Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 371 (2007) 247–264

Theoretical Computer Science

www.elsevier.com/locate/tcs

Horn axiomatizations for sequential data[☆]

José L. Balcázar*, Gemma C. Garriga

Departament de Llenguatges i Sistemes Informàtics, Laboratori d'Algorísmica Relacional, Complexitat i Aprenentatge, Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract

We propose a notion of deterministic association rules for ordered data. We prove that our proposed rules can be formally justified by a purely logical characterization, namely, a natural notion of empirical Horn approximation for ordered data which involves background Horn conditions; these ensure the consistency of the propositional theory obtained with the ordered context. The whole framework resorts to concept lattice models from Formal Concept Analysis, but adapted to ordered contexts. We also discuss a general method to mine these rules that can be easily incorporated into any algorithm for mining closed sequences, of which there are already some in the literature.

Keywords: Sequential patterns; Association rules; Closure operators; Propositional Horn theories

1. Introduction

The idea of extracting knowledge from sets of data emerged back in the 1990s, motivated by the decision support problem faced by several retail organizations that, thanks to technological progress, were able to store massive amounts of sales data. At that point, the research field of knowledge discovery started as an active area of investigation, and data mining, one of the most challenging steps of the process, was meant to provide efficient algorithms and techniques to automatize the exploratory analysis of the data. In the simplest form, such data is viewed as a set of transactions where each transaction is a set of items (attributes of the database), that is, a simple binary relation; this representation is known popularly as market-basket data.

One natural way of representing knowledge in this context is to look for causal relationships, where the presence of some facts suggests that other facts follow from them. One of the reasons of the success of the association rule framework is that in the presence of a community that tends to buy, say, sodas together with the less expensive spirits, a number of natural ideas to try to influence the behaviour of the buyers and profit from the patterns, easily come up. Approaches to find such associations started much before the area of knowledge discovery became so popular. For example, Duquenne and Guigues in [22] and also Luxenburger in [43], studied bases of minimal nonredundant sets

[☆] This work is supported in part by MCYT TIC 2002-04019-C03-01 (MOISES) and by the IST Programme of the European Union, under the PASCAL Network of Excellence, IST-2002-506778.

* Corresponding author.

E-mail addresses: balqui@lsi.upc.edu (J.L. Balcázar), garriga@lsi.upc.edu (G.C. Garriga).

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.11.009

of rules from where all other rules can be derived. The former studied these bases for association rules with 100% confidence, and the latter association rules with less than 100% confidence, but neither of them considered the support of the rules, i.e. number of transactions in the data supporting the rule. Nowadays, this task is widely known as the association rule mining problem, and it became a very popular topic since it was reformulated by Agrawal et al. in [1]. The reformulation made by Agrawal et al. introduced this notion of support, allowing the pruning of those rules whose number of occurrences in the data was not over a user-specified threshold.

Taking this association rule mining problem, there is a rich variety of algorithmic proposals whose strategy is to look for the frequent itemsets in the data, i.e. those sets of items with a number of occurrences (support) over a threshold and then, constructing implications between those discovered sets. The most well-known of these algorithms is Apriori [2]; it traverses the search space in a breadth-first fashion, also helped by the antimonotonicity property of support to prune unnecessary candidates. After this first proposal, many other algorithmic strategies and methods came out to improve the efficiency of Apriori: e.g. some of them suggested new structures to compact the original database into main memory, also others proposed a way to traverse the search space in a best-first fashion, or also performing the mining over a sample of the original transactions instead of all the data. Among many others, we may cite the works in [2,5,6,10,14,30–32,34,40,41,61].

Soon, the new problem was how to reduce the huge amount of association rules that were extracted by the algorithms. It was clear that some criteria were needed to make a judgement whether the extracted implication contained useful information or not. A classical way to rank the final rules is by means of statistical metrics (e.g. confidence [1,43], conviction [14], lift [13] and so on). There is a large number of proposals of how to measure the strength of implication of a rule, yet criticisms of various forms can be put forward for any measures; e.g. one of the criticisms for lift is its symmetry, which makes it impossible to orient rules. Surveys and comparisons, with appropriate references, are given in [4,11,12,24,28,33,50,52,55].

A complementary approach to ranking rules with statistical metrics consists of generating a basis of association rules from where the rest can be derived. As mentioned before, this approach was initially studied by Duquenne and Guigues in [22], and later by Luxenburger in [43]; yet, they did not consider any notion of minimum support for the rules. This idea evolved to considering only those frequent *closed itemsets*, instead of all the frequent itemsets, when first mining the data, and after that, generate only those rules indicated by the closure system (see e.g. [9,47,53,58,60, 62]). Using a similar idea, the work in [18] introduced a rule of inference and defined the notion of association rules cover as a minimal set of rules that are nonredundant with respect to this new rule of inference. Even if this idea of covering rules also results effective in practice, we will be more interested in the theory related to closed itemsets.

Closed itemsets are particularly interesting from a theoretical point of view, since their foundations are based on the mathematical theory of Formal Concept Analysis and concept lattices (cf. [16,19,25]). Broadly speaking, this theory is based on the definition of a Galois connection for a binary relation between a set of objects and a set of items, that is, the original data. This Galois connection gives rise to a closure system, i.e. a complete lattice of formal concepts. Each one of these concepts captures the information of closed itemsets, hence implications, in the data.

As the world evolves, there is the need to represent the data in more complex structures, such as sequences, trees or graphs. Dealing with these complex structures requires specific techniques and formalizations different from the ones commonly applied to single normalized tables. The most basic type that data can exhibit corresponds to the sequential categorical domain, i.e. elements follow in a specific sequential order. These elements in the sequence may have a simple form, such as a single item, or also have a more complex structure, such as sets of items or even a hierarchical organization. This task is more complex than the binary case because it faces the combinatorial explosion of searching and generating the new patterns, ranging from a plain structure (sequential subsequences) to more complex tree-like forms (such as partial orders).

The sequential mining problem was initially posed by Agrawal and Srikant in [3], and since then, most of the current work has focused on providing efficient algorithms for mining frequent patterns of various forms in the sequential data; e.g. works such as [44,45] are dedicated to the mining of partial orders, and others such as [48, 59] to the mining of frequent subsequences. However, compared to binary data, little work has been done to formulate a theory within the sequential mining framework. We consider that mining on sets of sequences is the first natural step to work towards the analysis of complex structured objects, and that the intuitions obtained in the sequential case will give a good insight into other complex combinatorial mining problems, such as having a set of graphs as our input data.

Therefore, this paper is devoted to the identification of an appropriate closure operator in order to apply closure-based mining processes to sequential data, developing its properties and formally justifying it by characterizing, in

Seq id	Input sequences
d_1	$\langle(AE)(C)(D)(A)\rangle$
d_2	$\langle(D)(ABE)(F)(BCD)\rangle$
d_3	$\langle(D)(A)(B)(F)\rangle$

Fig. 1. Example of a sequential database $\mathcal{D}$.

purely logical terms, the natural notion of deterministic association rules corresponding to this framework. We also discuss how to compute such closure operators and deterministic association rules on the basis of the output of existing algorithms for mining closed sequences.

A preliminary description of the most relevant contributions of the first half of this paper, together with some additional developments, was presented in [27]. The rest of the paper, in preliminary form, was presented at [8]. These contributions, together with a number of other related results, can be found as well in [26], corresponding to the thesis of the second author.

2. Preliminaries

Let $\mathcal{I} = \{i_1, \dots, i_n\}$ be a fixed set of items. A subset $I \subseteq \mathcal{I}$ is called an itemset. Formally, we deal with sequential categorical data, described as a collection of ordered transactions $\mathcal{D} = \{d_1, d_2, \dots d_n\}$, where each d_i is a *sequence*. The sequences $d_i \in \mathcal{D}$ will be called input sequences or transactions through all this text.

We consider a *sequence* to be an ordered list of itemsets. It can be represented as $\langle(I_1)(I_2)\dots(I_n)\rangle$, where each I_i is a subset of $\mathcal{I}$, and I_i comes before I_j if $i \leq j$. Note that we model each element of the sequence, not as an item, but as an itemset. This description of $\mathcal{D}$ corresponds exactly to the model of sequences originally proposed by Agrawal and Srikant in [3], and subsequently followed by other works on mining sequential data. Without loss of generality we assume that the items in each itemset are sorted in certain order (such as alphabetic order); itemsets will be displayed throughout this paper without the curly brackets, i.e. ACD represents indeed $\{A, C, D\}$. The universe of all the possible sequences will be denoted by $\mathcal{S}$. Note that the set of input sequences is a subset of this universe, i.e. $\mathcal{D} \subseteq \mathcal{S}$. A small example of $\mathcal{D}$ is presented in Fig. 1.

We say that a sequence $s = \langle(I_1)\dots(I_n)\rangle$ is a *subsequence* of another sequence $s' = \langle(I'_1)\dots(I'_m)\rangle$, denoted as $s \subseteq s'$, if there exist integers $1 \leq j_1 < j_2 \dots < j_n \leq m$ such that $I_1 \subseteq I'_{j_1}, \dots, I_n \subseteq I'_{j_n}$; then, we also say that s *is contained* in s'. For example, the sequence $\langle(C)(D)\rangle$ is contained in $\langle(AC)(D)(B)\rangle$, but it is not contained in $\langle(CD)(A)\rangle$. This is the commonly used interpretation proposed in [3,51]; we will describe briefly an alternative formalization in the last section of conclusions. We also define that $s \subset s'$ when $s \subseteq s'$ and $s \neq s'$. A sequence is *maximal* if it is not contained in any other sequence. A sequence s is *maximally contained* in s' if $s \subset s'$ and there is no s'' such that $s \subset s'' \subset s'$.

The transaction identifier list of a sequence s w.r.t. $\mathcal{D}$, denoted by $\text{tid}(s)$, is the list of input sequence identifiers from $\mathcal{D}$ where s is contained, e.g. we have that $\text{tid}(\langle(AE)(D)\rangle) = \{d_1, d_2\}$ for data in Fig. 1. For short, we will use natural numbers from 1 to n as identifiers, where n is the size of the original database. The *support* of a sequence s, denoted as $\text{supp}(s)$, is the number of occurrences of s in $\mathcal{D}$; e.g. $\text{supp}(\langle(AE)(D)\rangle) = |\text{tid}(\langle(AE)(D)\rangle)| = 2$. Note that, if $s \subseteq s'$, then the equalities $\text{supp}(s) = \text{supp}(s')$ and $\text{tid}(s) = \text{tid}(s')$ are equivalent. Of course both $\text{supp}(s)$ and $\text{tid}(s)$ do not only depend on s but also on the input database $\mathcal{D}$; we will not make explicit this dependence since we can assume $\mathcal{D}$ to be fixed for all our considerations in the paper.

2.1. *Mining closed sequential patterns*

A relevant task of the sequential mining problem is the identification of frequently-arising patterns or subsequences; in other words, finding those subsequences in $\mathcal{D}$ whose support is over a user-specified value. These frequent sequential patterns turn out to be useful in many domains, for instance, in anomaly detection for computer security [37–39]. Managing sequential patterns and counting their support in $\mathcal{D}$ is a challenging task since one needs to examine a combinatorially explosive number of possible frequent patterns. Many studies have contributed with algorithms to this problem e.g. [46,48,51,59]. Unfortunately, there are important cases where the number of frequent patterns is too large for a thorough examination and the algorithms face several computational problems; these include the cases of considering a very low threshold or a dense database.

Tid list	Closed Sequential Patterns
{1}	$\langle(AE)(C)(D)(A)\rangle$
{2}	$\langle(D)(ABE)(F)(BCD)\rangle$
{3}	$\langle(D)(A)(B)(F)\rangle$
{1, 2}	$\langle(AE)(C)\rangle$
{1, 2}	$\langle(AE)(D)\rangle$
{2, 3}	$\langle(D)(A)(B)\rangle$
{2, 3}	$\langle(D)(A)(F)\rangle$
{2, 3}	$\langle(D)(B)(F)\rangle$
{1, 2, 3}	$\langle(D)(A)\rangle$

Fig. 2. All closed sequences derived from data in Fig. 1.

Proper solutions to this were initially proposed in [57]. The main idea consists of mining just a compact and more significative set of sequential patterns called the *closed sequential patterns*. This idea parallels the notion of closed itemsets in a binary database, and indeed, both are defined as patterns not extendable to others with the same support. Formally, as in [57],

Definition 1. A sequence $s \in \mathcal{S}$ is **closed** (also known as closed sequential pattern) if there exists no sequence s' with $s \subset s'$ such that $\text{supp}(s) = \text{supp}(s')$.

For instance, taking data from Fig. 1, we have that $\langle(A)(F)\rangle$ is not closed since it can be extended to $\langle(D)(A)(F)\rangle$ in all the input sequences where it belongs. However, sequences such as $\langle(D)(A)\rangle$ or $\langle(AE)(C)\rangle$ are closed since they are maximal among those with the same tid list. The set of all the closed sequences and their tid lists from data in Fig. 1 are presented in Fig. 2.

In [57], the authors also present the first of a series of algorithms for mining closed sequences in $\mathcal{D}$ over a minimum support, named CloSpan. Later other algorithms followed up to improve its efficiency (e.g. TSP [54] or BIDE [56]). The way these algorithms work to identify those closed sequences in the data and their frequency is actually irrelevant for our purposes, and in general, we will refer to CloSpan as a representative of this group of algorithms that mine closed sequences. Mainly, we consider that the interestingness of using closed sequences relies on their theoretical characterization: while closed itemsets set up their basis on classical Formal Concept Analysis, ours is, as far as we are aware of, the first such characterization of the ordered counterpart.

As a token of the differences, note that it is easy to come up with examples (as the one given above) where several incomparable closed sequences share exactly the same set of transaction identifiers. In the case of sets, this would indicate nonmaximality and the union would be used, but this is no longer the case here. Therefore, the very notions of set-theoretic union and intersection lying under the standard formulation of the closure operator are to be reexamined in order to apply them to this case.

3. Lattice theory for sequences

Our first goal is to use Formal Concept Analysis to formalize a new closure system that characterizes sequential data. Since we are not dealing with the classical unordered context, setting all the conditions for the new closure operator is not a trivial task. To start with, it departs from the unordered case in the very definition of intersection; whereas the intersection of two itemsets is another itemset, the intersection of two or more sequences is not necessarily a single sequence. Let us consider the following definition.

Definition 2. The **intersection** of a collection of sequences $s_1, \ldots, s_n \in \mathcal{S}$, denoted $s_1 \cap \cdots \cap s_n$, is the set of subsequences maximally contained in all s_i.

Due to the maximality condition, the following property is clear: if $s \subseteq s_1, s \subseteq s_2, \ldots, s \subseteq s_n$ then there is some s' in $s_1 \cap \cdots \cap s_n$ such that $s \subseteq s'$. For example, the intersection of $s = \langle(AD)(C)(B)\rangle$ and $s' = \langle(A)(B)(D)(C)\rangle$ is the set of sequences $\{\langle(A)(C)\rangle, \langle(A)(B)\rangle, \langle(D)(C)\rangle\}$: all of them are contained in s and s' and among those having this property they are maximal; all other common subsequences are not maximal since they can be extended to one of these. The maximality condition of the intersection discards redundant information since the presence of, e.g. $\langle(A)(B)\rangle$, already informs of the presence of the itemsets (A) and (B) individually. Indeed, this notion of intersection addresses the intuition mentioned above: different closed sequences may coexist together in exactly the same input transactions. Then, the properties of the intersection naturally model the occurrence of maximal sequential patterns in the same set of input ordered data. These intuitions will be completely set up in this section by characterizing the concept lattice for sequences. We will compare sets of sequences according to the following relation $\preceq$.

Definition 3. We say that a set of sequences S is **more general** than another set of sequences S', denoted by $S \preceq S'$, if and only if for all $s \in S$, there exists $s' \in S'$ such that $s \subseteq s'$. Then S' is also said to be **more specific** than S.

According to the definition, the set of sequences $\{\langle(B)\rangle, \langle(C)\rangle, \langle(A)\rangle\}$ is more general than the set $\{\langle(B)(A)\rangle, \langle(C)(A)\rangle\}$. In fact this relation is not an ordering, but only a preorder; however, it will work as an order on those cases where no sequence in a set is a subsequence of another sequence of the same set, and this will be the case of interest (and the reason why the notion of intersection is restricted to maximal subsequences). Finally, note that $s \subseteq s'$ if and only if $\{s\} \preceq \{s'\}$, and that $S \preceq \{s'\}$ means that all the sequences in S are subsequences of s'.

3.1. A closure operator

We define the following two *derivation operators*: ϕ and ψ. To follow (approximately) the standard terminology of Formal Concept Analysis we call the natural numbers from 1 to n (the size of $\mathcal{D}$) *objects*, denoted $\mathcal{O} = \{1, \ldots, n\}$, and use capital letters O, O' for subsets of $\mathcal{O}$. Note that each object i identifies exactly one transaction d_i of $\mathcal{D}$; however, it could be the case that the sequence d_i itself does not identify i since there could be repeated sequences in the input database.

Given that in our ordered context the intersection of a set of input sequences may result in more than one sequence, we propose a first operator mapping a set of objects into a set of sequences, i.e. $\phi : 2^{\mathcal{O}} \longrightarrow 2^{\mathcal{S}}$. For consistency with this operator, the dual mapping is defined as $\psi : 2^{\mathcal{S}} \longrightarrow 2^{\mathcal{O}}$.

- For a set $O \subseteq \mathcal{O}$ of objects we define,

  $$\phi(O) = \{s \in \mathcal{S} | s \text{ maximally contained in } d_i, \text{ for all } i \in O\}.$$

  The mapping $\phi(O)$ returns the set of sequences common to *all* the objects in O. In fact, $\phi(O)$ is nothing but the intersection of the input sequences in $\mathcal{D}$ identified by O.
- Correspondingly, for a set $S \subseteq \mathcal{S}$ of sequences we define,

  $$\psi(S) = \{i \in \mathcal{O} | s \subseteq d_i, \text{ for all } s \in S\}.$$

  Function $\psi(S)$ returns the set of (indices of) input sequences that include *all* the sequences in S, that is, for the case of single sequences, $\psi(\{s\}) = \text{tid}(s)$, and, in general, $\psi(S)$ returns the transaction identifier list for the set of sequences S in $\mathcal{D}$. Of course $\psi(S)$ could be the empty set.

For example, for the data in Fig. 1 we have that $\phi(\{1, 3\}) = \{\langle(D)(A)\rangle\}$, and $\psi(\{\langle(AE)(D)\rangle, \langle(AE)(C)\rangle\}) = \{1, 2\}$. Now we have the following property, corresponding to the properties of Galois connections (although formally they must be used on orderings and we are only in the presence of a preorder, so that in its fully formal definition this pair is not completely a Galois connection).

Proposition 4. *For sets of objects $O, O' \subseteq \mathcal{O}$, and sets of sequences $S, S' \subseteq \mathcal{S}$, the following properties hold:*

(1) $O \subseteq O' \Rightarrow \phi(O') \preceq \phi(O)$ (1′) $S \preceq S' \Rightarrow \psi(S') \subseteq \psi(S)$
(2) $O \subseteq \psi(\phi(O))$ (2′) $S \preceq \phi(\psi(S))$.

Proof. Each one of these properties can be proved as follows:

(1) For all $s' \in \phi(O')$ we have that, for all $o' \in O'$, s' is contained in o', that is, in particular s' is contained in o for all $o \in O$, if $O \subseteq O'$, and therefore there exists $s \in \phi(O)$ such that $s' \subseteq s$, which means $\phi(O') \preceq \phi(O)$.
(2) For all $o \in O$ we have that s is contained in o for all $s \in \phi(O)$, and thus $o \in \psi(\phi(O))$.
(1′) For all $o' \in \psi(S')$ we have that, for all $s' \in S'$, s' is contained in o', that is, in particular s is contained in o' for all $s \in S$, if $S \preceq S'$, and thus $o' \in \psi(S)$, which means $\psi(S') \subseteq \psi(S)$.
(2′) For all $s \in S$ we have that s is contained in o for all $o \in \psi(S)$, and thus, there exists $s' \in \phi(\psi(S))$ such that $s \subseteq s'$, which implies $S \preceq \phi(\psi(S))$. □

Extent	Intent
$\{1, 2, 3\}$	$\{\langle(D)(A)\rangle\}$
$\{2, 3\}$	$\{\langle(D)(A)(B)\rangle, \langle(D)(A)(F)\rangle, \langle(D)(B)(F)\rangle\}$
$\{1, 2\}$	$\{\langle(AE)(C)\rangle, \langle(AE)(D)\rangle, \langle(D)(A)\rangle\}$
$\{1\}$	$\{\langle(AE)(C)(D)(A)\rangle\}$
$\{2\}$	$\{\langle(D)(ABE)(F)(BCD)\rangle\}$
$\{3\}$	$\{\langle(D)(A)(B)(F)\rangle\}$

Fig. 3. Closed concepts from data in Fig. 1.

From these properties, we can obtain two closure systems that are dually isomorphic to each other: one on sets of objects, obtained from the composition $\widehat{\Delta} = \psi \cdot \phi$, and another on sets of sequences, from $\Delta = \phi \cdot \psi$. In fact, Δ is our operator of interest, and works as follows: the closure $\Delta(S)$ of a set of sequences $S \in \mathcal{S}$, includes all the maximal sequences that are present in all objects having all sequences in S; that is, the intersection of all those input sequences $d \in \mathcal{D}$ such that $S \preceq \{d\}$. Taking the example from Fig. 1, we have that $\Delta(\{\langle(D)\rangle\})$ corresponds to the intersection of input sequences d_1, d_2 and d_3, i.e. the input sequences associated to the objects returned by the operator $\phi(\{\langle(D)\rangle\} = \{1, 2, 3\}$. Then, $\Delta(\{\langle(D)\rangle\}) = \{\langle(D)(A)\rangle\}$.

Proposition 5. *Compositions $\widehat{\Delta} = \psi \cdot \phi$ and $\Delta = \phi \cdot \psi$ are closure operators.*

Proof. According to [19] or [25], to show that Δ is a closure operator we need to prove: monotonicity: $S \preceq S'$ implies $\Delta(S) \preceq \Delta(S')$; extensivity: $S \preceq \Delta(S)$; and idempotency: $\Delta(\Delta(S)) = \Delta(S)$. These properties follow immediately from the facts in Proposition 4.

- Monotonicity: $S \preceq S'$ by (1′) yields to $\psi(S') \subseteq \psi(S)$, and by (1) we get $\phi(\psi(S)) \subseteq \phi(\psi(S'))$.
- Extensivity: follows directly from (2′).
- Idempotency: by property (2′) we have that $\phi(\psi(S)) \preceq \phi(\psi(\phi(\psi(S))))$, thus $\Delta(S) \preceq \Delta(\Delta(S))$; on the other hand, with $O := \psi(S)$ we obtain by property (2) that $\psi(S) \subseteq \psi(\phi(\psi(S)))$, and then properly (1) yields to $\phi(\psi(\phi(\psi(S)))) \preceq \phi(\psi(S))$, thus $\Delta(\Delta(S)) \preceq \Delta(S)$. Equality follows from the fact that the closure operator, by definition of ϕ, always gives a set where no sequence is a proper subsequence of another.

Symmetrically, the dual operator $\widehat{\Delta}$ can be proved to be a closure operator on the universe of objects. □

As customary, we can define the notion of closed sets of sequences.

Definition 6. Closed sets of sequences are those coinciding with their closure, that is, $\Delta(S) = S$.

A simple, but necessary observation is the following proposition.

Proposition 7. *All sequences in a closed set are maximal in it w.r.t. $\subseteq$.*

Proof. Since S is a closed set of sequences we have that $S = \Delta(S) = \phi(\psi(S))$. The definition of ϕ ensures that each $s \in S$ is a maximal sequence w.r.t. $\subseteq$, common to all of $\psi(S)$. □

Given the data of Fig. 1, the set of all closed concepts are shown in Fig. 3. Each concept corresponds to a pair (O, S), where S is a closed set of sequences (i.e. the intent of the concept), and $O = \psi(S)$ (i.e. the extent of the concept). These sets of objects form, at the same time, the dual system of closed set of objects.

Definition 8. If (O, S) and (O', S') are concepts, we say that (O, S) is a subconcept of (O', S') if $O' \subseteq O$ (equivalent to $S \preceq S'$).

In Fig. 4 we show the representation of the concepts of Fig. 3: each node corresponds to an intent of a concept, which is labelled by its extent; edges in the lattice correspond to the order between concepts $\leq$ (set-theoretical inclusion downwards by the extents, and $\preceq$ upwards by the intents). We also can see in the figure that, for each input sequence $d \in \mathcal{D}$, the set $\{d\}$ is a closed set; this always happens in general, also, and follows easily from the definition.

The set of sequences contained in all the input sequences will be called the *bottom* of the lattice; in most cases it will happen to be a trivial, somewhat artificial, element containing only the empty sequence. Similarly, we can also add an artificial set of sequences not contained in any object; this will represent the *top* of the concept lattice. In the

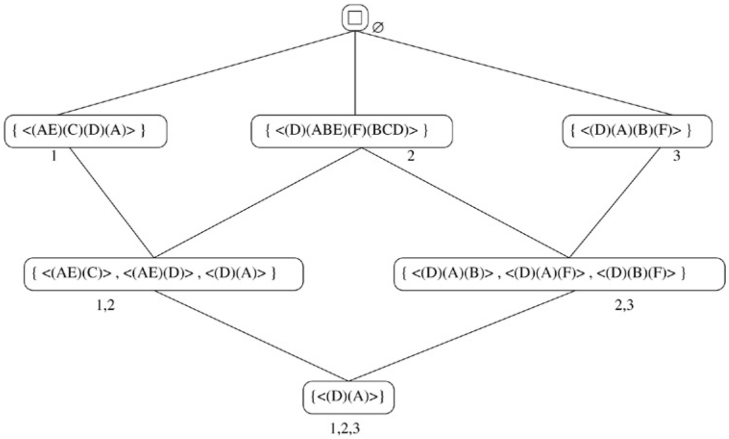


Fig. 4. Example of a concept lattice.

example from Fig. 4 an artificial top not belonging to any object is added, and we denote it by the unsatisfiable boolean constant □. This artificial element is added to the lattice just to the effect of our later arguments.

Finally, we say that a closed set of sequences S' is an *immediate predecessor* of another closed set of sequences S whenever $S' \preceq S$ and no other closed set S'' exists in the lattice with $S' \preceq S'' \preceq S$. For example, in Fig. 4 we have that the closed sets of sequences $\{\langle(AE)(C)\rangle, \langle(AE)(D)\rangle, \langle(D)(A)\rangle\}$ and $\{\langle(D)(A)(B)\rangle, \langle(D)(A)(F)\rangle, \langle(D)(B)(F)\rangle\}$ are immediate predecessors of the node $\{\langle(D)(ABE)(F)(BCD)\rangle\}$. Similarly, successors are found via the ascending paths of each node, and immediate successors are located just one level upwards.

4. Closed sequential patterns in the lattice

Our closure operator Δ can be used only on a set of sequences, not single individual sequences. In this section, we want to derive a notion of "closed" for an individual sequence, i.e. the notion of closed sequential patterns of algorithms such as CloSpan (as in Definition 1). This will provide the connection between the closed set of sequences defined with Formal Concept Analysis, and the closed patterns extracted by current mining algorithms.

First, it may be tempting to say that, individually, a sequence s is closed when the set $\{s\}$ is closed according to Δ; unfortunately, this intuition does not work here. For instance, for data in Fig. 1 we have that the sequence $\langle(AE)(C)\rangle$ is closed according to algorithm CloSpan (maximal among those sequences of support 2); however, when applying the closure operator to the set $\{\langle(AE)(C)\rangle\}$ we get: $\Delta(\{\langle(AE)(C)\rangle\}) = \{\langle(AE)(D)\rangle, \langle(AE)(C)\rangle, \langle(D)(A)\rangle\}$. So, we do not have a closed set of sequences. In general, our closure operator is not able to distinguish individual closed sequences directly, but still we keep the following clear property by extensivity of Δ: $\{s\} \preceq \Delta(\{s\})$.

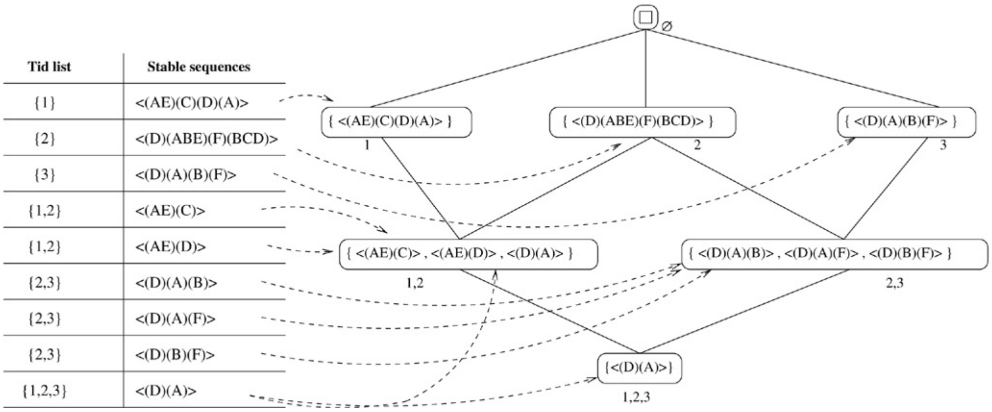
So, if we try to close a single sequence s with $\Delta(\{s\})$, we get a closed set of sequences where at least one of them is a supersequence of s. This property naturally leads to the following definition.

Definition 9. A single sequence s is **stable** under Δ if $s \in \Delta(\{s\})$.

Usually Δ is clear from the context and thus, it is omitted from the definition, so that we simply speak of stable sequences. This notion of stability gathers all those *maximal* sequences occurring in a *maximal* set of objects of the context; that is, individual sequences belonging to closed sets of sequences. In other words, a sequence s is stable for a set of objects O when we cannot add any other object to the set O without losing this s in the set of sequences obtained when mapping $\phi(O)$. Not all the sequences are stable; for example, sequence $\langle(A)(C)\rangle$ from the data of Fig. 1 is not stable since $\Delta(\{\langle(A)(C)\rangle\}) = \{\langle(AE)(C)\rangle, \langle(AE)(D)\rangle, \langle(D)(A)\rangle\}$. Indeed, just the closed sequential patterns can be characterized as stable sequences, i.e. we can prove that closed sequential patterns, extracted from data $\mathcal{D}$ by CloSpan following Definition 1, and stable sequences, defined via the closure operator in Definition 9, are the same sequences.

Lemma 10. *The set of stable sequences coincides with the set of closed sequential patterns.*

Proof. We will prove both directions of the equality: that any stable sequence is also a closed sequence, and that any closed sequence is in fact a stable sequence.



Tid list	Stable sequences
{1}	<(AE)(C)(D)(A)>
{2}	<(D)(ABE)(F)(BCD)>
{3}	<(D)(A)(B)(F)>
{1,2}	<(AE)(C)>
{1,2}	<(AE)(D)>
{2,3}	<(D)(A)(B)>
{2,3}	<(D)(A)(F)>
{2,3}	<(D)(B)(F)>
{1,2,3}	<(D)(A)>

Fig. 5. All the stable sequences derived from data in Fig. 1, and their correspondence with the lattice.

$\Rightarrow$/ If a sequence s is stable then $s \in \phi(\psi(\{s\}))$, which leads to: first, $|O| = \text{supp}(s)$ where $O = \psi(\{s\})$; and, second, s is *maximal* by Proposition 7. So, we can conclude that there is no other sequence s' such that $s \subset s'$ and s' is also contained in objects $o \in O$, that is, there is no supersequence of s with the same support. Thus, s is a closed sequence according to Definition 1.

$\Leftarrow$/ Let s be a closed sequence, and let $D \in \mathcal{D}$ be the set of all input sequences where this s is included ($|D| = \text{supp}(s)$). Because s is a closed sequential pattern, there is no s' such that $s \subset s'$ and $\text{supp}(s') = \text{supp}(s)$, or in other words, s is maximal contained in the set of transactions D. Thus, $s \in \phi(O)$, where O is formed by the tids of D in $\mathcal{D}$, and we necessarily have that $O = \psi(\{s\})$. Then, we get that $s \in \phi(\psi(\{s\}))$; so, s is stable. □

After proving that both sets of patterns are the same, we will rather name these specific sequences as stable sequences instead of closed sequential patterns; in this way, we will avoid the confusion of the term "closed", that we want to use to refer to the closure operator Δ, which only applies to sets of sequences.

Finally, to complete the characterization, we are interested in proving that the notion of stability, thus the notion of closed sequential patterns, is included in the concepts of our lattice. In other words, we are looking for a way to construct our closure system from the stable sequences, which we have just shown that can be mined by proper algorithms. The next result proves that the closed sets of sequences, thus the intents of our lattice, correspond to a set of raw stable sequences grouped together. This is the first step towards the lattice construction.

Theorem 11. *Let S be a closed set of sequences, then for all $s \in S$, s is stable.*

Proof. Let $\psi(S) = O$ and $\phi(O) = S$. For each single sequence $s \in S$ we examine the result of $\Delta(\{s\}) = \phi(\psi(\{s\}))$ to prove stability of s. Let $\psi(\{s\}) = O'$, so that:

- If $O' = O$, then $\phi(\psi(\{s\})) = S$, and since $s \in S$ we have that s is stable.
- If $O \subset O'$ (since it may well be that $\{s\}$ alone was contained in more objects than S, because $\{s\} \preceq S$), then by definition of Galois connection we have that $O \subseteq O'$, hence $\phi(O') \preceq \phi(O)$. Thus, $\phi(O') \preceq S$, which can be rewritten as $\phi(\psi\{s\}) \preceq S$. Since $s \in S$, then we get along with the extensivity of Δ, the following relation on the sets $\{s\} \preceq \phi(\psi(\{s\})) \preceq S$. However, we know that s is *maximal* in S by Proposition 7, so that there is no $s' \in S$ such that $s \subset s'$; therefore, $s \in \phi(\psi(\{s\})) = \Delta(\{s\})$ and we conclude that s is stable.
- It cannot occur that $O' \subset O$, since $\{s\} \preceq S$ and so $\{s\}$ has to be contained in at least the same objects where S is contained. □

Table of Fig. 5 shows the list of stable sequences that algorithms such as CloSpan would identify from the database presented in Fig. 1 (without minimum support condition). As is stated by the theorem, each one of the stable sequences corresponds to an individual sequence in the intents, and vice versa. We also observe from the figure that some stable sequences are contained in more than one node of the lattice. This is necessary to ensure the $\preceq$ upward relationship between the intents of the concepts. Algorithmically, the mined stable sequences with the same tid list are grouped in the same node of the lattice and then, these sets are updated with those other maximal stable sequences that ensure the $\preceq$ upward ordering between the intents (more details in [26]).

5. Horn axiomatizations for sequences

This section is devoted to the characterization of deterministic association rules with order. We will show here that our proposed rules can be formally justified by a natural notion of empirical Horn approximation for the ordered data. For this purpose, we first come back to the propositional logic framework and introduce some basic facts about Horn theories of the classical binary context.

Now assume a standard propositional logic language with propositional variables. The number of variables is finite, and we denote by $\mathcal{V}$ the set of all variables; but again, we could alternatively use an infinite set of variables provided that, for any fixed dataset, only finitely many propositional variables are ever needed (this is in fact the case of our application). A literal is either a propositional variable, called a positive literal, or its negation, called a negative literal. A clause is a disjunction of literals and can be seen simply as the set of the literals it contains. A clause is *Horn* if and only if it contains at most one positive literal. Horn clauses with a positive literal are called *definite*, and can be written as $H \to v$ where H is a conjunction of positive literals that were negative in the clause, whereas v is the single positive literal in the clause. Horn clauses without positive literals are called *nondefinite*, and can be written similarly as $H \to \Box$, where $\Box$ expresses unsatisfiability. A Horn formula is a conjunction of Horn clauses.

A *model* is a complete truth assignment, i.e. a mapping from the variables to $\{0, 1\}$. We denote by $m(v)$ the value that the model m assigns to the variable v. The intersection of two models is the bitwise conjunction, returning another model. A model satisfies a formula if the formula evaluates to true in the model. The set of all models will be denoted by $\mathcal{M}$.

A theory is a set of models. A theory is Horn if there is a Horn formula which axiomatizes it, in the sense that it is satisfied exactly by the models in the theory. When a theory contains another we say that the first is an upper bound for the second; for instance, by removing clauses from a Horn formula we get a larger or equal Horn theory. The following is known (cf. [17], or works such as e.g. [35]):

Fact 12. *Given a propositional theory T, there is exactly one minimal Horn theory containing it. Semantically, it contains all the models that are intersections of models of T. Syntactically, it can be described by the conjunction of all Horn clauses satisfied by all models from T.*

The theory obtained in this way is called sometimes the *empirical Horn approximation* of the original theory. Clearly, then, a theory T is Horn if and only if it is actually *closed under intersection*, so that it coincides with its empirical Horn approximation. These concepts are a cornerstone of the area of research known as Knowledge Compilation [15].

Interestingly, this framework allows us to cast the standard item set association rules of data mining in terms of the empirical Horn approximation of a set of binary models, as follows. In the main case of interest for data mining, the universe will be our set of items $\mathcal{I}$. Then, the closure operator is the same as in Formal Concept Analysis, and considerably simpler than the operator defined previously here: it just maps each set of items Z to the set of all the items that appear together with the items of Z in all the transactions. We denote this operator working on sets of items as Γ. In terms of a Galois connection, it is the composition of a function that leads from Z to its support, with a function that takes the intersection of all the transactions in that support [25]. It gives rise to closed sets of items, generators, and deterministic association rules. *Closed sets* are those sets of items that coincide with their closure, that is, $\Gamma(Z) = Z$ where $Z \subseteq \mathcal{I}$. When $\Gamma(G) = Z$ for a set G and G is minimal for that resulting Z, we say that G is a *generator* of Z. Implications of the form $G \to Z$ where G is a generator of Z, turn out to be the particular case of association rules where no support condition is imposed but confidence is 1 (or 100%) [47,49]. Such rules in this binary context are sometimes called *deterministic association rules*.

It turns out that it is possible to exactly characterize this set of deterministic association rules in terms of propositional logic: we can associate a propositional variable to each item; then transactions become models, and each association rule becomes a conjunction of Horn clauses with the same left hand side. Then:

Fact 13 (*[7]*). *Given a set of transactions, the conjunction of all the deterministic association rules defines exactly the empirical Horn approximation of the theory formed by the given tuples.*

So, the result determines that the empirical Horn approximation of the unordered data can be computed through the Formal Concept Analysis method of constructing deterministic association rules, that is, constructing the closed

Seq id	Sequence
d_1	$\langle(A)(B)(C)(D)\rangle$
d_2	$\langle(B)(C)(D)(A)\rangle$
d_3	$\langle(B)(C)(A)(D)\rangle$

Fig. 6. A simple example of ordered data $\mathcal{D}$.

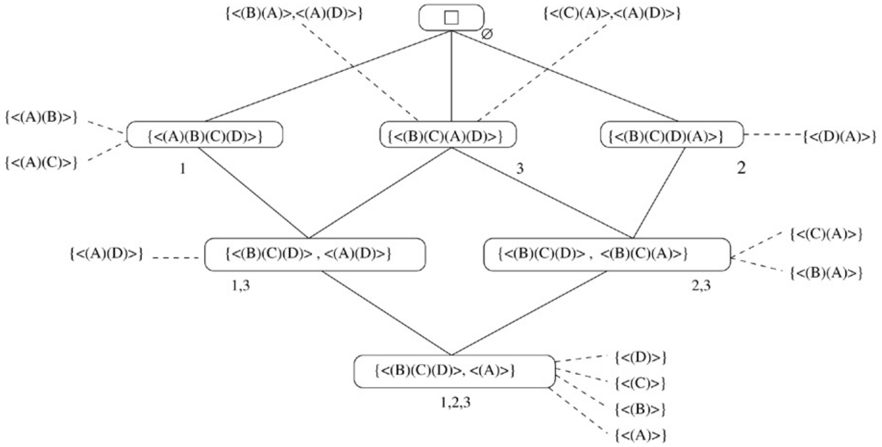


Fig. 7. Concept lattice for the dataset shown in Fig. 6 with all the minimal generators.

sets of items and identifying minimal generators for each closed set. A similar lattice theoretical approach was already used in works such as [20,21] to analyze functional dependencies in the data.

In this section we want to find a notion of deterministic association rules for the more complex case of sequential data (i.e. ordered context), and of course we would like to support our proposal by proving a similar characterization.

As in the binary context, a way to generate implications from the lattice is by defining generators. From these generators we will identify the notion of deterministic association rules with order.

Definition 14. We say that a set of sequences G is a **generator** of S if we have that $\Delta(G) = S$. We say that a generator G is **minimal** if there is no other G' such that $G' \preceq G$ and $G \neq G'$, such that $\Delta(G') = S$.

In our analysis, we will only consider minimal generators, since they do not contain redundancies. These will be graphically added to the concept lattice model with dashed lines. By the way of an example, in Fig. 6 we present a toy example where no repeated items are considered in the input sequences. The lattice with the minimal generators for this data is shown in Fig. 7. Minimal generators on the top of the lattice are not considered here, but, for the sake of illustration, it is easily seen that $\{\langle(C)(B)\rangle\}$ is among them.

Note that this simple data in Fig. 6 will ease the follow-up of the example, but of course, the theory developed here applies to the general model of data as well. At the end of this paper we will provide a more complex example to illustrate our results on data with repeated items. The following lemmas characterize exactly the relation between the generators and their associated closed set of sequences.

Lemma 15. *Let $\Delta(G) = S$; then $G \preceq S$ and, for all closed sets of sequences S' such that $S' \preceq S$ and $S' \neq S$, we have that $G \not\preceq S'$.*

Proof. That $G \preceq \Delta(G)$ follows from the fact that Δ is a closure operator. We prove the following contrapositive of the rest: for closed sets S and S', if $\Delta(G) = S$ and $G \preceq S' \preceq S$ then $S' = S$. Indeed, by monotonicity of Δ, $\Delta(G) \preceq \Delta(S') \preceq \Delta(S)$ and, being S and S' closed, this translates into $S \preceq S' \preceq S$. Using here the fact that all sequences in all closed sets are maximal in them, it follows that $S = S'$. □

Actually, this is just a rephrasing of the well-known fact that closure operators assign to each set the *minimal* closed set that is above it; in the standard case (binary data) the comparison is by the standard inclusion of sets, but here the peculiarity is that the comparison is according to $G \preceq S$, as defined above.

Lemma 16. *Let $G \preceq S$ where S is a closed set of sequences, and assume that, for all closed S', if $S' \preceq S$ and $S' \neq S$ then $G \not\preceq S'$; then G contains at least one minimal generator of S.*

Proof. Consider all subsets of G for which the same property indicated for G still holds. Since they are a finite family, at least one of them is minimal in the family (according to $\preceq$). Let $G_{\min}$ be this *minimal* subset of G that fulfils the property (or, any of them if there are several): $G_{\min} \preceq G \preceq S$, and for all closed $S' \preceq S$ such that $S' \neq S$, we have $G_{\min} \not\preceq S'$. Then, the minimal closed set of sequences containing $G_{\min}$ is S, and so, $\Delta(G_{\min}) = S$, being $G_{\min}$ one minimal generator contained in G. □

With the notion of generators set on place we are ready to define our family of deterministic association rules for sequences. We want to prove that this family of rules can be seen also as logical implications axiomatizing the empirical Horn approximation of an enriched theory. According to the same Formal Concept Analysis approach, as indicated above, we will consider in this section the deterministic association rules constructed from the generators of the lattice.

Definition 17. A **deterministic association rule with order** is a pair (G, S), denoted $G \to S$ where $G, S \subseteq \mathcal{S}$, such that $\Delta(G) = S$.

We say that a rule $G \to S$ *holds* for a given set of sequences $S' \subseteq \mathcal{S}$ if for all $s' \in S'$ either $G \not\preceq \{s'\}$ or $S \preceq \{s'\}$. Due to the construction of the closure operator Δ, we can argue directly that this method of constructing rules is sound, that is, all the rules of our proposed form that can be derived from an input set of sequences $\mathcal{D}$ do hold for each of those input sequences; we could say that our implications with order have confidence 1 (100%) in our data. Indeed, since $\{d\}$ is closed for each individual input sequence d of our database $\mathcal{D}$, we can consider any generator G and obtain, by monotonicity of Δ, $G \preceq \{d\}$ implies $\Delta(G) \preceq \Delta(\{d\}) = \{d\}$; that is, the implication $G \to \Delta(G)$ holds for $\{d\}$.

Now, we need to come back to the propositional logic framework and Horn theories and introduce background knowledge to define the empirical Horn approximation for ordered contexts. To motivate our choices, let us briefly discuss a feature of the analysis in [7]. Indeed, the first step there is to see each unordered transaction as a propositional model, and this is easy to obtain since, actually, it suffices to see the items as propositional variables. We can see this conceptual renaming as an isomorphism, or, even further, by using as propositional variables the very set of items, the translation is a mere identity function.

But this is no longer the case in our ordered contexts. Taking as propositional variables simply the items would not provide a sufficiently structured translation of our data sequences into propositional models. Thus, our next goal is to propose a more specific mapping that considers the ordered context. The resulting empirical Horn approximation of the ordered data will allow us to characterize the association rules defined in the previous section. By way of an example, consider Fig. 6, where the first object consists explicitly of the sequence $\langle(A)(B)(C)(D)\rangle$; however, it also contains implicitly all the subsequences $s' \subseteq \langle(A)(B)(C)(D)\rangle$. Thus, each input sequence can be also seen as a tuple of all those subsequences contained in it. Now we assign *one propositional variable to each subsequence* of each input sequence; and restrict the family of possible models by this background knowledge, thus discarding all models that would pretend to include a given sequence s but simultaneously avoid some subsequence of s.

More precisely, let m be a model: we impose on it the constraints that if $m(x) = 1$ for a propositional variable x, then $m(y) = 1$ for all those variables y such that y represents a subsequence of the sequence represented by x. For instance, if a propositional variable x corresponds to the sequence $\langle(A)(B)(C)\rangle$, then a model m assigning 1 to x should also assign 1 to the variable representing $\langle(A)(B)\rangle$, and similarly with other subsequences.

We define more specifically the interpretation of variables as sequences by an *injective* function $\xi : \mathcal{S} \to \mathcal{V}$, where $\mathcal{V}$ is the universe of all variables. For our convenience, we notationally extend this function with $\xi^{-1}(\square) = \square$, where $\square$ is the unsatisfiable boolean constant, corresponding also to the top of our lattice of closed sets of sequences. Now, each input sequence $d \in \mathcal{D}$ in the data corresponds to a model m_d: the one that sets to true exactly the variables $\xi(s')$ where $s' \subseteq d$; and we can find the empirical Horn approximation of the corresponding theory.

Definition 18. The set of models from $\mathcal{D}$ is $\text{models}(\mathcal{D}) = \{m_d \mid d \in \mathcal{D}\}$.

It is important that the constraints we have imposed to the models, that when $s' \subseteq s$ then $\xi(s) \to \xi(s')$, are indeed Horn clauses, which we call *background Horn conditions*, and hold on all input models, so that they are imposed automatically unto the whole Horn approximation: the conjunction of all Horn clauses satisfied by all the models corresponding to input sequences. We call this conjunction the *empirical Horn approximation for ordered data*, and any model there can be mapped back into a set of sequences that is closed downwards under the subsequence relation.

5.1. Main result

We are ready to present now the equivalence between the association rules extracted by the closure-based method, as presented above, and the empirical Horn approximation for ordered data.

Theorem 19. *Given a set of input sequences $\mathcal{D}$, the conjunction of all the deterministic association rules with order constructed by the closure system, seen as propositional formulas, and together with the background Horn conditions, axiomatizes exactly the empirical Horn approximation of the theory containing the set of models $M = models(\mathcal{D})$.*

Proof. We prove separately both directions: 1/ that the deterministic association rules (that is, their corresponding propositional implications) are implied by the empirical Horn approximation; and 2/ that all the clauses in the empirical Horn approximation are implied by the conjunction of the (propositional implications corresponding to) deterministic association rules.

$\Rightarrow$/ Consider a deterministic association rule $G \to S$ such that $\Delta(G) = S$. By distributivity, we can rewrite the rule as a conjunction of different implications $G \to s_i$ where $S = \{s_1, \ldots, s_m\} \in 2^{\mathcal{S}}$. As explained above, all the input sequences having as subsequences all the elements of G must have also s_i, so that the translation of $G \to s_i$ is a Horn clause that is true for all the given models in M and, by the theorems in works such as [35] (rephrased as Fact 12 here), it belongs to the empirical Horn approximation. Likewise, the background Horn conditions are also satisfied by all models and thus hold in the empirical Horn approximation.

$\Leftarrow$/ Let $F \to v$ be an arbitrary Horn clause where F is a set of variables, and v is a single variable. Assume this clause to be true for all the given models $M = \{m_d | d \in \mathcal{D}\}$ that correspond to the input sequences; note that these follow the constraints mentioned above: if $m \in M$, and $m(x) = 1$ for a propositional variable x, then $m(y) = 1$ for all those variables y such that $\xi^{-1}(y) \subseteq \xi^{-1}(x)$. In order to show that $F \to v$ is a consequence of the rules found from the concept lattice for $\mathcal{D}$, we will find an association rule that, upon translation, and in the presence of the background Horn conditions, logically implies our Horn clause.

Looking at F as a set of variables, we can consider the set of corresponding sequences $S' = \{\xi^{-1}(v) | v \in F\}$; let $\Delta(S') = S''$ be its closure. By previous Lemmas 15 and 16, we know that S' will contain at least one minimal generator of S'', that is, $G \preceq S'$ such that $\Delta(G) = S''$. Therefore, the rule $G \to S''$ will be one of the rules constructed by the Formal Concept Analysis method. On the other hand, we have assumed that the clause $F \to v$ holds for all the models M. By definition, it means that $S' \to \xi^{-1}(v)$ also holds in all the input sequences, in the sense that whenever $S' \preceq \{d\}$ for an input sequence d, also $\xi^{-1}(v) \subseteq d$; and this implies that $\{\xi^{-1}(v)\} \preceq \Delta(S') = S''$: so, for some sequence $s \in S''$ we have that $\xi^{-1}(v) \subseteq s$ or, equivalently, the Horn clause $\xi(s) \to v$ belongs to the background Horn conditions.

Finally, we have found that $G \to s$ is one of the rules composing $G \to S''$, which is one of the association rules coming from the closure system. Since $G \preceq S'$, the variables corresponding to sequences from G are all in F, and thus the clause $F' \to \xi(s)$ with $F' \subseteq F$ corresponds to one of the association rules. By subsumption, and one resolution step with $\xi(s) \to v$, we see that $F \to v$ follows indeed from the association rules plus the background Horn conditions. □

Note that this proof works also well when the Horn clause is nondefinite, that is, when considering $F \to \square$. In this case, no model from M satisfies all the variables in F, so, $S' \not\preceq \{d\}$ for all $d \in \mathcal{D}$; indeed we have that $\Delta(S') = \square$ (top of the lattice not included in any input sequence).

This characterization of Theorem 19 brings an immediate consequence: the closure operator of sets of sequences, named Δ, is equivalent to the closure operator for sets of items, named Γ, when the background Horn conditions hold in the considered models. In this case, both lattices turn out to be isomorphic.

Corollary 20. *Given a set of input sequences $\mathcal{D}$, let S be a set of sequences and Z be a set of propositional variables such that $Z = \{\xi(s') | s' \subseteq s \in S\}$, then, $\Delta(S) = S$ if and only if $\Gamma(Z) = Z$ for the set of models $M = \{m_d | d \in \mathcal{D}\} \subseteq \mathcal{M}$.*

Proof. For $\Delta(S) = S$, let M' be the set of models in $M' \subseteq M$ corresponding to $M' = \{m_d | d \in \mathcal{D}, S \preceq \{d\}\}$. By construction, all the variables in Z, as defined by the theorem, will be true in each one of the models in M', that keep the background Horn conditions. Moreover, since Z is constructed from S, and S is a closed set of sequences, we have that Z must be a maximal itemset and all its variables are not true in any other model apart from M'. Thus, we have that $\Gamma(Z) = Z$. In case of being S the top of the lattice, by definition we have that $\xi(\Box) = \Box$, thus getting the unsatisfiable boolean constant representing the top of the binary lattice.

Reciprocally, to prove the other direction of the theorem, let $\Gamma(Z) = Z$, that is, the set Z is a maximal set of variables true in a subset of models named M', where $M' = \{m | m \in M, m(x) = 1 \forall x \in Z\} \subseteq M$. If we consider that these models hold the background Horn conditions, we can find the set of input sequences $D' \subseteq \mathcal{D}$ equivalent to M'. For this set we have that $S \preceq \{d\}$ for all $d \in D'$. Again, by construction we necessarily have that S is not included in any other input sequence and it corresponds to the intersection of input sequences in D'. Thus, $\Delta(S) = S$. In case of dealing with the top element, i.e. $Z = \Box$, we always have $\xi^{-1}(\Box) = \Box$, hence, getting also the top of the lattice for the ordered context. □

6. Computing the rules with order

Next step is to discuss the algorithmic solutions for calculating all our implication rules with order. As proved before, the closure operator Δ characterizes the closed patterns of CloSpan (which are closed in the sense of not being extendable in support, thus stable) as those that belong to a closed set. This fact makes CloSpan a good candidate algorithm to construct the concepts of our lattice model. However, computing the deterministic association rules in the ordered data (equivalently, the empirical Horn approximation for the ordered context) we seem to need as well all the minimal generators, in order to output all rules $G \to S$ where S is closed and G is a minimal generator of S.

Thus, an important step is to add to any current algorithm of mining closed sequential patterns, the calculation of generators of each closed set. We want to compute them by means of a general method, so that it can be plugged into any underlying algorithm of mining closed sequential patterns such as either CloSpan (or BIDE or TSP). In this way, after computing the closed sets of sequences, the chosen algorithm can directly calculate the minimal generators as well, without incurring in inconvenient overheads for intersecting sequences of the database. In this section we show how to compute generators of S as a sort of transversal of appropriately defined differences between S and all immediate closed predecessors in the lattice.

The difficulty of this proposal will rely on the formalization of both steps: 1/ what it is exactly the difference between two sets of sequences, and 2/ how to properly define the appropriate variant of transversal. The motivation to look for such an approach is that the concept lattice we have obtained is isomorphic to a standard concept lattice (Corollary 20) for which such a method of computing rules does already exist [49]; note however that it is not immediate to carry over the isomorphism into the generators, so that we prefer to develop our method fully within the closure operator on sets of sequences.

For comparison purposes, we quote a result that we found in [49] and that we would like to export here, whereby the generators of a closed set in the unordered context obtained by a closure operator Γ are characterized (the original statement differs from ours but their equivalence is readily seen).

Fact 21. *Let Z be a closed set of items $Z = \Gamma(Z)$; the minimal generators of Z are found as the minimal transversal of the hypergraph of the differences $Z - Z'$ where Z' are the immediate closed subsets of Z in the lattice of itemsets.*

Along this line, other interesting works also developing their results on hypergraph transversals are e.g. [23,31] or [36]. The transversal hypergraph consists of sets that intersect each and every of the given differences (called *faces* in [49], a term that comes from related matroid-theoretic facts). Also, it is not difficult to see that it suffices to state that the generator intersects the differences with $Z - Z'$ for the closed immediate subsets of Z. For instance, let $Z = ABC$ be a closed set of items, whose immediate closed predecessors in the lattice are $Z'_1 = AB$ and $Z'_2 = AC$; then, the minimal generators of Z can be found by traversing the hypergraph of differences $H = \{Z - Z'_1, Z - Z'_2\}$, that is, $H = \{C, B\}$. The minimal transversal of H is CB, and so it is the minimal generator of Z.

We would like to have a similar result as Fact 21 for the minimal generators of the closed sets of sequences. After Corollary 20, we know that a closed set of sequences S can be seen as an equivalent closed set of variables $Z = \{\xi(s') | s' \subseteq s \in S\}$. Therefore, it is possible to characterize the generators of S through the transversals of the hypergraph of differences from the transformed closed Z. For the sake of clarity, here we decide to rewrite this method directly in the language of sequences.

We preserve here the term *faces* for our appropriate formalization of the differences between one closed set and its immediate closed predecessors (according to $\preceq$); for closed S, each face of S is $S - S'$, where $S' \preceq S$ is an immediate closed predecessor of S, and the difference is defined as:

$$S - S' = \{s | \{s\} \preceq S \text{ but } \{s\} \not\preceq S'\}.$$

The main property now is:

Lemma 22. *Let S be a closed set of sequences and $G \preceq S$; then $\Delta(G) = S$ if and only if G intersects all the faces of S.*

Here by *G intersecting a face $S - S'$* we understand set-theoretic intersection, that is, there must exist a common sequence in both. This corresponds to our notion of transversal for ordered data.

Proof. Assume first that G does not intersect the face $S - S'$, for some $S' \preceq S$; thus, no $s \in G$ fulfills the condition in the definition of the face. Since $G \preceq S$, for all such s, $\{s\} \preceq S$ as well, and this implies $\{s\} \preceq S'$, or actually $G \preceq S'$. Now, by monotonicity of Δ, from $G \preceq S' \preceq S$ and the fact that sequences in closed sets are maximal we obtain $S = S'$ just as in the proof of Lemma 15; and S' is not an immediate predecessor so that $S - S'$ is not a face. Conversely, assume that G indeed intersects all the faces; from $G \preceq S$ and monotonicity again we have $\Delta(G) \preceq S$. Equality will follow as we need, if we prove that $\Delta(G)$ is not an immediate predecessor. Indeed, by Lemma 15, $G \preceq \Delta(G)$, so for all $s \in G$, $\{s\} \preceq \Delta(G)$, which negates the condition in the definition of $S - \Delta(G)$. Thus it can't happen that any s is both in G and in $S - \Delta(G)$, and this last difference cannot be a face because G intersects all of them. This implies that $\Delta(G)$ is not an immediate predecessor. □

Again, we only need to consider immediate predecessors: if G intersects the faces corresponding to immediate predecessors, it must also intersect the other faces, which are larger. Additionally, we may be only interested in minimal generators (according to $\preceq$) since nonminimal generators only yield redundant association rules. However, a result such as Fact 21, that exactly characterizes minimal generators as minimal transversals, does not provide a direct way to compute minimal generators for our closed sets of sequences. In fact, not all the minimal generators that are obtained through Fact 21 in the propositional transformation correspond to minimal generators of the isomorphic closed set of sequences. A clarifying example will be provided in the next subsection.

Note that in order to construct the faces, we only need that, at the time of analyzing a given closed set S, the closed predecessors are known: we do not need the whole lattice. This allows for a sort of incremental processing, whereby as soon as the algorithm has discovered S, it can immediately construct the minimal generators for that S using its immediate predecessors in the lattice (which will be known with the bottom-up construction). The minimal generators G of a closed S found in this way can be used to construct the association rules of the ordered context, as it was explained above. A more graphical illustration of this construction will be provided in the next subsection.

6.1. An illustrative example of the method

Let $S = \{\langle(B)(C)(A)(D)\rangle\}$ be a closed set of sequences, as shown in the lattice of Fig. 7; the immediate predecessors of S are the closed set of sequences $S'_1 = \{\langle(B)(C)(D)\rangle, \langle(A)(D)\rangle\}$, and $S'_2 = \{\langle(B)(C)(D)\rangle, \langle(B)(C)(A)\rangle\}$. The minimal new subsequences in S not contained in S'_1 are the following set $F_1 = \{\langle(B)(A)\rangle, \langle(C)(A)\rangle\}$, and the minimal new subsequence in S not contained in S'_2 is $F_2 = \{\langle(A)(D)\rangle\}$, and these are the two faces. Now, to find the minimal generators of S we must minimally traverse these faces of S, by considering each subsequence as an atomic variable (i.e. set theoretic intersection between subsequences), and we obtain two generators: $G_1 = \{\langle(A)(D)\rangle, \langle(B)(A)\rangle\}$ and $G_2 = \{\langle(A)(D)\rangle, \langle(C)(A)\rangle\}$, which are exactly the minimal generators of S (see Fig. 7).

Observe what would have happened if we had applied the method proposed by Fact 21 over the isomorphic propositional transformation of this same node $S = \{\langle(B)(C)(A)(D)\rangle\}$: apart from the two generators G_1 and G_2 mentioned above, we would obtain other sets such as $\{\langle(B)(A)(D)\rangle\}$ or $\{\langle(B)(C)(A)(D)\rangle\}$. These correspond to minimal generators in the transformed space of propositional variables, yet their interpretation as a sequence does not satisfy the minimality condition required to be a minimal generator for the closed set S.

Seq id	Input sequences
d_1	$\langle(C)(B)(C)(A)(C)\rangle$
d_2	$\langle(C)(B)(A)(C)(C)(C)(A)\rangle$
d_3	$\langle(A)(C)(A)(C)(C)(A)(A)(A)\rangle$
d_4	$\langle(C)(A)(C)\rangle$

Fig. 8. Another set of sequential data.

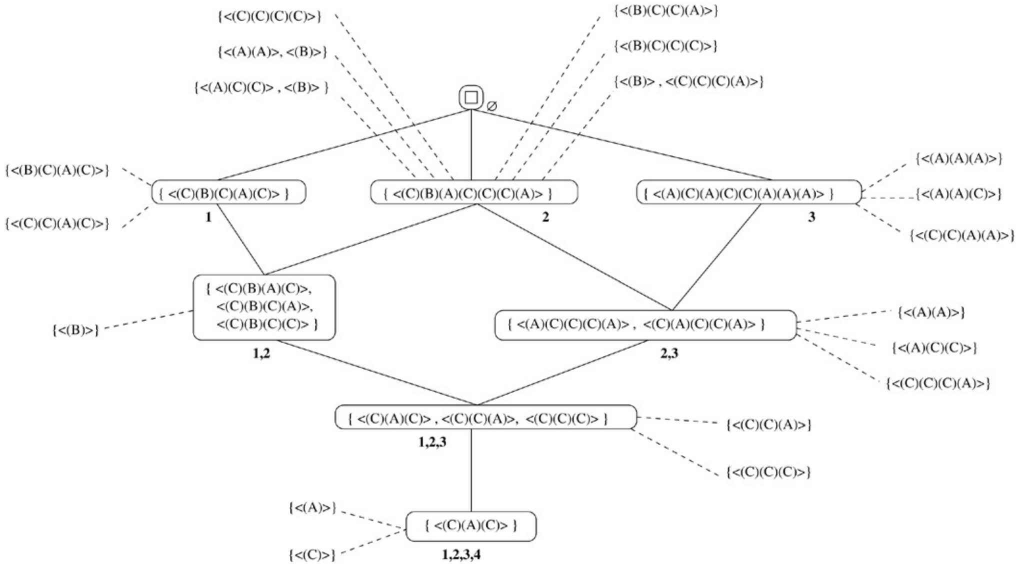


Fig. 9. Concept lattice for data in Fig. 8 with all the minimal generators.

For the sake of illustration, we provide also a more complex example of a set sequences with repeated items. We take the data from the following Fig. 8. The lattice of formal concepts along with the sets of generators is shown in Fig. 9. For a graphical follow up of the construction of our generators, take the closed set of sequences $S = \{\langle(C)(B)(A)(C)(C)(C)(A)\rangle\}$, whose immediate predecessors are $S'_1 = \{\langle(C)(B)(A)(C)\rangle, \langle(C)(B)(C)(A\rangle, \langle(C)(B)(C)(C)\rangle\}$ on the one hand, and the closed $S'_2 = \{\langle(A)(C)(C)(C)(A)\rangle, \langle(C)(A)(C)(C)(A)\rangle$ on the other hand. The minimal new subsequences in S not contained in S'_1 create the face $F_1 = \{\langle(A)(A)\rangle, \langle(A)(C)(C)\rangle, \langle(B)(C)(C)(A)\rangle, \langle(B)(C)(C)(C)\rangle, \langle(C)(C)(C)(A)\rangle, \langle(C)(C)(C)(C)\rangle\}$; the other face generated w.r.t. S'_2 is $F_2 = \{\langle(B)\rangle, \langle(C)(C)(C)(C)\rangle\}$. Note that we should consider that minimal new subsequences contained in S and not contained in the immediate predecessor, could have repeated items. For example, the sequence with one item $\langle(A)\rangle$ is not a minimal difference of S w.r.t. S'_1, yet the repeated $\langle(A)(A)\rangle$ is.

Now, to find the minimal generators of S, the algorithm minimally traverses these faces of S with a set-theoretic intersection. Note that the fact of having repeated items do not change the procedure: the set $\{\langle(C)(C)(C)(C)\rangle\}$ is a minimal generator by itself since it traverses both faces atomically. On the other hand, we also have sets such as $\{\langle(B)\rangle, \langle(A)(C)(C)\rangle\}$, being also a minimal traversal of the two faces, thus a minimal generator of S. At this point, we would like to make the special remark about two minimal transversals: $\{\langle(B)\rangle, \langle(B)(C)(C)(A)\rangle\}$ and $\{\langle(B)\rangle, \langle(B)(C)(C)(C)\rangle\}$. These generators also atomically intersect both F_1 and F_2, but indeed, they are equivalent to the nonredundant sets $\{\langle(B)(C)(C)(A)\rangle\}$ and $\{\langle(B)(C)(C)(C)\rangle\}$, which are the ones depicted in the lattice of Fig. 9. The rest of generators obtained by this algorithmic procedure can be followed from there.

Once generators are computed, we will be able to output the set of all deterministic association rules, forming the empirical Horn approximation of the sequential data. Each deterministic rule is created from $G \to S$ where $\Delta(G) = S$. By means of an example, we take the lattice in Fig. 9. From the bottom node we would generate two deterministic rules: $\langle(A)\rangle \to \langle(C)(A)(C)\rangle$ with one of the generators, and $\langle(C)\rangle \to \langle(C)(A)(C)\rangle$ with the other generator. Another example is a node of the same lattice, corresponding to the set $\{\langle(C)(B)(A)(C)\rangle, \langle(C)(B)(C)(A\rangle, \langle(C)(B)(C)(C)\rangle\}$, and whose unique generator is $\{\langle(B)\rangle\}$. In this case, we also output three deterministic rules corresponding to the

Tid list	Closed Sequential Patterns
$\{1\}$	$\langle(AE)(C)(D)(A)\rangle$
$\{2\}$	$\langle(D)(ABE)(F)(BCD)\rangle$
$\{3\}$	$\langle(D)(A)(B)(F)\rangle$
$\{1, 2\}$	$\langle(AE)(C)(D)\rangle$
$\{2, 3\}$	$\langle(D)(A)(B)(F)\rangle$
$\{1, 2, 3\}$	$\langle(D)(A)\rangle$

Fig. 10. All closed sequences derived from data in Fig. 1 with our new interpretation of the notion of subsequence.

generator implying each one of the three stable sequences in the closed set, that is, $\langle(B)\rangle \to \langle(C)(B)(A)(C)\rangle$, also $\langle(B)\rangle \to \langle(C)(B)(C)(A)\rangle$, and $\langle(B)\rangle \to \langle(C)(B)(C)(C)\rangle$.

7. Conclusions and further questions

Deterministic association rules are a powerful notion for mining transactional data, in particular when the data comes from scientific observations, where natural laws are actually enforcing coocurrence of attributes. Beyond their practical use, argued in [49], it was known that they had a precise meaning in terms of propositional logic (Fact 13 above).

One limitation of this study was its flat, unstructured transactional domain. In more and more cases, forthcoming evolutions of these processes will be required to take structure into account, e.g. in terms of the soon to be ubiquitous XML-structured datasets, that actually correspond combinatorially to trees. We believe that extending the frameworks of closure operators and association rules to these richer datasets is a worthy endeavour, and it is clear to us that the very first step was to consider sequences. This paper provides a closure operator and a mathematically justified notion of association rules for this case.

There are several levels of abstraction at which to pursue this research further, and we are actually working on them. One clear line is the application of our operator on sequences to other data mining tasks, together with the implementation of a system that allows actually for such analysis to be done in practice; work is in progress along this issue in [29]. Then, at a very abstract level, the natural question is the extension to other combinatorial structures (see also [42]); some reasonably successful preliminary studies [26] make us confident that pretty soon we will have interesting results to report along this line.

On the other hand, as a very concrete proposal of a variation, we want to observe the following. The notion of subsequence that we introduced is taken directly from the initial work of Agrawal and Srikant in [3,51], and this became the common accepted formalization. However, we may think of another interpretation of this subsequence operation, that is, considering that $s \subseteq s'$ if there exist integers $j_1 \le j_2 \cdots \le j_n$ (instead of $j_1 < j_2 \cdots < j_n$) such that $I_1 \subseteq I'_{j_1}, \ldots, I_n \subseteq I'_{j_n}$; thus allowing a sequence to be completely included in one of the itemsets of another sequence. For example, under this interpretation a sequence such as $\langle(A)(D)(B)\rangle$ would be included in another such as $\langle(ACD)(B)\rangle$. Here, we kept the interpretation given by Agrawal and Srikant in order to allow a better comparison with former works; yet we feel that this simple reinterpretation of subsequence can compact much more the number of discovered sequential patterns in $\mathcal{D}$.

For the sake of comparison, in Fig. 10 we provide the list of closed sequences that would be derived from the same data in Fig. 1, yet considering this redefined notion of subsequence. By allowing the total inclusion of one sequence into one itemset we get a more compacted set of final patterns: the classical interpretation of Agrawal and Srikant leads to nine closed sequences, whereas with the new interpretation we get six closed patterns. This example of advantage would justify a future formal study in order to provide reasons for preference of the old or the new definition.

References

[1] R. Agrawal, T. Imielinski, A.N. Swami, Mining association rules between sets of items in large databases, in: P. Buneman, S. Jajodia (Eds.), Proceedings of the 1993 ACM SIGMOD Int. Conference on Management of Data, 1993, pp. 207–216.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo, Fast discovery of association rules, in: Advances in Knowledge Discovery and Data Mining, 1996, pp. 307–328.

[3] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the 11th International Conference on Data Engineering, IEEE Computer Society Press, 1995, pp. 3–14.

[4] J. Azé, Extraction de Connaissances à partir de Données Numériques et Textuelles, Ph.D. Thesis, Université Paris Sud, 2003.

[5] J. Baixeries, G.C. Garriga, Sampling strategies for finding frequent sets, Revue des sciences et technologies de l'information (RTSI) 17 (1) (2003) 159–170.

[6] J. Baixeries, G.C. Garriga, J.L. Balcázar, Best-first strategies for mining frequent sets, Journal d'Extraction des connaissances et apprentissage 1 (4) (2002) 100–106.

[7] J.L. Balcázar, J. Baixeries, Discrete deterministic datamining as knowledge compilation, in: SIAM Int. Workshop on Discrete Mathematics and Data Mining, 2003.

[8] J.L. Balcázar, G.C. Garriga, On Horn axiomatizations for sequential data, in: Proceedings of the 10th Int. Conference on Database Theory, 2005, pp. 215–229.

[9] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, L. Lakhal, Mining minimal non-redundant association rules using frequent closed itemsets, Lecture Notes in Computer Science 1861 (2000) 972–986.

[10] R. Bayardo, Efficiently mining long patterns from databases, SIGMOD Record 27 (2) (1998) 85–93.

[11] R. Bayardo, R. Agrawal, Mining the most interesting rules, in: ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 1999, pp. 145–154.

[12] F. Berzal, I. Blanco, D. Sánchez, M.A. Vila, Measuring the accuracy and interest of association rules: A new framework, Journal Intelligent Data Analysis 6 (2002) 221–235.

[13] S. Brin, R. Motwani, C. Silverstein, Beyond market baskets: Generalizing association rules to correlations, in: Proceedings of the ACM SIGMOD Int. Conference on the Management of Data, 1997, pp. 265–276.

[14] S. Brin, R. Motwani, J.D. Ullman, S. Tsur, Dynamic itemset counting and implication rules for market basket data, in: Proceedings ACM SIGMOD Int. Conference on Management of Data, 1997, pp. 255–264.

[15] M. Cadoli, Knowledge compilation and approximation: Terminology, questions, references, in: AI/MATH-96, 4th. Int. Symposium on Artificial Intelligence and Mathematics, 1996.

[16] C. Carpineto, G. Romano, Concept Data Analysis. Theory and Applications, Wiley, 2004.

[17] C. Chang, J. Keisler, Model Theory, Elsevier, Amsterdam, Holland, 1990.

[18] L. Cristofor, D. Simovici, Generating an informative cover for association rules, in: Proceedings of the IEEE Int. Conference on Data Mining, 2002, pp. 597–600.

[19] B.A. Davey, H.A. Priestly, Introduction to Lattices and Order, 2002, Cambridge.

[20] A. Day, The lattice theory of functional dependencies and normal decompositions, International Journal of Algebra and Computation 2 (4) (1992) 409–431.

[21] J. Demetrovics, L. Libkin, I.B. Muchnik, Functional dependencies in relational databases: A lattice point of view, Discrete Applied Mathematics 40 (2) (1992) 155–185.

[22] V. Duquenne, J.-L. Guigues, Famille minimale d'implication informatives résultant d'un tableau de données binaires, Mathématiques et Sciences Humaines 24 (95) (1986) 5–18.

[23] T. Eiter, G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, SIAM Journal on Computing 24 (6) (1995) 1278–1304.

[24] J. Fürnkranz, P.A. Flach, An analysis of rule evaluation metrics, in: Proceedings of the 20th Int. Conference on Machine Learning, 2003, pp. 202–209.

[25] B. Ganter, R. Wille, Formal Concept Analysis. Mathematical Foundations, Springer, 1998.

[26] G.C. Garriga, Formal methods for mining structured objects, Ph.D. Dissertation, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya (in preparation).

[27] G.C. Garriga, Towards a formal framework for mining general patterns from structured data, in: Int. KDD Workshop on Multirelational Datamining, 2003, pp. 215–229.

[28] G.C. Garriga, Statistical strategies to remove all the uniteresting association rules, in: Proceedings of 16th European Conference on Artificial Intelligence, 2004, pp. 430–435.

[29] G.C. Garriga, Summarizing sequential data with closed partial orders, in: Proceedings of the SIAM Int. Conference on Data Mining, 2005, pp. 380–391.

[30] B. Goethals, M. Zaki, Advances in frequent itemset mining implementations: report on fimi'03, SIGKDD Explor. Newsl. 6 (1) (2004) 109–117.

[31] D. Gunopulos, R. Khardon, H. Mannila, S. Saluja, H. Toivonen, R.S. Sharma, Discovering all most specific sentences, ACM Transactions on Database Systems 28 (2) (2003).

[32] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the ACM SIGMOD Intl. Conference on Management of Data, ACM Press, 2000, pp. 1–12.

[33] R. Hilderman, H. Hamilton, Evaluation of interestingness measures for ranking discovered knowledge, in: Proceedings of 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2001, pp. 247–259.

[34] J. Hipp, U. Güntzer, G. Nakhaeizadeh, Algorithms for association rule mining: A general survey and comparison, SIGKDD Explor. Newsl. 2 (1) (2000) 58–64.

[35] H. Kautz, M. Kearns, B. Selman, Horn approximations of empirical data, Artificial Intelligence 74 (1) (1995) 129–145.

[36] D. Kavvadias, C.H. Papadimitriou, M. Sideri, On horn envelopes and hypergraph transversals, in: ISAAC'93: Proceedings of the 4th International Symposium on Algorithms and Computation, Springer-Verlag, London, UK, 1993, pp. 399–405.

[37] T. Lane, C.E. Brodley, Sequence matching and learning in anomaly detection for computer security, in: Proceedings AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management, 1997, pp. 43–49.

[38] W. Lee, S.J. Stolfo, P.K. Chan, Learning patterns from unix process execution traces for intrusion detection, in: Proceedings AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management, 1997, pp. 50–56.

[39] W. Lee, S.J. Stolfo, K. Mok, A data mining framework for building intrusion detection models, in: Proceedings of the IEEE Symposium on Security and Privacy, 1999, pp. 120–132.

[40] D. Lin, Z.M. Kedem, Pincer search: A new algorithm for discovering the maximum frequent set, in: Proceedings of the Int. Conference on Extending Database Technology (EDBT), 1998, pp. 105–119.

[41] J.L. Lin, M.H. Dunham, Mining association rules: Anti-skew algorithms, in: Proceedings of the 14th Int. Conference on Data Engineering, 23–27 February, 1998, Orlando, FL, USA, IEEE Computer Society, 1998, pp. 486–493.

[42] M. Liquiere, J. Sallantin, Structural machine learning with galois lattice and graphs, in: Proceedings of the 15th Int. Conference on Machine Learning, 1998, pp. 305–313.

[43] M. Luxenburger, Implications partielles dans un contexte, Math. Inf. Sci. Hum. 29 (113) (1991) 35–55.

[44] H. Mannila, C. Meek, Global partial orders from sequential data, in: Proceedings of the 6th Int. Conference on Knowledge Discovery in Databases, 2000, pp. 161–168.

[45] H. Mannila, H. Toivonen, A.I. Verkamo, Discovering frequent episodes in sequences, Data Mining and Knowledge Discovery 1 (3) (1997) 259–289.

[46] C. Martins-Antunes, A.L. Oliveira, Sequential pattern mining algorithms: Trade-offs between speed and memory, in: PKDD'04 Workshop on Mining Graphs, Trees and Sequences, 2004.

[47] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal, Closed set based discovery of small covers for association rules, in: Proceedings of the 15th Int. Conference on Advanced Databases, 1999, pp. 361–381.

[48] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M. Hsu, PrefixSpan: Mining sequential patterns by prefixprojected growth, in: Proceedings of the 17th Int. Conference on Data Engineering, 2001, pp. 215–224.

[49] J.L. Pfaltz, C.M. Taylor, Scientific knowledge discovery through iterative transformations of concept lattices, in: SIAM Int. Workshop on Discrete Mathematics and Data Mining, 2002, pp. 65–74.

[50] A. Silberschatz, A. Tuzhilin, On subjective measures of interestingness in knowledge discovery, in: Proceedings of the 1st Int. Conf. on Knowledge Discovery and Data Mining, 1995, pp. 275–281.

[51] R. Srikant, R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, in: Proceedings of the 5th Int. Conference Extending Database Technology, EDBT'96, vol. 1057, 1996, pp. 3–17.

[52] P. Tan, V. Kumar, J. Srivastava, Selecting the right interestingness measure for association patterns, in: ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, 2002, pp. 32–41.

[53] R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, Mining bases for association rules using closed sets, in: Proceedings of the 16th Int. Conference on Data Engineering, ICDE'00, IEEE Computer Society, 2000, p. 307.

[54] P. Tzvetkov, X. Yan, J. Han, TSP: Mining top-k closed sequential patterns, in: Proceedings of the 3rd IEEE International Conference on Data Mining, 2003, pp. 347–358.

[55] B. Vaillant, P. Lenca, S. Lallich, A clustering of interestingness measures, in: Discovery Science, 2004, pp. 290–297.

[56] J. Wang, J. Han, BIDE: Efficient mining of frequent closed sequences, in: Proceedings of the 19th Int. Conference on Data Engineering, 2003, pp. 79–90.

[57] X. Yan, J. Han, R. Afshar, CloSpan: Mining closed sequential patterns in large datasets, in: Proceedings of the Int. Conference SIAM Data Mining, 2003, pp. 166–177.

[58] M. Zaki, Generating non-redundant association rules, in: Proceedings of the 6th Int. Conference on Knowledge Discovery and Data Mining, 2000, pp. 34–43.

[59] M. Zaki, SPADE: An efficient algorithm for mining frequent sequences, in: Unsupervised Learning, Machine Learning Journal 42 (1–2) (2001) 31–60 (special issue).

[60] M. Zaki, Mining non-redundant association rules, Data Mining and Knowledge Discovery: An International Journal 4 (3) (2004) 223–248.

[61] M. Zaki, K. Gouda, Fast vertical mining using diffsets, in: Proceedings of the 9th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, KDD'03, ACM Press, 2003, pp. 326–335.

[62] M. Zaki, M. Ogihara, Theoretical foundations of association rules, in: SIGMOD-DMKD Int. Workshop on Research Issues in Data Mining and Knowledge Discovery, 1998.