of a global operator, is referred to as *body* of the rule. In this paper we will use $\rightarrow$ to denote the classical logical implication, and $\Rightarrow$ to denote implicative rules, to emphasise the fact that rules are *not* logical implications in the classical sense. 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. 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 $T, [x, y] \not\Vdash \perp$ 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 exists $[w, z]$ 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 $[x, y]R_X[w, z]$, we have that $T, [w, z] \Vdash \psi$;
- $T, [x, y] \Vdash [G](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda)$ if, for all $[w, z]$ such that $T, [w, z] \Vdash \lambda_1 \wedge \cdots \wedge \lambda_k$ we have that $T, [w, z] \Vdash \lambda$;
- $T, [x, y] \Vdash \psi_1 \wedge \psi_2$ if $T, [x, y] \Vdash \psi_1$ and $T, [x, y] \Vdash \psi_2$.

In this work, we are interested in finite domains only, so, from now on, $D = \{0, 1, \ldots, N-1\}$. The set of propositional letters that are true on a given interval is also called *label* of the interval.

The chosen semantics for HS formulas is strict, irreflexive, and non-homogeneous; that is, point-intervals are excluded, the range of the Allen's relations R_X does not include the current interval, and there is no relationship between the truth value of a proposition over an interval and the truth value of the same proposition over its sub-intervals (in the homogeneous semantics, a letter p is true over an interval if and only if it is true in each of its sub-interval). Other choices are possible that may have an impact, sometimes dramatic, in the computational properties of the resulting logic (see, e.g., [7, 25] for homogeneous HS, and [9] for reflexive HS). However, in this paper we are concerned about rule extraction, not rule satisfiability, which means that in our case the semantical choices have little effect.

3 Applying Temporal Rules

There are several application domains in which temporal rules may have a relevant role. Here, we limit ourselves to some illuminating examples.

Rules in the medical domain. In the medical context a timeline may represent the *medical history* of a patient, that is, the collection of all relevant pieces of information about tests, results, symptoms, and hospitalizations of the patient that occurred during the entire observation period [13, 12]. As a concrete example, having a fever can be represented by the propositional letter Low – meaning lower than 40 degrees – or $High$ – meaning higher than or equal to 40 degrees; similarly, the proposition letter $Headache$ can be used to indicate the presence of a headache. Consider, now, a certain $Therapy$ that may be administered to a group of patient under consideration, and consider the problem of establishing the possible counter-effects that such medication or combination of medications may have, in particular, with the insurgence of headaches. Assume, now, that a relevant number of patients under observation show some relationship between the therapy and the insurgence of headache, and, in particular, a headache seems to start during the administration, but only if the patient is running high fever. Such a situation may be described by a Horn HS rule, as follows:

$$\rho : [G](Therapy \wedge \langle \overline{D} \rangle Fever \Rightarrow \langle O \rangle Headache).$$

Rules in natural language processing. In the context of natural language processing a timeline may represent a *conversation* between two individuals. As a matter of fact, it is sometimes interesting to label each interval of time of a conversation with one or more *contexts*, that is, a particular topic that is being discussed [31, 5, 6], in order to discover the existence of unexpected or interesting temporal relationships among them. Suppose, for example, that a certain company wants to analyze conversations between *agents* and *clients*. The agents contact the clients with the aim of selling a certain product, and it is known that certain contexts, such as the price of the product ($Price$), its known advantages ($Advantages$) over other products, and its possible minor defects ($Disadvantages$) are interesting. By analyzing a sufficiently high number of conversations, we may discover that if some known disadvantage is mentioned while discussing the price, the client typically shows some kind of negative reaction ($Negative$), which can be described by the rule:

$$\rho : [G](Price \wedge \langle D \rangle Disadvantages \Rightarrow \langle L \rangle Negative).$$

4 Extracting Interval Temporal Rules

In this section we describe our algorithm for temporal rule extraction from a temporal data set. We start by recalling some well-known concepts of static rule extraction, and by formalizing the problem of evaluating a static rule; then, we generalize our approach to the temporal case, and, finally, we describe a temporal version of the APRIORI algorithm for rule extraction.

Evaluating static rules. Extracting static rules from a non-temporal data base is founded on two simple concepts: support and confidence. Consider an abstract data base $\mathcal{T} = \{T_1, \ldots, T_n\}$, where each T_i is a *transaction* (that is, an instance) and it is characterized by a set of *items* $p_1, \ldots, p_m$ (each item is completely described by a propositional letter). A transaction T_i can be seen as a propositional model, and, therefore, the notion of T_i *satisfying* a set $P = \{p_1, p_2, \ldots\}$ of items ($T_i \Vdash P$) can be defined naturally. Such a notion can be immediately generalized to a set $\mathcal{T}' \subseteq \mathcal{T}$ of transactions, so that we can say that $\mathcal{T}'$ *satisfies* a set P of items ($\mathcal{T}' \Vdash P$) if every transaction in it does. Now, given a rule:

$$\rho : p_1 \wedge p_2 \wedge \ldots \wedge p_k \Rightarrow p,$$

Agrawal, Imieliński and Swami [2] implicitly define a notion of ρ *holding* on $\mathcal{T}$, which we denote $\mathcal{T} \Vdash^\dagger \rho$; given two real numbers s (*support*) and c (*confidence*), where $s, c \in (0, 1]$, we say that:

$$\mathcal{T} \Vdash^\dagger \rho \quad \text{if} \quad \exists \mathcal{T}' \subseteq \mathcal{T} \text{ s.t. } \begin{array}{l} \frac{|\mathcal{T}'|}{|\mathcal{T}|} \geq s, \mathcal{T}' \Vdash \{p_1, \ldots, p_k, p\}, \text{ and} \\ \forall \mathcal{T}'' \supseteq \mathcal{T}'(\mathcal{T}'' \Vdash \{p_1, \ldots, p_k\} \rightarrow \frac{|\mathcal{T}'|}{|\mathcal{T}''|} \geq c), \end{array} \tag{3}$$

that is, we say that ρ holds on $\mathcal{T}$ if the fraction of transactions that show all items is sufficiently high (guaranteeing that the rule has enough support), and the fraction of transactions that only show the antecedent but not the consequent is sufficiently low (ensuring that the rule is – statistically – confident); we say, therefore, that a rule is *meaningful* if it has enough support and confidence. In the context of the definition of $\Vdash^\dagger$, the set $\{p_1, p_2, \ldots, p_k, p\}$ of propositions that occur in a rule that holds on $\mathcal{T}$ is said to be a *frequent set of items* (or a *frequent set of propositions*), because its support is high.

Evaluating temporal rules. In our case, transactions are generalized to ordered sets of transactions, that is, timelines, and a temporal data set $\mathcal{T} = \{T_1, \ldots, T_n\}$ is a collection of timelines. In addition to being meaningful, we want our rules to be *temporally meaningful*. To better understand this concept, consider, again, the medical domain, and assume that *NoSym* denotes the absence of a particular symptom. Under the standard logical semantics, the bodies of rules as the following ones:

$$\rho_1 : [G](\mathit{Therapy} \Rightarrow [A]\mathit{NoSym}),$$
$$\rho_2 : [G](\mathit{Therapy} \Rightarrow [L]\mathit{NoSym}),$$

expressing the fact that *after the therapy the symptom disappears*, are true on every interval that ends at the last point of the model. Since there are no intervals that start at the last point or later the literals $[A]\mathit{NoSym}$ and $[L]\mathit{NoSym}$ are true, and this makes the implication trivially true. However, in this situation the rules are not temporally meaningful: we do not have enough points to check whether the rule holds or not (in the same sense as the relation $\Vdash^\dagger$ given above for static rules). Similarly, when establishing if a certain universal literal holds on an interval, for example, $[B]p$ on $[x, y]$, we should consider the literal as holding on $[x, y]$ not only when $[B]p$ is logically true on $[x, y]$, but also also when p is falsified by a small fraction of the intervals of the type $[x, z]$ with $z < y$. Finally, if the body of a rule holds over almost every interval of a timeline, it seems natural to conclude that the rule holds on that timeline, even if there are some intervals that falsify the rules.

To avoid degenerate situations, in addition to support (s) and confidence (c) we introduce several new parameters that help us to define a notion of literal (or temporal item) holding at an interval on a timeline, as well as a notion of global rule holding on a timeline. For each relation R_X we introduce a real parameter called *universal confidence*, given by a pair of real numbers (e_X, u_X), which we use to determine on which intervals it does not make sense to ask whether a certain universal literal $[X]\lambda$ holds (parameter e_X), and on which fraction of the set of intervals captured by $[X]$ should a certain literal λ be true for $[X]\lambda$ to

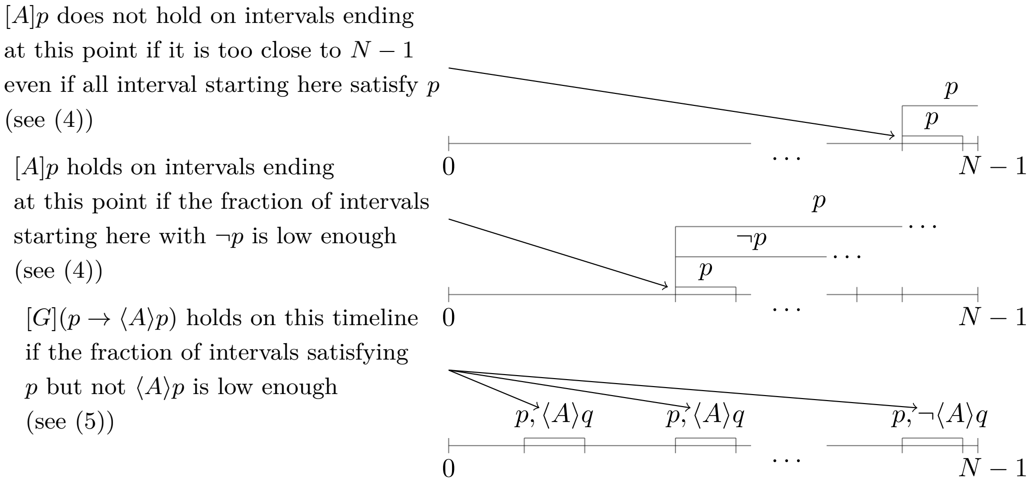

Figure 2 A pictorial (intuitive) representation of the concept of universal confidence (top and middle), and of that of global confidence (bottom).

be considered as holding on a given interval (parameter u_X). Recall that $T = \langle D, V \rangle$ and that D is finite, that is, $D = \{0, 1, \ldots, N-1\}$; for a given interval $[x, y]$, we define the notion of *literal holding at an interval* ($T, [x, y] \Vdash^{\dagger} \lambda$) using the classical notion of $T, [x, y] \Vdash \lambda$ if λ is a propositional letter or an existential literal, and the following definition for universal literals:

$$
\begin{array}{lll}
T, [x, y] \Vdash^{\dagger} [A]\lambda & \text{if} & \frac{y+1}{N} \leq e_A \text{ and } \forall D' \subseteq D(\forall z \in D'((z > y) \wedge T, [y, z] \nVdash^{\dagger} \lambda) \rightarrow \frac{|D'|}{|N-y-1|} \leq u_A); \\
T, [x, y] \Vdash^{\dagger} [B]\lambda & \text{if} & \frac{y-x}{N} \geq e_B \text{ and } \forall D' \subseteq D(\forall z \in D'((x < z < y) \wedge T, [x, z] \nVdash^{\dagger} \lambda) \rightarrow \frac{|D'|}{|y-x|} \leq u_B); \\
T, [x, y] \Vdash^{\dagger} [E]\lambda & \text{if} & \frac{y-x}{N} \geq e_E \text{ and } \forall D' \subseteq D(\forall z \in D'((x < z < y) \wedge T, [z, y] \nVdash^{\dagger} \lambda) \rightarrow \frac{|D'|}{|y-x|} \leq u_E).
\end{array}
\tag{4}
$$

The remaining modalities can be dealt with in a similar way. Notice that by setting $e_A = 1, e_B = e_E = 0$ and $u_A = u_B = u_E = 0$, one obtains the original, irreflexive semantics of HS. Moreover, simple modifications to the above semantics allow one to define reflexive and non-strict version of each operator.

Given a literal λ we say that λ *holds on a timeline* $T = (D, V)$, and we denote it by $T \Vdash^{\dagger} \lambda$, if there exists an interval $[x, y]$ such that $T, [x, y] \Vdash^{\dagger} \lambda$, and given a set $\Lambda = \{\lambda_1, \lambda_2, \ldots\}$, we say that Λ *holds on a timeline* T *at an interval* $[x, y]$, denoted by $T, [x, y] \Vdash^{\dagger} \Lambda$, if $T, [x, y] \Vdash^{\dagger} \lambda_j$ for each $\lambda_j \in \Lambda$. We also say that Λ *holds on a timeline* T, and write $T \Vdash^{\dagger} \Lambda$, to indicate that there exists an interval $[x, y]$ such that $T, [x, y] \Vdash^{\dagger} \Lambda$, and we write $T, I \Vdash^{\dagger} \Lambda$, where $I \subseteq I(D)$ if for each interval $[x, y] \in I$, we have $T, [x, y] \Vdash^{\dagger} \Lambda$. As in the static case, we indicate by $\mathcal{T}' \Vdash^{\dagger} \Lambda$ the fact that each timeline T in a set of timelines $\mathcal{T}' \subseteq \mathcal{T}$ is such that $T \Vdash^{\dagger} \Lambda$. Now, we introduce the real parameter $0 < g \leq 1$ (*global confidence*) to modulate on which fraction of all intervals of a timeline should the body of a rule hold to be considered a global rule. More precisely, given:

$$\rho : [G](\lambda_1 \wedge \ldots \wedge \lambda_k \Rightarrow \lambda),$$

we define the notion of ρ *holding on a timeline* T ($T \Vdash^{\dagger} \rho$) by imposing:

$$
\begin{array}{lll}
T \Vdash^{\dagger} \rho & \text{if} & T \Vdash^{\dagger} \{\lambda_1, \ldots, \lambda_k, \lambda\} \text{ and} \\
& & \forall I(I \subseteq I(D) \wedge T, I \Vdash^{\dagger} \{\lambda_1, \ldots, \lambda_k, \neg\lambda\} \rightarrow \frac{|I|}{|I(D)|} \leq g).
\end{array}
\tag{5}
$$

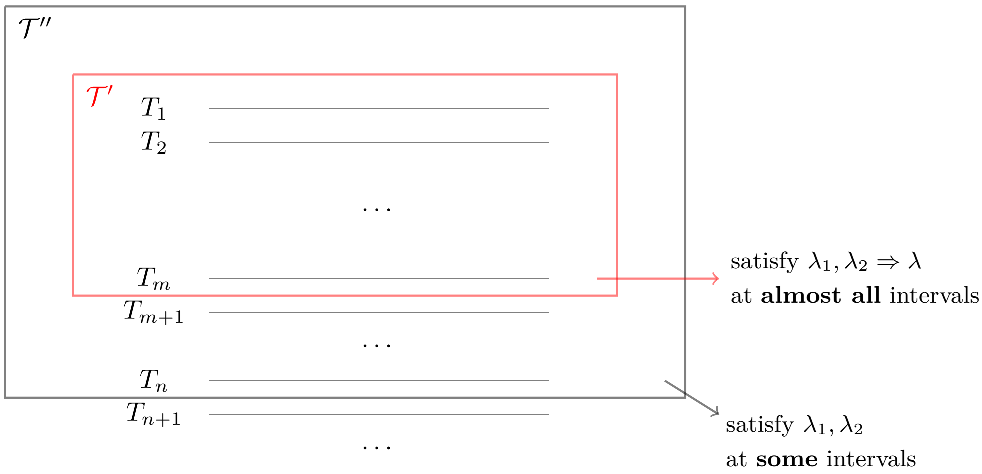

Figure 3 A pictorial (intuitive) representation of a rule $\lambda_1 \wedge \lambda_2 \Rightarrow \lambda$ holding on a temporal data set $\mathcal{T}$.

Setting $g = 0$ is equivalent to interpret a rule as a logical implication w.r.t. a single model (timeline). Fig. 2 gives an intuitive explanation of the use of universal and global confidence, which, together, we call *temporal confidence*.

Finally, we can generalize static evaluation to temporal evaluation, by using support and confidence, and define when ρ *holds in the temporal data set* $\mathcal{T}$ as follows:

$$\mathcal{T} \Vdash^{\dagger} \rho \quad \text{if} \quad \exists \mathcal{T}' \subseteq \mathcal{T} \quad \text{such that } \frac{|\mathcal{T}'|}{|\mathcal{T}|} \geq s \text{ and } \forall T \in \mathcal{T}', T \Vdash^{\dagger} \rho \text{ and} \\ \forall \mathcal{T}'' \supseteq \mathcal{T}'(\mathcal{T}'' \Vdash^{\dagger} \{\lambda_1, \dots, \lambda_k\} \rightarrow \frac{|\mathcal{T}'|}{|\mathcal{T}''|} \geq c). \tag{6}$$

In the context of temporal rule extraction, therefore, we say that ρ holds on $\mathcal{T}$ if it has enough support (the fraction of timelines in which the rule holds as defined in (5) is high) and it is confident (the fraction of timelines in which at least one interval satisfies $\{\lambda_1, \dots, \lambda_k\}$ but no interval satisfy $\{\lambda_1, \dots, \lambda_k, \lambda\}$ is low). A pictorial example of a simple rule holding on a temporal data set is given in Fig. 3.

To conclude, observe that while in static rules items are predetermined by the abstract data base of transactions, temporal items are not: for a given set $\mathcal{AP}$ of propositions, there are infinitely many possible literals, because there is no natural bound to the modal depth of the literals in the rules (i.e., the formulas) that are being mined. The most general solution to this problem is to introduce a parameter $m \in \mathbb{N}$, and limit ourselves to build rules in which literals have modal depth less or equal to m (the set of all such literals will be denoted by $\mathcal{L}_m(\mathcal{AP})$). Setting $m = 0$ is equivalent to extracting static rules that hold on (almost) every interval of the timelines in the support.

Temporal APRIORI. Our temporal extension of APRIORI is described in Algorithm 1. The procedure takes as input a temporal data set $\mathcal{T}$, the values for support, confidence, temporal confidence, and modal depth, and returns all and only global rules (limited to the specified modal depth) that hold on $\mathcal{T}$. In the following we analyse each step individually assuming, for the sake of simplicity, that $\mathcal{T}$ has n timelines of N points each, and that each timeline in $\mathcal{T}$ is *non-sparse* [26], that is, we assume that the number of intervals with non-empty propositional labels is at least linear in N (in this way, if timelines are represented as in Fig. 4, the complexity of the representation is linear in N).

Algorithm 1 Temporal APRIORI.

Require: $\mathcal{T}, s, c, g, (e_X, u_X), m$
1: $\Gamma, \mathcal{S} = \emptyset$;
2: compute the set $\mathcal{AP}$ of propositional letters occurring in $\mathcal{T}$;
3: compute the set $\mathcal{L}_m(\mathcal{AP})$;
4: label each $T_j \in \mathcal{T}$ and each interval $[x, y] \in I(D)$ with the maximal $\Lambda \subseteq \mathcal{L}_m(\mathcal{AP})$ such that $T_j, [x, y] \Vdash^\dagger \Lambda$;
5: **for** $\Lambda \subseteq \mathcal{L}_m(\mathcal{AP})$ **do**
6:   **if** $\exists \mathcal{T}' \subseteq \mathcal{T}$ such that $\frac{|\mathcal{T}'|}{|\mathcal{T}|} \geq s$ and $\mathcal{T}' \Vdash^\dagger \Lambda$ **then**
7:     $\mathcal{S} = \mathcal{S} \cup \{\Lambda\}$;
8:   **end if**
9: **end for**
10: **for** $\Lambda \in \mathcal{S}$ **do**
11:   **for** $\lambda \in \Lambda$ **do**
12:     put $\rho = [G](\lambda_1 \wedge \ldots \wedge \lambda_k \Rightarrow \lambda)$, where $\{\lambda_1, \ldots, \lambda_k, \lambda\} = \Lambda$;
13:     **if** $\mathcal{T} \Vdash^\dagger \rho$ **then**
14:       $\Gamma = \Gamma \cup \{\rho\}$
15:     **end if**
16:   **end for**
17: **end for**
18: **for** $\rho \in \Gamma$ **do**
19:   **if** ρ is redundant **then**
20:     $\Gamma = \Gamma \setminus \{\rho\}$;
21:   **end if**
22: **end for**
23: **return** Γ

- Computing temporal items for each timeline (lines 1–4). In our representation, scanning the data base to compute $\mathcal{AP}$ requires $O(|\mathcal{AP}| \cdot n)$ steps. Then, labeling each $T_i \in \mathcal{T}$ with the maximal set Λ of literals that holds at some interval of T_i while building, at the same time, the set $\mathcal{L}_m(\mathcal{AP})$ of all and only temporal items of depth up to m, requires executing (a simplified version of) the finite model checking algorithm for HS described in [26]. Under the assumption of non-sparseness, the algorithm requires $O(N^2 \cdot n \cdot |\mathcal{L}_m(\mathcal{AP})|)$ steps, using a symbolic representation, and $O(N^3 \cdot n \cdot |\mathcal{L}_m(\mathcal{AP})|)$ steps using the standard explicit representation.
- Computing frequent sets of literals (lines 5–9). The problem of finding frequent sets of temporal items can be reduced to the classical problem of finding frequent sets of items by adapting classical efficient solutions, such as FP-GROWTH [20], to the interval setting. FP-GROWTH builds a compact *frequent patterns tree* (also called *fp-tree*) that contains all necessary information for frequent sets generation. In analogy with the classical solution, our version of FP-GROWTH queries every timeline T in the temporal data base $\mathcal{T}$ a constant number of times to build the fp-tree [23]. In contrast to the classical solution, every query may need to examine every interval of the timeline to compute the frequency of some set of temporal items and and decide its position on the tree, and thus the complexity of a single query increases from $O(1)$ to $O(N^2)$. In terms of the overall complexity of the construction, this adds a factor $O(N^2)$ to the original solution, yet leaving the complexity linear in terms of the dimension of the temporal data base and polynomial in the number of frequent items. Since the number of frequent

N
$p : [0,1],[0,2],[0,3],[1,3],[2,3],\ldots$
$q : [0,1],[0,3],\ldots$
$\ldots$

Figure 4 A succinct representation of a timeline: the first line specifies the size of the frame (number of points in the model); the next lines encode the valuation function V.

items is bounded by the number of temporal items in $\mathcal{L}_m(\mathcal{AP})$, the worst-case complexity of FP-GROWTH in our setting is $O(n \cdot |\mathcal{L}_m(\mathcal{AP})| + |\mathcal{L}_m(\mathcal{AP})| \log |\mathcal{L}_m(\mathcal{AP})|)$, where $n \cdot |\mathcal{L}_m(\mathcal{AP})|$ is the cost of building the fp-tree and $|\mathcal{L}_m(\mathcal{AP})| \log |\mathcal{L}_m(\mathcal{AP})|$ the cost of sorting the set of frequent items by decreasing support (as required by the algortithm). Enumerating the frequent sets can be performed by suitably querying the fp-tree built by FP-GROWTH, which is, at this point, essentially indistinguishable from a non-temporal fp-tree. We can assume that the structure containing every frequent set, that is, $\mathcal{S}$, can be designed in such a way that the cost of querying the frequency of any $\Lambda \in \mathcal{S}$ is constant.

- Generating and testing rules (lines 10–17). For a single frequent set Λ, we generate $|\Lambda|$ different rules. Testing each one of the rules against (5) requires:
  (i) two lookup operations (each with constant cost) on the data structure that contains $\mathcal{S}$, to compare the support of the antecedent with the support of the entire set, and establish if the rule is confident, plus
  (ii) $O(n \cdot N^2)$ operations to establish if the rule is also global.
  Therefore this step costs $O(|\mathcal{S}| \cdot |\mathcal{L}_m(\mathcal{AP})| \cdot n \cdot N^2)$. While the size of the frequent patterns tree is polynomial in the dimension of the data set, the number of possible frequent sets that can be extracted from it (that is, $|\mathcal{S}|$) may be exponential, and its actual size depends on the particular data set as well as the setting of each parameter.
- Checking redundancy. From the set Γ of all meaningful rules that hold on $\mathcal{T}$, we eliminate all redundant ones. A rule $[G](\lambda_1 \wedge \ldots \wedge \lambda_k \Rightarrow \lambda)$ is redundant if Γ includes also a rule with λ as a consequence and with an antecedent that is a subset of $\{\lambda_1, \ldots, \lambda_k\}$. This step can be easily completed within $O(|\Gamma|^2 \cdot |\mathcal{L}_m(\mathcal{AP})|^2)$ operations (this limit can be improved with a suitable symbolic representation of the rules).

By summing up all the above steps we obtain the following overall worst-case complexity for Temporal APRIORI:

$$O(|\mathcal{S}| \cdot |\mathcal{L}_m(\mathcal{AP})| \cdot n \cdot N^2 + |\mathcal{L}_m(\mathcal{AP})| \log |\mathcal{L}_m(\mathcal{AP})| + |\Gamma|^2 \cdot |\mathcal{L}_m(\mathcal{AP})|^2).$$

The set $\mathcal{L}_m(\mathcal{AP})$ contains all positive temporal literals that can be built over the set of propositional letters $\mathcal{AP}$ with maximum modal depth m. Since the syntax of HS includes 24 different temporal modalities (one diamond and one box operator for each of the 12 Allen's relations), we have that $|\mathcal{L}_m(\mathcal{AP})| = O(24^m \cdot |\mathcal{AP}|)$. Moreover, in the worst-case scenario the set $\mathcal{S}$ includes all subsets of $\mathcal{L}_m(\mathcal{AP})$ while the set of meaningful rules includes all temporal rules that can be built from $\mathcal{L}_m(\mathcal{AP})$. These observations lead to a worst-case upper bound on the complexity of Temporal APRIORI that is exponential in m and $|\mathcal{AP}|$, but polynomial in n and N. As a general observation, for enumerative algorithms the exponential number of solutions makes the usual analysis of running time, that considers only the input size and ignore the output size, scarcely informative. In the literature, notions of efficiency have been developed for enumerative algorithms [22, 18], but a precise assessment of the complexity of Temporal APRIORI following such notions is outside the scope of this paper; here, we limit ourselves to some empirical considerations. First of all, performance studies show that the size of a fp-tree is usually substantially smaller than the original data base [20] and well beyond the worst-case upper bound of a complete tree. Similarly, providing a tighter upper bound to the number of frequent sets (and therefore to the number of rules) that can be extracted from a data set is not really possible as the number of frequent sets depends on the application domain and on the actual data. As we have observed, efficient implementation solutions for APRIORI can be adapted to our temporal generalization, and further optimizations should be focused on the temporal steps only. Experimental studies show that the elapsed time for efficient implementations of APRIORI grows linearly with the dimension of the data base. For comparison, assume that N and m are constants, and that the set $\mathcal{AP}$ (and therefore the set $\mathcal{L}_m(\mathcal{AP})$) is fixed. Then, the complexity of Temporal APRIORI is $O(n)$ plus the complexity of checking redundancy in rules. Since this latter aspect is not directly discussed in APRIORI, and the available experimental data do not include the redundancy check, in our simplified circumstances, Temporal APRIORI has the same complexity as APRIORI.

While considering a situation in which the set of interesting items grows is not really interesting, there are certain scenarios in which the length of the timelines may grow. Indeed, this may occur in both examples discussed in Section 3. In the medical scenario the length of timelines may grow as the duration of recorded medical histories increases with time. In the natural language knowledge extraction, timelines may grow as the average duration of a conversation increases. If N is not constant, the complexity of Temporal APRIORI (besides redundancy checking) is $O(N^2 \cdot n)$, and the bottleneck is temporal items labeling and rule globality checking.

5 Experimental Results

Testing the ability of Temporal APRIORI to extract meaningful rules from a temporal data set presents two main difficulties. First, there are very few publicly available data sets with a relevant temporal component. Such a data set, moreover, should be preprocessed in order to highlight the interesting propositions, and to hide irrelevant aspects (this is a standard procedure in knowledge extraction), which requires a profound knowledge of the applicative domain. In this way, the experiment would be influenced by such a preparation, making it more difficult for the performances of the algorithm to emerge. Second, the rules of a real-life data set are unknown beforehand; although there are indirect techniques that allow one to evaluate a set of rules, these evaluations are not always adequate to assess the capabilities of a new learning algorithm.

We designed a simple technique to generate an artificial data set $\mathcal{T}$ in such a way that we can control the rules to be extracted. Given a set Γ of input rules written in the language of Horn HS over a set $\mathcal{AP}$ of propositional letters, a natural number $N \geq 2$, fixed the parameter (e_X, u_X) for every relation R_X, and fixed the support s_ρ for every rule $\rho \in \Gamma$, our generator produces a temporal data set $\mathcal{T}$, where every timeline T has domain D_T of length N. To this end, for each rule $\rho \in \Gamma$, where $\rho : (\lambda_1 \wedge \ldots \wedge \lambda_k \Rightarrow \lambda)$, we first choose randomly the set $\mathcal{T}_\rho$ of timelines that will satisfy the rule ρ. Then, for each timeline $T \in \mathcal{T}_\rho$, we choose, randomly, an interval $[x, y]$, and we fix the propositional labeling of the intervals in T so that $T, [x, y] \Vdash^\dagger \{\lambda_1, \ldots, \lambda_k, \lambda\}$, which entails setting the labeling of a certain subset $S_{T,\rho}$ of $I(D_T)$ (e.g., forcing $\langle A \rangle p$ to be true on $[x, y]$ entails forcing p to be true on some interval $[y, z]$). The propositional labeling of every interval in the set $I(D_T) \setminus S_{T,\rho}$ is chosen randomly. Then, in order to ensure that ρ is global on T, every interval in $I(D)$ is considered (again) and its labeling fixed, if necessary. Notice that no contradictions may arise, since only positive

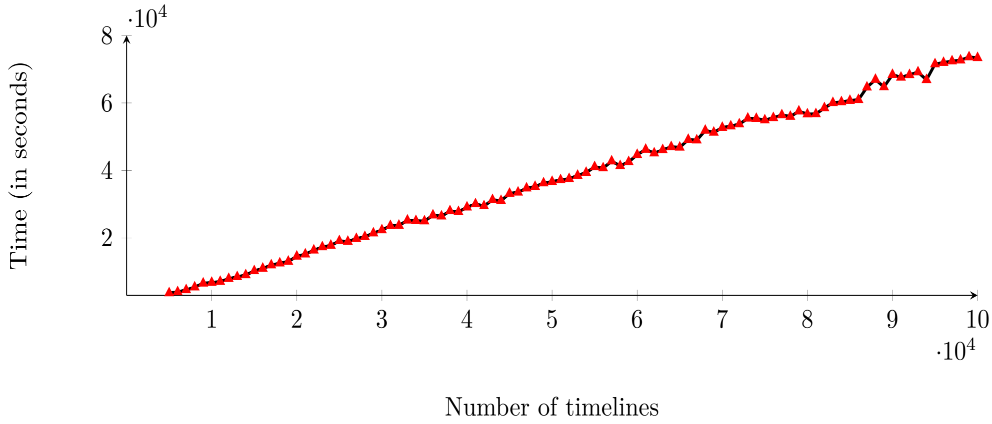

Figure 5 Elapsed time for rule extraction.

propositional literals are added at each step in each labeling: obviously, we generate only satisfiable examples of rules without $\perp$. In this way, the confidence of each rule ρ is 1, as well as its global confidence, and its support is equal to, or grater, than s_ρ. The aim of testing a rule extraction algorithm on an artificial data set is not that of finding an optimal set of parameters, which makes sense on real data sets only: in a controlled environment, every meaningful rule can be eventually discovered. Therefore, besides the proof-of-concept, our test is focused on performances only. We generated 96 temporal data sets, where timelines have a domain of cardinality less or equal to 100 points. The size of temporal data sets ranges from a minimum of 500 to a maximum of 10000 timelines. We fixed the following three rules:

$$\rho_1 : [G](p_1 \wedge [A]p_2 \wedge [B]p_3 \Rightarrow [A]p_4)$$
$$\rho_2 : [G]([A]p_1 \wedge [A]p_2 \Rightarrow [A]p_3)$$
$$\rho_3 : [G]([A]p_1 \wedge [E]p_2 \Rightarrow [B]p_3),$$

and we generated each data set in such a way that (at least) 60% of timelines satisfy ρ_1, the 80% of which satisfy also ρ_2, and the 60% of the latter satisfy also ρ_3.

As we have observed in the previous section, under certain conditions the complexity of Temporal APRIORI is linear in the number of timelines in $\mathcal{T}$. In our experiment, run on an Intel Core i7 with 4 cores at 2.80GHz, such conditions are met: N and m are constant (respectively 100 points an modal depth 1), and the set $\mathcal{AP} = \{p_1, p_2, p_3, p_4\}$ is fixed. Fig. 5 shows the expected linear behaviour.

6 Conclusions

Rule-based methods are algorithms that extract regularities in data that can be expressed in the form of *if-then* association rules, and are very popular in machine learning and data mining. The first and most recognized algorithm for extracting static association rules is known as APRIORI. The idea underlying APRIORI is that from a set of instances, referred to as transactions, each one characterized by a set of items (propositional letters), one can extract, first, frequent sets of items (that is, sets of propositions that occur often together), and, from them, rules. The latter are evaluated by using their support, that is, the fraction of transactions that satisfy all items in the rule, and their confidence, that is, the set of transactions that satisfy only the set of items corresponding to the antecedent of the rule.

This work is inspired by the fact that static rules, as those extracted by APRIORI, can be seen as implications written in the Horn fragment of propositional logic, though they are not interpreted as such. With the aim of extracting temporal rules from a temporal data set using similar principles, we first considered a suitable language that extends Horn propositional logic with temporal capabilities, that is, the Horn fragment of the temporal logic HS. Such a language has been recently studied, along with its computational properties, and the fact that it directly generalizes Horn propositional logic, and that some of its fragments have interesting computational characteristics, make it the ideal candidate for our purposes. Second, we generalized APRIORI in such a way that it is able to extract rules written in Horn HS. To this end, we introduced the notion of temporal confidence, which, paired with the classical notions of support and confidence, allowed us to define a relaxed version of the truth relation, and, ultimately, to devise the algorithm Temporal APRIORI, which we tested on an artificial data set.

While HS is a very general and expressively powerful language, the ability of Temporal APRIORI to extract meaningful rules decreases as the modal depth of the rules to be captured (that is, the cardinality of $\mathcal{L}_m(\mathcal{AP})$) increases. For example, in the language of HS, interpreted over finite linearly ordered sets, one can express the rule: *if p holds on some interval with length more than K, then q will hold on some of its subintervals.* However, this requires forcing an interval to have a minimal length, which in HS can be expressed as:

$$len_{>K} = \underbrace{\langle B\rangle\langle B\rangle \ldots \langle B\rangle}_{K} \top,$$

In turn, extracting a rule with $len_{>K}$ among its antecedents entails setting $m \geq K$, which, ultimately, leads the algorithm to generate and test a very high number of rules, and, in many cases, to return many rules that are not significant from the domain point of view. A similar situation arises with *coarser* Allen's relations in HS [30], which are relations that allow one to express an incomplete temporal knowledge (e.g.: *the symptom disappears after or immediately after the therapy*) and are expressible in the language of HS, but with formulas that do not immediately translate to Horn HS:

$$[G](Therapy \Rightarrow ([A]NoSym \vee [L]NoSym)).$$

As future work, we plan to improve Temporal APRIORI to extract rules with complex temporal literals and with an enriched set of basic relations in a direct way (that is, enriching the language in which the rules are written), effectively circumventing the above problems. Moreover, we plan to test Temporal APRIORI on natural data sets coming from relevant application domains like the medical domain and the natural language processing domain. This will allow us to evaluate the impact of parameters setting, as well as the expected improvement on the quality of extracted rules, with respect to real case studies.

References

1. L. Aceto, D. Della Monica, V. Goranko, A. Ingólfsdóttir, A. Montanari, and G. Sciavicco. A complete classification of the expressiveness of interval logics of Allen's relations: the general and the dense cases. *Acta Informatica*, 53(3):207–246, 2016.
2. R. Agrawal, T. Imieliński, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
3. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th International Conference on Data Engineering*, pages 3–14, 1995.

4 J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

5 R. Alluhaibi. Simple interval temporal logic for natural language assertion descriptions. In *Proc. of the 11th International Conference on Computational Semantics*, pages 283–293, 2015.

6 R.A. Baeza-Yates. Challenges in the interaction of information retrieval and natural language processing. In *Proc. of the 5th International on Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 445–456, 2004.

7 L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Satisfiability and model checking for the logic of sub-intervals under the homogeneity assumption. In *Proc. of the 44th International Colloquium on Automata, Languages, and Programming*, pages 120:1–120:14, 2017.

8 D. Bresolin, D. Della Monica, A. Montanari, and G. Sciavicco. The light side of interval temporal logic: the Bernays-Schönfinkel fragment of CDT. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):11–39, 2014.

9 D. Bresolin, A. Kurucz, E. Muñoz-Velasco, V. Ryzhikov, G. Sciavicco, and M. Zakharyaschev. Horn fragments of the Halpern-Shoham interval temporal logic. *ACM Transactions on Computational Logic*, 18(3):22:1–22:39, 2017.

10 D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco. Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. *Theoretical Compututer Science*, 560:269–291, 2014.

11 D. Bresolin, E. Muñoz-Velasco, and G. Sciavicco. Fast(er) reasoning in interval temporal logic. In *Proc. of the 26th Annual Conference on Computer Science Logic*, volume 82 of *LIPIcs*, pages 17:1–17:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

12 C. Combi, M. Mantovani, and P. Sala. Discovering quantitative temporal functional dependencies on clinical data. In *Proc. of the 2017 IEEE International Conference on Healthcare Informatics*, pages 248–257, 2017.

13 C. Combi and P. Sala. Mining approximate interval-based temporal dependencies. *Acta Informatica*, 53(6-8):547–585, 2016.

14 P. Cotofrei and K. Stoffel. Classification rules + time = temporal rules. In *Proc. of the 2002 International Conference on Computational Science*, pages 572–581, 2002.

15 S. de Amo, D. A. Furtado, A. Giacometti, and D. Laurent. An apriori-based approach for first-order temporal pattern. *Journal of Information and Data Management*, 1(1):57–70, 2010.

16 J. Fürnkranz, D. Gamberger, and N. Lavrac. *Foundations of Rule Learning.* Cognitive Technologies. Springer, 2012.

17 J. Fürnkranz and T. Kliegr. A brief overview of rule learning. In *Proc. of the 9th International Symposium Rule Technologies: Foundations, Tools, and Applications*, pages 54–69, 2015.

18 Leslie Ann Goldberg. *Efficient Algorithms for Listing Combinatorial Structures.* Distinguished Dissertations in Computer Science. Cambridge University Press, New York, NY, USA, 1993.

19 J.Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

20 J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

21 F. Jiménez, G. Sánchez, and J.M. Juárez. Multi-objective evolutionary algorithms for fuzzy classification in survival prediction. *Artificial Intelligence in Medicine*, 60(3):197–219, 2014.

22 David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.

23 W.A. Kosters, W. Pijls, and V. Popova. Complexity analysis of depth first and fp-growth implementations of APRIORI. In *Proc. of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 284–292, 2003.
24 Z. Michalewicz. *Genetic algorithms + data structures = evolution programs.* Artificial Intelligence. Springer, 1992.
25 A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron. Checking interval properties of computations. *Acta Informatica*, 53(6-8):587–619, 2016.
26 D. Della Monica, D. de Frutos-Escrig, A. Montanari, A. Murano, and G. Sciavicco. Evaluation of temporal datasets via interval temporal logic model checking. In *Proc. of the 24th International Symposium on Temporal Representation and Reasoning*, volume 90 of *LIPIcs*, pages 11:1–11:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
27 A. Montanari, I. Pratt-Hartmann, and P. Sala. Decidability of the logics of the reflexive sub-interval and super-interval relations over finite linear orders. In *Proc. of the 17th International Symposium on Temporal Representation and Reasoning*, pages 27–34. IEEE Computer Society, 2010.
28 A. Montanari, G. Puppis, and P. Sala. Maximal decidable fragments of Halpern and Shoham's modal logic of intervals. In *Proc. of the 37th International Colloquium on Automata, Languages and Programming*, volume 6199 of *LNCS*, pages 345–356. Springer, 2010.
29 S. Muggleton. Inductive logic programming: Issues, results and the challenge of learning language in logic. *Artificial Intelligence*, 114(1-2):283–296, 1999.
30 E. Muñoz-Velasco, M. Pelegrín-García, P. Sala, and G. Sciavicco. On coarser interval temporal logics and their satisfiability problem. In *Proc. of the 16th Conference of the Spanish Association for Artificial Intelligence*, volume 9422 of *LNCS*, pages 105–115. Springer, 2015.
31 I. Pratt-Hartmann. Temporal prepositions and their logic. *Artificial Intelligence*, 166(1–2):1–36, 2005.
32 L. De Raedt. *Logical and relational learning.* Cognitive Technologies. Springer, 2008.