Discriminant chronicle mining

Yann Dauxais, David Gross-Amblard, Thomas Guyet, André Happe

▶ To cite this version:

Discriminant chronicle mining

Yann Dauxais and David Gross-Amblard and Thomas Guyet and André Happe

Abstract Sequential pattern mining attempts to extract frequent behaviors from a sequential dataset. When sequences are labeled, it is interesting to extract behaviors that characterize each sequence class. This task is called discriminant pattern mining. In this paper, we introduce discriminant chronicle mining. Conceptually, a chronicle is a temporal graph whose vertices are events and whose edges represent numerical temporal constraints between these events. We propose *DCM*, an algorithm that mines discriminant chronicles. It is based on rule learning methods that extract the temporal constraints. Computational performances and discriminant power of extracted chronicles are evaluated on synthetic and real data. Finally, we apply this algorithm to the case study consisting in analyzing care pathways of epileptic patients.

1 Introduction

Discriminant pattern mining is dedicated to the mining of patterns in structured and labeled examples. Discriminant patterns can be used to label new examples and also inform about the specific characteristics of examples with a given label. In this work, examples are labeled temporal sequences. A temporal sequence is made of timestamped events and the label is assigned to sequence itself. Our objective is not

Yann Dauxais
Rennes University-1/IRISA-UMR 6074, e-mail: `yann.dauxais@irisa.fr`

David Gross-Amblard
Rennes University-1/IRISA-UMR 6074, e-mail: `dga@irisa.fr`

Thomas Guyet
AGROCAMPUS-OUEST/IRISA-UMR 6074, e-mail: `thomas.guyet@irisa.fr`

André Happe
CHRU Brest/EA-7449 REPERES e-mail: `andre.happe@chu-brest.fr`

to predict future events but to assign a label to a sequence. This latter task is also called sequence classification: each label can be seen as a class of sequences, and predicting the label is determining to which class a new sequence belongs.

Sequence classification can be encountered in numerous application fields: comparing groups of clients (*e.g.*, large family *vs* single child family), analyzing data related to supermarket purchases, identifying behavior of customers (who churns *vs* loyal customers), etc. Our application field is the analysis of care pathways, *i.e.* sequences of medical events of a patient (drug deliveries, hospital stays, etc.): we would like to characterize care pathways specificities for patients who have a disease. The objective is to identify therapeutic sequences that may foster diseases or to retrieve these patients in large unlabeled datasets. In all these contexts, the temporal dimension may hold the key information to discriminate sequences from one class to the other. For instance, a short delay between the delivery of two adverse drugs may help discriminate patients, sick from healthy patients. This is especially important for clinicians. By taking quantitative temporal constraints into account, we aim at improving classification accuracy, but discriminant patterns will also provide better insights about the therapeutic sequences.

Discovering such patterns raises important challenges both in the fields of pattern mining and machine learning. On the one hand, pattern mining approaches are based on a (exhaustive) search strategy in a structured version space. Our objective is to discover *quantitative* temporal information. This generates too large a search space for applying classical approaches. On the other hand, machine learning algorithms have difficulties to handle structured data such as sequences. As a consequence, we propose a specific pattern model and a new approach that combines solutions from these two research fields.

In this article, we explore temporal patterns called *chronicles* [Dousson and Duong, 1999]. A chronicle is a set of events linked by quantitative temporal constraints. In constraint satisfaction domain, chronicle can be seen as temporal constraint network [Dechter et al., 1991]. These complex but highly expressive patterns enable to take into account the quantitative temporal dimension of the data contrary to classical sequential patterns.

The contribution of this article is threefold:

1. We propose a new setting of mining discriminant chronicle from labeled sequences of timestamped events.
2. We propose the *DCM* algorithm, which relies on a rule learning algorithm to extract discriminant temporal constraints. To the best of our knowledge, this is the first approach that extracts discriminant patterns with quantitative temporal information.
3. We evaluate the *DCM* algorithm with a real case study, which consists in analyzing care pathways to answer a pharmaco-epidemiological question.

The remaining of the article is organized as follows. Next section presents related works. The section 3 introduces the discriminant chronicle mining task, while section 4 presents the *DCM* algorithm, our solution for this task. Section 5 presents

how to take decision from a set of chronicles. Sections 6 and 7 evaluate the *DCM* algorithm. After a first round of experiments on synthetic data, we compare the results of discriminant chronicles in classification context with those of the discriminant sequential patterns [Fradkin and Mörchen, 2015] on *UCI* datasets. Finally, we apply the *DCM* algorithm in order to extend the analysis of Polard et *al.* [Polard et al., 2015]. It aims at identifying possible associations between hospitalizations for a seizure and anti-epileptic drug switches from care pathways of epileptic patients.

2 Related work

Temporal pattern mining is a research field that proposes, and studies, algorithms to extract interesting patterns from temporal data. Such techniques have been mostly used in the medical application field.

Since the early articles on pattern mining [Agrawal and Srikant, 1995], most temporal pattern mining approaches have been focused on mining frequent patterns. Such method can be organized according to the temporal nature of the patterns they extract. *Sequential patterns* only takes the order of the events into account. Numerous works are based on this type of pattern and we refer the reader to existing reviews to have a wide scope of this field [Mabroukeh and Ezeife, 2010, Mooney and Roddick, 2013]. Sequential pattern mining has been used in [Wright et al., 2015] to identify temporal relationships between drugs. These relationships help to predict which medication a prescriber is likely to choose next. The total sequential order defined on the itemset of a sequential pattern may be too constrained to extract some recurrences in a sequence set. Some patterns like *episodes* ([Mannila et al., 1997] when extracted from data streams) or *partially ordered patterns* [Fabrègue et al., 2013] (when extracted from sequence set) were proposed to relieve this constraint. In [Achar et al., 2012], a framework is proposed to generalize and unify the different algorithms dedicated to frequent episodes mining based on the Apriori framework. *Temporal rules* [Concaro et al., 2009, Berlingerio et al., 2007], or more complex patterns like *chronicles* [Dousson and Duong, 1999, Cram et al., 2012, Huang et al., 2012, Alvarez et al., 2013], model inter-event duration based on the event timestamps. Chronicles have been originally introduced by Dousson and Duong [Dousson and Duong, 1999]. Cram et *al.* [Cram et al., 2012] proposed a complete algorithm to extract chronicles. These patterns have been used in the medical field [Huang et al., 2012, Alvarez et al., 2013]. Finally, *time interval patterns* [Moskovitch and Shahar, 2015, Guyet and Quiniou, 2011] capture patterns with typical timestamps and duration of events. Unlike the previous types of patterns, the search space size of such classes of pattern and the notion of a typical pattern does not allow to define complete algorithms. Unlike sequential patterns with numerical temporal constraints, such as *maxgap* ou *maxspan* [Pei et al., 2002], these approaches discover the numerical temporal characteristics of a pattern.

For large datasets, the number of temporal patterns may be huge and many of them not interesting. Extracting fewer but more significant patterns, becomes the

goal in this case. The first research directions specified additional constraints on the expected sequential patterns [Pei et al., 2002] including constraints on patterns (*e.g.*, pattern length constraints), on their occurrences (*e.g.*, area constraints) or on the pattern set (*e.g.*, the closed pattern constraint). More recent approaches try to use soft constraints, such as preferences, instead of hard constraints. For example, in [Lakshmanan et al., 2013], the temporal patterns are ranked according to their correlation with a particular patient outcome. It enables the expert to identify the potentially most interesting patterns easily. Nonetheless, they are not less numerous.

Mining discriminant sequential pattern can be seen as an additional hard constraint on the expected patterns. Several definitions, related to the notion of discrimination, have been proposed. Novak et *al.* [Novak et al., 2009] gathered these approaches, including emerging patterns [Dong and Li, 1999], contrast patterns [Bay and Pazzani, 2001], subgroup discovery [Herrera et al., 2011] and more recently Exceptional Model Mining [Duivesteijn et al., 2016]. All these approaches have been explored for itemset mining and can be easily extended to sequential pattern mining. But extracting quantitative temporal constraints becomes too complex when applying the same complete strategy. In our framework, the discrimination constraint is based on the growth rate of pattern frequency, which is similar to the emerging pattern approach.

Compared to all studies on frequent temporal data mining, there are only few proposals to mine discriminant temporal patterns. The early but solely approach uses inductive logic to extract patterns with quantified inter-event duration [Quiniou et al., 2001]. This approach requires expert knowledge to bound the search space. Indeed, the event types and a qualitative description of the boundaries (*e.g.*, *short*, *normal* or *long*) are specified by the user.

An alternative approach for learning how to label sequences is pattern-based classification [Bringmann et al., 2011]. In this approach, patterns are extracted and then used as features to create a dataset representing sequences by pattern occurrences. The correlation between sequence labels and pattern occurrences is learned by a (standard) classifier, such as a *SVM* classifier. Uninteresting (*e.g.*, redundant) patterns may be removed before the classification task in order to improve the prediction accuracy. Batal et *al.* [Batal et al., 2013] proposed to use discriminant temporal pattern as potentially interesting subset of frequent patterns. Their temporal patterns, called minimal predictive temporal patterns, are couples made of an itemset and Allen's temporal relations constraining the itemset occurrences [Allen, 1984]. The principle of minimal predictive temporal pattern selection is to compare a pattern $\mathscr{P}$ with those included in it to prune the search tree if $\mathscr{P}$ is predictive enough. In [Fradkin and Mörchen, 2015], timestamped event sequences are described by vectors of binary features encoding the presence/absence of patterns. Feature vectors are then classified by a *SVM* classifier. Their most accurate approach extracts all frequent sequential patterns and uses them all as features. The large number of extracted patterns and their redundancy is a problem for the efficiency of the *SVM* classifier. Fradkin et *al.* proposed several strategies to reduce the number of patterns to use in *SVM*. Different approaches (*BIDE-D*, *BIDE-DC*, *SMBT* and *SMBT-FS*)

dedicated to discriminant pattern mining are developed and tested to reduce this number while keeping the accuracy as high as possible.

Contrary to pattern mining approaches, the main drawback of these approaches is their interpretability (see for instance [Lipton, 2016] for a definition of interpretability). Interpretability is an essential property in our application field, but black-box classifier make hard to understand the association of patterns to behaviors. Our approach consists in extracting patterns that hold the discrimination information such that they can be individually interpreted: one discriminant chronicle describes an interpretable behavior encountered in the data, that is related to a specific sequence label.

Discriminant temporal patterns are finally used to present a few number of patterns to the experts [Fabrègue et al., 2014]. In this paper, discriminant patterns are extracted from hydro-ecological data to assist analysts with the extraction of knowledge. This approach does not extract discriminant patterns, but it selects those that satisfy a discrimination constraint among the frequent patterns.

Because of the great expressiveness of chronicles and their algorithmic properties, we are interested in frequent and discriminant chronicles mining. As in [Fabrègue et al., 2014], our goal is to support clinicians in extracting knowledge from data. Thus, our patterns must be interpretable rules instead of a black-box classifier. Mining discriminant chronicles offers a trade-off between interpretability and accuracy.

3 Discriminant chronicles

This section introduces more formally the basic definitions related to the discriminant chronicle mining task. In this paper, we address the task of mining sequences with timestamped events. More specifically, we have a supervised mining task. Each sequence of the dataset is labeled. From another point of view, we can say that sequences are categorized in several classes. We are interested in discovering patterns that are more frequent in one class than in the others. To take time into account, we search for patterns called chronicles [Dousson and Duong, 1999]. Briefly described, a chronicle is a set of events linked by temporal constraints.

3.1 Sequences and chronicles

Let $\mathbb{E}$ be a set of event types and $\mathbb{T}$ be a temporal domain where $\mathbb{T} \subseteq \mathbb{R}$. We assume that $\mathbb{E}$ is totally ordered by $\leq_{\mathbb{E}}$. An **event** is a pair (e,t) such that $e \in \mathbb{E}$ and $t \in \mathbb{T}$. A **sequence** is a tuple $\langle SID, \langle (e_1,t_1),(e_2,t_2),\ldots,(e_n,t_n) \rangle, L \rangle$ where SID is the sequence index, $\langle (e_1,t_1),(e_2,t_2),\ldots,(e_n,t_n) \rangle$ a finite event sequence and $L \in \mathbb{L}$ where $\mathbb{L}$ is a label set. Sequence items are ordered by $\leq_{\mathbb{T}}$ defined as $\forall i,j \in [1,n]$, $(e_i,t_i) \leq_{\mathbb{T}} (e_j,t_j) \Leftrightarrow t_i < t_j \vee (t_i = t_j \wedge e_i <_{\mathbb{E}} e_j)$.

SID	Sequence	Label
1	$(A,1), (B,3), (A,4), (C,5), (C,6), (D,7)$	$+$
2	$(B,2), (D,4), (A,5), (C,7)$	$+$
3	$(A,1), (B,4), (C,5), (B,6), (C,8), (D,9)$	$+$
4	$(B,4), (A,6), (E,8), (C,9)$	$-$
5	$(B,1), (A,3), (C,4)$	$-$
6	$(C,4), (B,5), (A,6), (C,7), (D,10)$	$-$

Table 1 Set of six sequences labeled with two classes $\{+,-\}$.

Example 1. Table 1 represents a set of six sequences containing five event types (A, B, C, D and E) and labeled with two different labels $\mathbb{L} = \{+,-\}$. In this example $\leq_{\mathbb{E}}$ is the lexicographic order.

A **temporal constraint** is a tuple (e_1, e_2, t^-, t^+), also noted $e_1[t^-, t^+]e_2$, where $e_1, e_2 \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ and $t^-, t^+ \in \mathbb{T} \cup \{-\infty, +\infty\}$, $t^- \leq t^+$. A temporal constraint $e_1[t^-, t^+]e_2$ is satisfied by a couple of events $((e,t), (e',t'))$ iff $e = e_1$, $e' = e_2$ and $t' - t \in [t^-, t^+]$. We define the inclusion of two temporal constraints as the inclusion of their intervals if their event types are identical. More formally, $e_1[a,b]e_2 \subseteq e'_1[a',b']e'_2$ iff $e_1 = e'_1, e_2 = e'_2$ and $[a,b] \subseteq [a',b']$.

A **chronicle** is a pair $(\mathscr{E}, \mathscr{T})$:

- The set $\mathscr{E} = \{\!\{e_1 \ldots e_n\}\!\}$, $e_i \in \mathbb{E}$ and $\forall i,j, 1 \leq i < j \leq n$, where $e_i \leq_{\mathbb{E}} e_j$ is a **multiset**, *i.e.* $\mathscr{E}$ can contain several occurrences of a same event type.
- The set $\mathscr{T}$ is a temporal constraint set: $\mathscr{T} = \{e[a,b]e' \mid e,e' \in \mathscr{E}, e \leq_{\mathbb{E}} e'\}$. As the constraint $e[a,b]e'$ is equivalent to $e'[-b,-a]e$, we impose the order on items, $\leq_{\mathbb{E}}$, to desambiguate which one is graphically represented in the chronicle. Moreover, when $e = e'$, we can obtain two different chronicles which are equivalent: one based on $e[a,b]e'$ where $a \geq 0$ [1] and one other based on $e'[-b,-a]e$. In such a case we choose the positive one because it is a more natural representation. Intuitively, two similar events can always be ordered: one is the earliest of their type in the sequence and the other is the latest. The positive temporal constraint represents this order.

Example 2. Fig. 1 illustrates three chronicles represented by directed graphs. Chronicle $\mathscr{C} = (\mathscr{E}, \mathscr{T})$ where $\mathscr{E} = \{\!\{e_1 = A, e_2 = B, e_3 = C, e_4 = C, e_5 = D\}\!\}$ and $\mathscr{T} = \{e_1[-1,3]e_2, e_1[-3,5]e_3, e_2[-2,2]e_3, e_2[4,5]e_5, e_3[1,3]e_4\}$ is illustrated at the top left. This graph is not complete. No edge between two events is equivalent to the temporal constraint $]-\infty, \infty[$, *i.e.*, there is no constraint. It is worth noticing that the oriented edges are directed by the order $\leq_{\mathbb{E}}$ and the labels on these edges the temporal constraints. For example, the edge between B and D represents $B \leq_{\mathbb{E}} D$. The label $[4,5]$ represents so the constraint $B[4,5]D$. It can be read as D occurs between 4 and 5 temporal units (*e.g.*, days) after B. Another example is the edge between A and B. This edge represents $A \leq_{\mathbb{E}} B$ and its label $A[-1,3]B$. Due to the negative

---

[1] Having a and b with different signs would have no meaning considering that the two events can be swapped. $e[-3,4]e$ is practically equivalent to $e[0,4]e$.
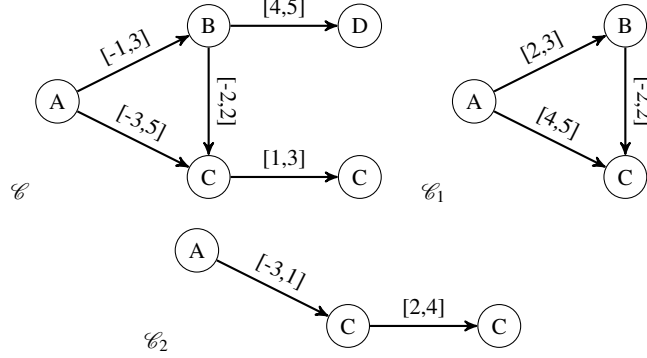
Fig. 1 Examples of three chronicles occurring in Table 1 (*cf.* Examples 2 and 5). No edge between two events is equivalent to the temporal constraint $]-\infty,\infty[$.

lower bound -1, it can be read as *B* occurs between 1 temporal unit before *A* and 3 units after *A*. Thus, the order $\leq_{\mathbb{E}}$ defines a reading order on the constraints and not a sequential order on the events.

It can be noticed that chronicles are temporal constraint networks [Dechter et al., 1991]. Temporal constraints are interdependent in such a way that we have to take care of, on the one side, their inconsistency and, on the other side, of their redundancy. Dechter et *al.* proposed an algorithm to reduce temporal constraints of a chronicle to an equivalent minimal chronicle, *i.e.* a chronicle with minimal interval lengths. Such reduction is useful to compare chronicles that seem different but share a same reduced form. Sahuguède et *al.* also proposed a distance to compare chronicles [Sahuguède et al., 2018].

Example 3. Fig. 2 illustrates reduced chronicles equivalent to $\mathscr{C}_1$ and $\mathscr{C}_2$ from Fig. 1. We see that the difference between $\mathscr{C}_1$ and $\mathscr{C}_1'$ is the temporal constraint $B[-2,2]C$ for $\mathscr{C}_1$ and the temporal constraint $B[1,2]C$ for $\mathscr{C}_1'$. Indeed, as we have $A[4,5]C$ and $A[2,3]B$, it is not possible to observe *C* 2 time units before *B*. If *B* occurs 2 time units after *A*, *C* must occur 2 or 3 time units after *B* to satisfy $A[4,5]C$. If *B* occurs 3 time units after *A*, *C* must occur 1 or 2 time units after *B* to satisfy $A[4,5]C$. As 3 was not in the original temporal interval $[-2,2]$, we obtained with this reasoning the new temporal constraint $B[1,2]C$.

The reasoning is similar for $\mathscr{C}_2$ except that the interval between *A* and the second *C* is initially $]-\infty,+\infty[$. By adding the lower bounds of the temporal constraints between *A* and the first *C* (-3) and between the two *C* (2) we obtain the lower bound between *A* and the second *C* (-1) and by adding the upper bounds (1 and 4) we obtain the upper bound (5).

Given two chronicles $\mathscr{C}_1 = (\mathscr{E}_1, \mathscr{T}_1)$ and $\mathscr{C}_2 = (\mathscr{E}_2, \mathscr{T}_2)$, we define the partial order $\preceq$ where $\mathscr{C}_1 \preceq \mathscr{C}_2$ if $\mathscr{E}_2 \subseteq \mathscr{E}_1$ and there is a strictly increasing function f where $\forall i,j,\ 1 \leq i < j \leq |\mathscr{E}_2|,\ 1 \leq f(i) < f(j) \leq |\mathscr{E}_1|,\ e_i, e_j \in \mathscr{E}_2,\ e_{f(i)}, e_{f(j)} \in \mathscr{E}_1,$ $e_{f(i)}[a,b]e_{f(j)} \in \mathscr{T}_1,\ e_i[a',b']e_j \in \mathscr{T}_2,\ e_{f(i)}[a,b]e_{f(j)} \subseteq e_i[a',b']e_j$. If $\mathscr{C}_1 \preceq \mathscr{C}_2$ and
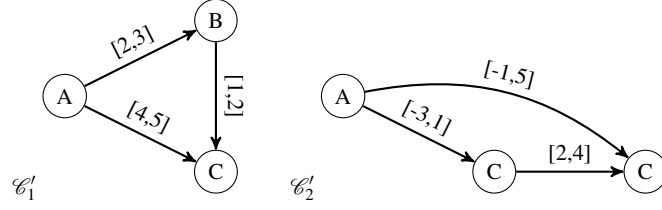
Fig. 2 The chronicles $\mathscr{C}'_1$ and $\mathscr{C}'_2$ are respectively the reduced chronicles of $\mathscr{C}_1$ and $\mathscr{C}_2$ of Fig. 1.
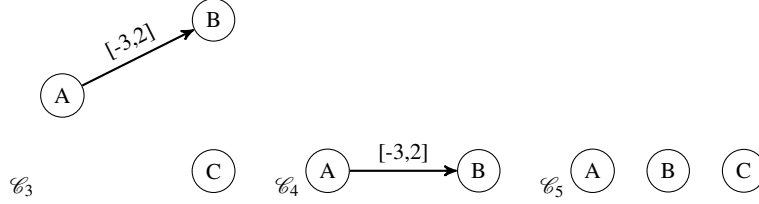

Fig. 3 Example of a chronicle specialization/generalization. $\mathscr{C}_4$ and $\mathscr{C}_5$ are generalizations of $\mathscr{C}_3$. $\mathscr{C}_4$ has a more general multiset and $\mathscr{C}_5$ has more general temporal constraints.

$\mathscr{C}_1 \neq \mathscr{C}_2$, we say that $\mathscr{C}_1$ is **more specific** than $\mathscr{C}_2$. On the contrary, $\mathscr{C}_2$ is **more general** than $\mathscr{C}_1$.

Example 4. Fig. 3 illustrates a chronicle $\mathscr{C}_3 = (\mathscr{E}_3, \mathscr{T}_3)$ and two more general chronicles $\mathscr{C}_4 = (\mathscr{E}_4, \mathscr{T}_4)$ and $\mathscr{C}_5 = (\mathscr{E}_5, \mathscr{T}_5)$. We have $\mathscr{C}_3 \preceq \mathscr{C}_4$ because $\mathscr{E}_4 \subset \mathscr{E}_3$ and $\mathscr{T}_3 = \mathscr{T}_4$. In the other hand we have $\mathscr{C}_3 \preceq \mathscr{C}_5$ because $\mathscr{E}_3 = \mathscr{E}_5$, but $\mathscr{T}_3$ has a more specific constraint than $\mathscr{T}_5$. In fact, $\mathscr{T}_5$ having non constraint, any additional constraint is more specific.

3.2 Chronicle support

Let $s = \langle (e_1, t_1), \ldots, (e_n, t_n) \rangle$ be a sequence and $\mathscr{C} = (\mathscr{E} = \{\!\{e'_1, \ldots e'_m\}\!\}, \mathscr{T})$ be a chronicle. An **occurrence** of $\mathscr{C}$ in s is a subsequence $\widetilde{s} = \langle (e_{f(1)}, t_{f(1)}), \ldots, (e_{f(m)}, t_{f(m)}) \rangle$ such that 1) $f : [1, m] \mapsto [1, n]$ is an injective function, 2) $\forall i,\, e'_i = e_{f(i)}$ and 3) $\forall i, j,\, t_{f(j)} - t_{f(i)} \in [a, b]$ where $e'_i[a, b]e'_j \in \mathscr{T}$. It is worth noting that f is not necessarily increasing. In fact, there is a difference between (i) the order of the chronicle multiset defined on items, $\leq_{\mathbb{E}}$, and (ii) the order of events in sequences, $\leq_{\mathbb{T}}$, defined on the temporal domain. The chronicle $\mathscr{C}$ **occurs** in s, denoted $\mathscr{C} \in s$, if and only if there is at least one occurrence of $\mathscr{C}$ in s. The **support** of a chronicle $\mathscr{C}$ in a sequence set $\mathscr{S}$ is the number of sequences in which $\mathscr{C}$ occurs: $supp(\mathscr{C}, \mathscr{S}) = |\{s \in \mathscr{S} \mid \mathscr{C} \in s\}|$. Given a minimal support threshold σ_{min}, a chronicle is **frequent** if and only if $supp(\mathscr{C}, \mathscr{S}) \geq \sigma_{min}$.

Example 5. Chronicle $\mathscr{C}$ (see Fig. 1 at the top left), occurs in sequences 1, 3 and 6 of Table 1. On the one hand, $\mathscr{C}$ does not occur in the other sequences because there

is no matching of $\mathcal{E} = \{\!\{A,B,C,C,D\}\!\}$, the multiset of $\mathscr{C}$, *i.e.*, there is strictly less than one occurrence of A or one of B or two of C or one of D. On the other hand, $\mathscr{C}$ occurs in sequences 1, 3 and 6 because they contain at least one matching of $\mathcal{E}$ satisfying the temporal constraints. For example in sequence 6, the only matching of $\mathcal{E}$ is the whole sequence. Then, B occurs 1 unit before A which satisfies the temporal constraint $A[-1,3]B$ because it corresponds to the lower bound of the interval. The first event C occurs 2 units before A which satisfies the constraint $A[-3,5]C$ because $-2 \in [-3,5]$. And the three other constraints $B[-2,2]C$, $C[1,3]C$ and $B[4,5]D$ are satisfied by the first C occurring 1 unit before B, the second C occurring 3 units after the first C and D occurring 5 units after B respectively.

We notice that there are two occurrences of $\mathscr{C}$ in sequence 1. Nonetheless, its support is $supp(\mathscr{C},\mathscr{S}) = 3$. This chronicle is frequent in $\mathscr{S}$ for any minimal support threshold σ_{min} lower or equal to 3. The two other chronicles, denoted $\mathscr{C}_1$ and $\mathscr{C}_2$, occur respectively in sequences 1 and 3; and in sequence 6. Their supports are $supp(\mathscr{C}_1,\mathscr{S}) = 2$ and $supp(\mathscr{C}_2,\mathscr{S}) = 1$.

The support of chronicles is anti-monotonic using the previously defined partial order $\preceq$. Indeed, for two chronicles $\mathscr{C}_1$ and $\mathscr{C}_2$ and a sequence set $\mathscr{S}$, $\mathscr{C}_1 \preceq \mathscr{C}_2$ implies $supp(\mathscr{C}_1,\mathscr{S}) \leq supp(\mathscr{C}_2,\mathscr{S})$. This property derives from the fact that the most specific chronicle $\mathscr{C}_1$ cannot occur in more sequences than $\mathscr{C}_2$ because it is more constrained. This property has been formally proved and used in previous works on chronicles [Dousson and Duong, 1999, Cram et al., 2012, Huang et al., 2012, Alvarez et al., 2013].

3.3 Discriminant chronicles mining

Let $\mathscr{S}^+$ and $\mathscr{S}^-$ be two sets of sequences, and $\sigma_{min} \in \mathbb{N}^+$, $g_{min} \in [1,\infty[$ be two user-defined parameters. A chronicle $\mathscr{C}$ is **discriminant** for $\mathscr{S}^+$ iff $supp(\mathscr{C},\mathscr{S}^+) \geq \sigma_{min}$ and $supp(\mathscr{C},\mathscr{S}^+) \geq g_{min} \times supp(\mathscr{C},\mathscr{S}^-)$. In other words, a chronicle is discriminant iff its **growth rate** $g(\mathscr{C},\mathscr{S})$ is above the given threshold g_{min}. To take into account the special case of a null support in the negative sequence set, the growth rate is defined as follows:

$$g(\mathscr{C},\mathscr{S}) = \begin{cases} \frac{supp(\mathscr{C},\mathscr{S}^+)}{supp(\mathscr{C},\mathscr{S}^-)} & \text{, if } supp(\mathscr{C},\mathscr{S}^-) > 0 \\ +\infty & \text{, otherwise.} \end{cases}$$

Contrary to the minimal support constraint which is anti-monotone, the minimal growth constraint is not.

Example 6. With chronicle $\mathscr{C}$ of Fig. 1, $supp(\mathscr{C},\mathscr{S}^+) = 2$, $supp(\mathscr{C},\mathscr{S}^-) = 1$, where $\mathscr{S}^+$ (resp. $\mathscr{S}^-$) is the sequence set of Table 1 labeled with $+$ (resp. $-$). Considering that $g(\mathscr{C},\mathscr{S}) = 2$, $\mathscr{C}$ is discriminant if $g_{min} \leq 2$. For chronicles $\mathscr{C}_1$ and $\mathscr{C}_2$, $supp(\mathscr{C}_1,\mathscr{S}^+) = 2$ and $supp(\mathscr{C}_1,\mathscr{S}^-) = 0$ so $g(\mathscr{C}_1,\mathscr{S}) = +\infty$ and $supp(\mathscr{C}_2,\mathscr{S}^+) = 0$ and $supp(\mathscr{C}_2,\mathscr{S}^-) = 1$ so $g(\mathscr{C}_2,\mathscr{S}) = 0$. $\mathscr{C}_2$ is not discriminant, but $\mathscr{C}_1$ is for any g_{min} value.

Algorithme 1 Algorithm *DCM* for discriminant chronicles mining

---

Require: $\mathscr{S}^+$, $\mathscr{S}^-$: sets of sequences, σ_{min}: minimal support threshold, g_{min}: minimal growth
   rate threshold
1: $\mathbb{M} \leftarrow$ EXTRACTMULTISETS$(\mathscr{S}^+, \sigma_{min})$ ▷ $\mathbb{M}$ is the set of frequent multisets
2: $\mathbb{C} \leftarrow \emptyset$ ▷ $\mathbb{C}$ is the set of discriminant chronicles
3: **for all** $ms \in \mathbb{M}$ **do**
4:     **if** $supp(\mathscr{S}^+, (ms, \mathscr{T}_\infty)) > g_{min} \times supp(\mathscr{S}^-, (ms, \mathscr{T}_\infty))$ **then**
5:         $\mathbb{C} \leftarrow \mathbb{C} \cup \{(ms, \mathscr{T}_\infty)\}$ ▷ Discriminant chronicle without temporal constraints
6:     **else**
7:         **for all** $\mathscr{T} \in$ EXTRACTDISCRCONSTRAINTS$(\mathscr{S}^+, \mathscr{S}^-, ms, g_{min}, \sigma_{min})$ **do**
8:             $\mathbb{C} \leftarrow \mathbb{C} \cup \{(ms, \mathscr{T})\}$ ▷ Add a new discriminant chronicle
9: **return** $\mathbb{C}$

The support constraint, using σ_{min}, prunes the infrequent, and so insignificant,
chronicles. For example, a chronicle like $\mathscr{C}_1$ such that $g(\mathscr{C}_1, \mathscr{S}) = +\infty$ but $supp(\mathscr{C}_1,$
$\mathscr{S}^+) = 2$ is discriminant but would not be interesting for a bigger sequence set
because it generalizes only two sequences. Pruning can be done efficiently thanks
to the anti-monotonicity of the support. More specifically, if a chronicle[2] $(\mathscr{E}, \mathscr{T}_\infty)$
is not frequent, then no chronicle of the form $(\mathscr{E}, \mathscr{T})$ will be frequent. This means
that temporal constraints may be extracted only for frequent multisets.

Extracting the complete set of discriminant chronicles is not interesting because
it may contain discriminant chronicles with same multiset and similar temporal
constraints. Such chronicles could match an almost similar set of sequences and
could be considered as redundant by the analyst. It is preferable to extract chroni-
cles whose temporal constraints are the most generalized. The approach proposed
in the next section efficiently extracts a set of meaningful discriminant chronicles
above given support and growth rate thresholds.

4 *DCM* algorithm

In this section, we present the algorithm *DCM* (Discriminant Chronicle Mining).
We first present the overall algorithm and then detail its main steps.

The *DCM* algorithm (see Algorithm 1) extracts discriminant chronicles in two
steps: First, it extracts the frequent multisets, which are chronicles without temporal
constraints, and then it mines discriminant temporal constraints from these multi-
sets.

At first, line 1 (EXTRACTMULTISETS) extracts $\mathbb{M}$, the frequent multiset set in
$\mathscr{S}^+$. This task is described in Section 4.1. In a second step, lines 3 to 8 extract
discriminant temporal constraints for each multiset. The naive approach would be
to extract discriminant temporal constraints for all frequent multisets. A multiset
$\mathscr{E}$ (*i.e.* a chronicle $(\mathscr{E}, \mathscr{T}_\infty)$), which is discriminant, may yield numerous similar
discriminant chronicles with most specific temporal constraints. We consider them

---

[2] $\mathscr{T}_\infty$ denotes the set of temporal constraints $\{e]-\infty, +\infty[e' \mid e, e' \in \mathscr{E}\}$.

as useless and, as a consequence, line 4 tests whether the multiset *ms* is discriminant. If so, $(ms, \mathcal{T}_\infty)$ is added to the discriminant patterns set without more specification of the temporal constraints. Otherwise, lines 7-8 generate chronicles from discriminant temporal constraints identified by the routine EXTRACTDISCRCONSTRAINTS. This routine is detailed in Section 4.2.

4.1 Multiset mining

This section elaborates on the frequent multiset mining. Compared to frequent itemset mining, it takes care of the cardinality of items *i.e.* its number of repetitions.

This task can be easily solved by applying a frequent regular itemset mining algorithm on a transaction set encoding multiple occurrences of a same item as several items. Such dataset used for multiset mining contains a transaction for each positive sequence in the initial temporal sequence set. For each transaction, an item $a \in \mathbb{E}$ occurring n times in a sequence is encoded by n items: $I_1^a, \dots, I_n^a$.

SID	Sequence	Items						
1	$(A,1), (B,3), (A,4), (C,5), (C,6), (D,7)$	I_1^A I_2^A I_1^B			I_1^C I_2^C I_1^D			
2	$(B,2), (D,4), (A,5), (C,7)$	I_1^A		I_1^B	I_1^C	I_1^D		
3	$(A,1), (B,4), (C,5), (B,6), (C,8), (D,9)$	I_1^A		I_1^B I_2^B I_1^C I_2^C I_1^D				

Table 2 The transaction set (last column of the table) obtained from the positive sequences of the dataset of Table 1 to extract frequent multisets.

Example 7. Table 2 shows the transaction set obtained for the set of temporal sequences of Table 1 (page 6). The transactions correspond only to the positive sequences. Each transaction contains the same number of items as the number of events in the corresponding sequence. In first transaction, the two occurrences of the events of type A became two distinct items I_1^A and I_2^A.

A frequent itemset of size m, $(I_{i_k}^{e_k})_{1 \leq k \leq m}$, extracted from this dataset is transformed into the multiset containing, i_k occurrences of the event e_k. Frequent itemsets with two items $I_{i_k}^{e_k}$, $I_{i_l}^{e_l}$ such that $e_k = e_l$ and $i_k \neq i_l$ are redundant and, thus, these itemsets are ignored to avoid generating several times the same multiset. This could be tackled by adding pattern constraints into the itemset mining algorithm. Practically, we implemented a post-processing step.

Example 8. For $\sigma_{min} = 2$, the itemset $\{I_1^A, I_1^B, I_1^C\}$ is frequent in the dataset of the Table 2 since it occurs in each transaction. Then, the extracted itemset is translated into the multiset $\{\!\{A, B, C\}\!\}$ because I_1^A, I_1^B and I_1^C correspond to one occurrence of A, one occurrence of B and one occurrence of C respectively. Another frequent itemset is $\{I_1^A, I_1^B, I_2^C\}$. This itemset is frequent for $\sigma_{min} = 2$ because it occurs in the

SID	$\mathcal{A}_{A \to B}$	$\mathcal{A}_{B \to C}$	$\mathcal{A}_{A \to C}$	Label
1	2	2	4	+
1	2	3	5	+
1	−1	2	1	+
1	−1	3	2	+
2	−3	5	2	+
3	3	1	4	+
3	3	4	7	+
3	5	2	7	+
3	5	−1	4	+
5	−2	3	1	−
6	−1	−1	−2	−
6	−1	2	1	−

Table 3 Relational dataset for the multiset $\{\!\{A,B,C\}\!\}$ corresponding to its occurrences in Table 1.

two transactions 1 and 3. It can be translated as the multiset $\{\!\{A,B,C,C\}\!\}$ because I_2^C corresponds to two occurrences of C. To avoid redundancy, itemsets containing two items referring to the same event will be ignored. An example of such itemsets is $\{I_1^A, I_1^B, I_1^C, I_2^C\}$. This itemset is ignored because it is equivalent to $\{I_1^A, I_1^B, I_2^C\}$: containing I_1^C and I_2^C together is equivalent to containing I_2^C alone.

4.2 Temporal constraints mining

The general idea of EXTRACTDISCCONSTRAINTS is to extract discriminant temporal constraints using a standard numerical rule learning algorithm [Cohen, 1995].

Let $\mathcal{E} = \{\!\{e_1 \ldots e_n\}\!\}$ be a frequent multiset. A relational[3] dataset, denoted $\mathcal{D}$, is generated with all occurrences of $\mathcal{E}$ in $\mathcal{S}$. As $\mathcal{C} = (\mathcal{E}, \mathcal{T}) \preceq (\mathcal{E}, \mathcal{T}_\infty)$, $\mathcal{D}$ is sufficient to describe the entire occurrences set of $\mathcal{C}$. The numerical attributes of $\mathcal{D}$ are inter-event duration between each pair (e_i, e_j) where $e_i, e_j \in \mathcal{E}$ and $e_i \leq_\mathbb{E} e_j$, denoted by $\mathcal{A}_{e_i \to e_j}$. An example is labeled by its sequence label ($L \in \mathbb{L}$). If a sequence has several occurrences of $\mathcal{E}$, then each occurrence yields one example.

A rule learning algorithm induces numerical rules from $\mathcal{D}$. A rule has a label in the conclusion and its premise is a conjunction of conditions on attribute values. Conditions are inequalities in the form: $\mathcal{A}_{e_i \to e_j} \geq x \wedge \mathcal{A}_{e_i \to e_j} \leq y$, where $(x, y) \in \overline{\mathbb{R}}^2$. Such a rule is translated as a temporal constraint set, $\mathcal{T} = \{e_i[x,y]e_j\}$. The couple $(\mathcal{E}, \mathcal{T})$ is then a potential discriminant chronicle. At this stage, we are not sure that the chronicle is discriminant. In fact, the rule learning algorithm extracts discriminant temporal constraints based on the dataset $\mathcal{D}$, but multiple instances of the multiset in a sequence may bias the support evaluation. This limitation is discussed in section 4.2.2.

---

[3] In some context, relational dataset designates a dataset whose organization is based on a relational data model. In our context, the relational model designed an attribute-value dataset by opposition to a sequence dataset. It can be seen as a single relation dataset.

Example 9. Table 3 is the relational dataset obtained from the occurrences of $\{A, B, C\}$ containing by the dataset of Table 1. The attribute $\mathscr{A}_{A \to B}$ denotes the duration between A and B. We can see on Table 3 that an example contains the *SID* of the sequence, the duration for each pair of events and the label of the sequence. It is worth noticing that several examples may come from the same sequence. The rule $\mathscr{A}_{A \to C} \geq 2 \implies +$ characterizes almost all the examples labeled by $+$ in Table 3 and characterizes all the different *SID* labeled by $+$. It is translated into the discriminant temporal constraints $\{A[2, +\infty[C\}$ which gives the discriminant chronicle $\mathscr{C} = (\{e_1 = A, e_2 = B, e_3 = C\}, \{e_1[2, +\infty[e_3\})$.

It is interesting to notice that the discriminant temporal constraints $\{A[-1, +\infty[B, B[3, +\infty[C\}$ is less discriminant from the point of view of the dataset of Table 3 *i.e.*, these constraints cover less positive occurrences, but it discriminates perfectly the sequences, as it occurs in all positive sequences (at least for one multiset instance) but it occurs in none of the negative sequences. This illustrates the possible biases due to multiple instances.

4.2.1 Rule learning

The rule learning task is done in practice by the $Ripper_k$ algorithm [Cohen, 1995], a relational rule learning algorithm. This algorithm was selected because of its high accuracy and because it allows the extraction of unordered rules. An unordered rule is independent from the others. Thereby, each rule of an unordered rule set is still true if read alone. It is not true for ordered rules. The problem with ordered rules learners (*e.g.*, CN2, C4.5) comes with the validity of a rule at position n. This rule is valid only if the previous $n-1$ rules are not valid. Thereby if a chronicle set is based on ordered rules, some chronicles may not be discriminant if read alone.

$Ripper_k$ splits the dataset in two parts: *Grow* and *Prune*. *Grow* is used to construct the conjunction of conditions discriminating examples of one class with respect to the others. Growing stops when it is no longer possible to add a condition to the conjunction that improves accuracy. *Prune* is used to prune the constructed rule. If the conjunction is more discriminating without its last condition, this rule is removed from the conjunction and the last new condition is tested until the accuracy can no longer be improved. This approach allows to manage the overfitting made on *Grow*. If the accuracy is not satisfied for this conjunction of conditions, the search stops for that label. Otherwise the conjunction is returned as a rule, the examples associated with it are removed from *Grow* and the search starts again. The aforementioned steps are repeated for each label of the dataset.

The choice of using a rule learning algorithm based on an incomplete heuristic, in this case a heuristic based on the MDL principle (*Minimum Description Length*), is required for computational reasons. However, $Ripper_k$ combines (1) a reasonable algorithmic complexity – the computation times remain reasonable, (2) good classification performance – the chronicles extracted are therefore well representative of

the dataset – and (3) reduced rule sets – retrieved chains remain easily interpretable [Lattner et al., 2003].

In the following section, we detailed two limitations of the proposed algorithm.

4.2.2 Multiple instance limit

The multiple instance problem [Foulds and Frank, 2010] generally refers to the counting problems, which happen when a pattern appears multiple times in an object. In our case, an object is a sequence. This situation is encountered when building the dataset (*cf.* Table 3) and distorts the count of the number of objects validating a conjunction of rules made by $Ripper_k$. When examples with the same *SID* (*e.g.*, first four sequences of the table 3) are covered by a rule, they must only count for one object (*i.e.* a *SID*). This specific setting in not handled by $Ripper_k$.

Our solution is to post-process the conjunctions of $Ripper_k$ rules and to retain only those that effectively validate the minimum support and growth rate constraints. This solution does not ensure that discriminant chronicles are extracted *i.e.*, all extracted chronicles could be pruned but does not add computational heaviness.

A second solution to this problem would be to modify the counting system of $Ripper_k$ so that it evaluates a conjunction of conditions by counting the distinct *SID*. But the heuristic of $Ripper_k$ is not made to exclude examples after having selected others, so it is not efficient for datasets containing many examples with same *SID*.

The third solution would be to use methods adapted to a multiple instance setting. For example, the miSVM approach [Doran and Ray, 2014] identifies for each object which instance has to be used by the classifier. This instance is seen as a witness of the object class. One possible approach for our algorithm would be to preprocess the dataset using the miSVM in order to identify the witness instances. In such a case, $Ripper_k$ would be applied on a mono-instance dataset. This solution seems to be the most reliable but it adds an important complexity to the process. Besides, our first experiments shown that it removes too many patterns.

Taking into account multiple instances directly in the rule learning step would therefore necessarily be done with adding computational complexity. Since multiple instances are rarely encountered in our applications, we have focused on computational efficiency by correcting the results with the post-processing strategy.

4.2.3 Minimal growth rate constraint satisfaction

During the post-processing of the rules for the previous problem, the algorithm prunes the rules according to the growth rate threshold. Indeed, the use of the g_{min} parameter as growth rate threshold in $Ripper_k$ may prune interesting rules. Errors can occur when the *Grow* and *Prune* sets are not balanced and the proportion of the examples in a dataset is not representative of the set. On the other hand, the iterative process of rule conjunction generation removes positive and negative examples from

Grow. Thereby, if too many negative examples are removed from *Grow*, the growth rate of a rule is true for *Prune* but not for *Grow*. However, this problem remains marginal since it was already taken into account in the $Ripper_k$ development.

To conclude this section, the implementation of *DCM* based on $Ripper_k$ is correct because of the pruning step, but it is not complete. The algorithm benefits from its incompleteness since the extracted pattern set is smaller and then easier to present to a user and is extracted quicker.

5 Decision based on discriminant chronicle sets

Once discriminant chronicles have been extracted, they can be used to label new sequences. This section introduces this decision process.

The decision based on the extracted discriminant chronicles is not the main idea of this paper, however, we need to describe the general principles since we use it as evaluation. A discriminant chronicle extracted for a class can be seen as a rule for which an occurrence of this chronicle implies the associated label. We can so extract rules of the form $\mu(\mathscr{C},s) \implies l_i$ for each class where $\mathscr{C}$ is a chronicle, s a sequence, μ the mapping function determining if $\mathscr{C}$ occurs in s and l_i the label of the ith class. The discriminant chronicle mining task extracts patterns that occur also in other classes that those which are initially discriminated. Moreover, two extracted discriminant chronicles could have been extracted for different growth rates and supports. Therefore it is possible to have to predict the label of a sequence s, the two rules $\mu(\mathscr{C},s) \implies l_i$ and $\mu(\mathscr{C}',s) \implies l_j$ where $l_i \neq l_j$ and $\mu(\mathscr{C},s) = \mu(\mathscr{C}',s) = 1$. In such case we have to apply an order on the extracted rule set to take a decision.

To rank the chronicles, we choose to use a measure based on the same growth rate definition that we use to extract them. Discriminant chronicles can have different supports and in the case of two discriminant chronicles $\mathscr{C}_1$ and $\mathscr{C}_2$ extracted for l_1 and l_2 respectively, $supp(\mathscr{C}_1, l_1) = 10$, $supp(\mathscr{C}_1, \overline{l_1}) = 0$, $supp(\mathscr{C}_2, l_2) = 100$ and $supp(\mathscr{C}_2, \overline{l_2}) = 1$, we prefer to give more confidence to $\mathscr{C}_2$ than to $\mathscr{C}_1$ even if $\mathscr{C}_1$ has a better growth rate. The reason for this preference is due to a wider covering of $\mathscr{C}_2$ and, thereby, a lower sensibility to the false positive rate. We choose to recompute the growth rate of each chronicle with adding a virtual sequence to the negative support. This measure is actually used in the subgroup discovery domain and is presented in [Herrera et al., 2011] as the precision measure $Q_g(R) = \frac{TP}{FP+g}$ where, in our case, $TP = supp(\mathscr{C}, l_i)$, $FP = supp(\mathscr{C}, \overline{l_i})$ and g is set up to 1.

6 Benchmark

This section is dedicated to evaluate the efficiency of our algorithm on synthetic and real data. The real datasets have been used for other pattern-based classification

approaches [Fradkin and Mörchen, 2015, Bornemann et al., 2016] and are different from our case study. The *DCM* implementation is written in C++ and relies on preexisting implementations of *LCM* [Uno et al., 2004] and *Ripper$_k$* [Cohen, 1995].

6.1 Synthetic data

In this section, we show the results of the evaluation of DCM on synthetic data.

6.1.1 Dataset generation

The general principle of the simulator is to generate sequences based on two different chronicles $\mathscr{C}^+$ and $\mathscr{C}^-$. In a first step, one of these chronicles is assigned to every sequence of the two datasets $\mathscr{S}^+$ and $\mathscr{S}^-$ such that $\mathscr{C}^+$ (resp. $\mathscr{C}^-$) is introduced mainly in the dataset $\mathscr{S}^+$ (resp. $\mathscr{S}^-$) (see figure 4). In a second step, the chronicle assigned to the sequence is used as a "template": the sequence holds the items of the chronicle with timestamps randomly drawn but satisfying the temporal constraints, and additional random items.
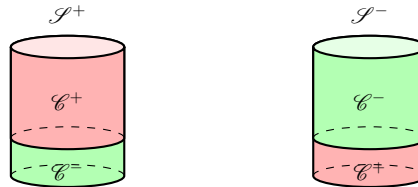

Fig. 4 Distribution of chronicles $\mathscr{C}^+$ and $\mathscr{C}^-$ in sequence datasets $\mathscr{S}^+$ and $\mathscr{S}^-$.

The dataset $\mathscr{S} = \mathscr{S}^+ \cup \mathscr{S}^-$ contains labeled sequences ($\mathbb{L} = \{+, -\}$). Sequences labeled with $+$ (resp. $-$) are then characterized by $\mathscr{C}^+$ (resp. $\mathscr{C}^-$). Two parameters manage the introduction of these chronicles: $\widehat{f}_{min}$ and $\widehat{g}_{min}$.

- The parameter $\widehat{f}_{min}$ is the percentage of sequences in the majority class in which a chronicle appears.
- The parameter $\widehat{g}_{min}$ is the ratio between the sequence numbers of the two classes in which this chronicle appears.

The objective of the experiment is to evaluate the ability of *DCM* to extract correct discriminant temporal constraints. To this end, we use chronicles for $\mathscr{C}^+$ and $\mathscr{C}^-$ that differ only on one temporal constraints. The more similar the chronicles, the harder the mining task.

More precisely, the *Base$_N$* datasets have been created from chronicles based on the same multiset $\{\!\{A,B\}\!\}$ of size 2. Therefore each chronicle has a unique time interval. For each dataset, the intervals of $\mathscr{C}^-$ and $\mathscr{C}^+$ overlap more or less. For all these

datasets, we set $\mathscr{C}^+ = (\{\!\{A,B\}\!\}, \{A[3,10]B\})$; we set $\mathscr{C}^- = (\{\!\{A,B\}\!\}, \{A[7,8]B\})$ to generate $Base_1$. $Base_2$, $Base_3$, $Base_4$ and $Base_5$ are generated using similar constraints for chronicles $\mathscr{C}^-$: respectively, $A[6,8]B$, $A[4,8]B$, $A[4,9]B$ and $A[4,10]B$. These chronicles are illustrated in the Table 4.

Dataset	$\mathscr{C}^+$	$\mathscr{C}^-$	Expected discriminant chronicles	
$Base_1$	$A \xrightarrow{[3,10]} B$	$A \xrightarrow{[7,8]} B$	$A \xrightarrow{[-\infty,6]} B$	$A \xrightarrow{[9,+\infty]} B$
$Base_2$	$A \xrightarrow{[3,10]} B$	$A \xrightarrow{[6,8]} B$	$A \xrightarrow{[-\infty,5]} B$	$A \xrightarrow{[9,+\infty]} B$
$Base_3$	$A \xrightarrow{[3,10]} B$	$A \xrightarrow{[4,8]} B$	$A \xrightarrow{[-\infty,3]} B$	$A \xrightarrow{[9,+\infty]} B$
$Base_4$	$A \xrightarrow{[3,10]} B$	$A \xrightarrow{[4,9]} B$	$A \xrightarrow{[-\infty,3]} B$	$A \xrightarrow{[10,+\infty]} B$
$Base_5$	$A \xrightarrow{[3,10]} B$	$A \xrightarrow{[4,10]} B$	$A \xrightarrow{[-\infty,3]} B$	

Table 4 Chronicles introduced in each dataset type and expected discriminant chronicles for $\mathscr{S}^+$ in these datasets for each pair of introduced chronicles. As discriminant chronicles of $Base_1$ are more general than those of $Base_5$, $Base_1$ sequences are easier to discriminate than those of $Base_5$.

It is important to notice that the discriminant mining task does not aim at extracting $\mathscr{C}^+$ and $\mathscr{C}^-$ from $\mathscr{S}$. Its goal is to extract patterns describing the maximum number of occurrences of $\mathscr{C}^+$ and the minimum number of occurrences of $\mathscr{C}^-$. The third column of Table 4 gives the expected discriminant chronicles to extract.

For each dataset type $Base_N$, the results are averaged over 20 datasets generated independently. The simulator is set to generate by default datasets containing 800 sequences. The average length of sequences is 10. The size of the vocabulary is set at 50 items and $\widehat{f}_{min}$ and $\widehat{g}_{min}$ are respectively set to 80% and 10%.

The ability of *DCM* to extract the correct patterns is evaluated by comparing the sets of extracted patterns with the expected patterns. Two measures are used: $\Delta c_{(m,\widehat{m})}$ the coverage ratio and $\Delta g_{(m,\widehat{m})}$ the growth ratio. The variable m and $\widehat{m}$ denote extracted and expected patterns respectively.

- $\Delta c_{(m,\widehat{m})} = \dfrac{|\{s \in \mathscr{S}^+ \mid m \in s\} \cap \{s \in \mathscr{S}^+ \mid \widehat{m} \in s\}|}{|\{s \in \mathscr{S}^+ \mid \widehat{m} \in s\}|}$ is a **coverage ratio** similar to the unilateral Jaccard similarity [Santisteban and Tejada-Cárcamo, 2015] coefficient. It represents the ratio between the number of sequences of the $\mathscr{S}^+$ class containing one occurrence shared by m and $\widehat{m}$; and the number of sequences in $\mathscr{S}^+$ containing $\widehat{m}$. This measure determines how much similar are the sets of sequences labeled as positive by the two patterns. It is a recall measure.

- $\Delta g_{(m,\widehat{m})} = \begin{cases} min\left(\dfrac{g_{(m,\mathscr{S})}}{g_{(\widehat{m},\mathscr{S})}}, 1\right) & \text{, if } g_{(\widehat{m},\mathscr{S})} < +\infty \\ 1 & \text{, if } g_{(m,\mathscr{S})} = g_{(\widehat{m},\mathscr{S})} = +\infty \\ 0 & \text{otherwise} \end{cases}$

is the **growth ratio**. It represents the ratio of the growth rates of m to $\widehat{m}$.

For each expected patterns $\widehat{m}$, a single pair $(\Delta c_{(m,\widehat{m})}, \Delta g_{(m,\widehat{m})})$ is retained such that $\Delta c_{(m,\widehat{m})}$ is the highest for any extracted pattern m and that $\Delta g_{(m,\widehat{m})}$ is the highest if there are several maximal $\Delta c_{(m,\widehat{m})}$. To simplify the notations, $(\Delta c, \Delta g)$ is used to denote this pair. In addition to the previous measure used as recall, we use this measure as accuracy.

6.1.2 Results

Figure 5 shows the extraction results of *DCM* on the 20 datasets generated for each *Base$_N$*. The coverage ratio is given in abscissa and the growth ratio in ordinates. Given a coverage ratio Δg and a growth ratio Δg, the dot size gives the number of chronicles similar to one of the expected chronicles (see Table 4). This number is the sum of patterns for the 20 datasets, meaning that the maximal dot size is 40 for database 1 to 4 (20 datasets with 2 expected patterns) and 20 for database 5.

The biggest dots per database, from 1 to 5, represent respectively 31, 40, 30, 30 and 15 extracted chronicles that was expected.

We notice on Figure 5 that *DCM* extracts at least 75% of the expected discriminant patterns in each dataset a discriminant pattern whose coverage and growth ratios are perfect ($\Delta c = 1$ and $\Delta g = 1$). In addition, for the *Base$_4$* and *Base$_5$* datasets, *DCM* does not extract any patterns for certain introduced discriminant patterns, but only 1 out of 40 for *Base$_4$* and 2 on 20 for *Base$_5$*.

To explain these errors, we extract chronicles from the same dataset with a minimum growth rate $g_{min} = 1.1$. With this setting, all expected *Base$_4$* patterns match with at least extracted pattern (with at least $\Delta c = 1$ and $\Delta g = 0.83$). It is worth noting that the extracted pattern corresponding to $\Delta c = 0.47$ for the first extraction of the *Base*4 patterns reaches the same Δc and Δg ratios ($\Delta c = 1$ and $\Delta g = 0.83$).

These results on simple synthetic data show that *DCM* actually extracts the discriminating patterns with robustness. For datasets that can only be discriminated through the temporal dimension, *DCM* extracts the useful discriminant temporal constraints representing the expected discriminant patterns.

6.2 BIDE-D comparison

We compare *DCM* to the *BIDE-D* algorithms on datasets from [Fradkin and Mörchen, 2015]. These datasets come from a variety of applications. In order to
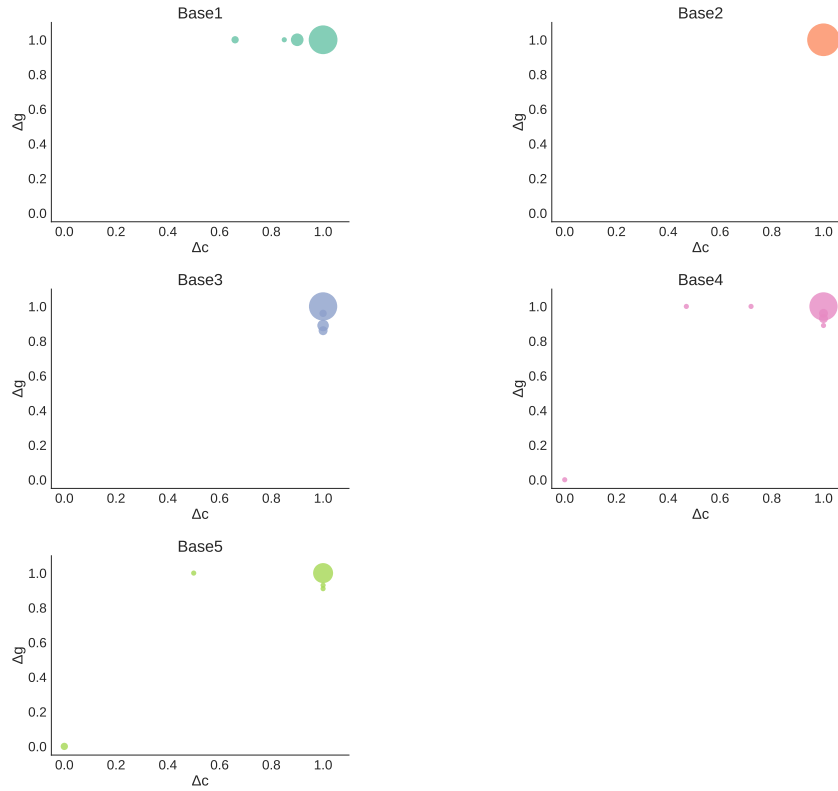
Fig. 5 Scatter plots of the extracted patterns represented by $(\Delta c, \Delta g)$ for each dataset. For each expected discriminant pattern $\widehat{m}$ only the best extracted pattern m according to $(\Delta c_{(m,\widehat{m})}, \Delta g_{(m,\widehat{m})})$ is used in these plots. The dot size represents the number of extracted patterns.

compare *DCM* with *BIDE-D* because these datasets are neither as simple as *blocks* nor as *Auslan*2, we chose to focus on *asl-bu*, *asl-gt* and *context*.

- **asl-bu**: The intervals are transcriptions from videos of American Sign Language expressions provided by Boston University [Papapetrou et al., 2005]. It consists of observation interval sequences with labels such as *head mvmt: nod rapid* or *shoulders forward* that belong to one of 7 classes like *yes-no question* or *rhetorical question*.
- **asl-gt**: The intervals are derived from 16-dimensional numerical time series with features derived from videos of American Sign Language expressions [Starner et al., 1998]. The numerical time series are discretized into $2-4$ states. Each sequence represents one of the 40 words such as *brown* or *fish*.
- **context**: The intervals are derived from categoric and numeric data describing the context of a mobile device carried by humans in different situations [Mäntyjärvi et al., 2004]. Numeric sensors are discretized using $2-3$ bins chosen manually

based on exploratory data analysis. Each sequence represents one of five scenarios such as *street* or *meeting*.

These datasets are databases of intervals interpreted as sequential databases by considering begin and end boundaries of an interval as two distinct events.

The couple $\langle \mathscr{C}, L \rangle$ of discriminant chronicles, where $L \in \mathbb{L}$, is used to predict the label of a sequence. A label L will be predicted for a sequence in case this sequence contains the chronicle $\mathscr{C}$. In the case of several chronicles appearing in the sequence, we label the sequence with the predicted label by the chronicle with the highest growth rate. This simple classification method is more accurate than the use of a standard machine learning classifier to evaluate the extracted chronicles. On the other hand, discriminant chronicle accuracy would not be overrated in comparison to *BIDE-D* approaches. Indeed, the *BIDE-D* approaches use a *SVM* classifier and *SVM* is generally more accurate than a simple rule-based classifier.

The results discussed below are presented in Table 5. We set a size limit of 6 items per chronicle in order to limit the number of patterns. The results presented in [Fradkin and Mörchen, 2015] are given in Table 6.

dataset	σ_{min}	g_{min} 2	3	4	5
asl-bu	**0.4**	52.64 (±2.21)	49.20 (±0.51)	52.41 (±3.51)	48.28 (±1.99)
	0.5	51.26 (±1.31)	51.72 (±1.99)	49.66 (±6.82)	48.51 (±5.60)
	0.6	51.72 (±3.35)	50.34 (±3.58)	44.14 (±3.31)	39.08 (±3.72)
asl-gt	**0.2**	31.55 (±0.91)	31.61 (±0.92)	30.20 (±1.82)	30.15 (±0.92)
	0.3	31.17 (±0.44)	29.18 (±1.53)	27.75 (±1.58)	26.96 (±1.89)
	0.4	27.34 (±2.10)	25.82 (±0.42)	25.91 (±0.12)	25.32 (±0.19)
	0.5	25.44 (±0.34)	25.20 (±0.13)	24.68 (±0.50)	24.12 (±0.41)
	0.6	24.30 (±0.42)	23.92 (±0.53)	23.89 (±0.52)	23.13 (±0.44)
context	**0.2**	64.78 (±2.83)	57.39 (±4.76)	46.09 (±3.89)	53.48 (±6.07)
	0.3	56.09 (±5.83)	42.61 (±7.62)	52.61 (±3.22)	36.96 (±7.53)
	0.4	47.83 (±4.07)	39.57 (±3.57)	50.43 (±5.41)	47.39 (±4.96)
	0.5	53.91 (±4.46)	38.70 (±0.97)	30.43 (±5.10)	47.83 (±7.37)
	0.6	50.87 (±2.48)	34.78 (±4.35)	30.87 (±4.18)	28.70 (±4.46)

Table 5 Mean accuracy (in %) with respect to σ_{min} (minimal support threshold), g_{min} (minimal growth rate threshold) for different datasets.

On *asl-gt BIDE-D* outperforms *DCM*. Where accuracy ranges from 27.31% for $\sigma_{min} = 0.6$ to 82.94% for $\sigma_{min} = 0.2$ for *BIDE-D*. But it hardly exceeds 30% for discriminant chronicles. The standard deviation of *BIDE-D* and *DCM* are similar. Many patterns are extracted but are poorly distributed among the dataset labels. While discriminant chronicles can represent discriminant sequential patterns, these poor results on *asl-gt* are due to a lack of discriminant patterns. This lack is due to the large number of classes that are 40 for *asl-gt*. As a result, a single pattern is

dataset	σ_{min}	BIDE-C	BIDE	BIDE-D (k=90)	BIDE-DC (k=90)
asl-bu	**0.2**	60.01 ($\pm$ 5.47)	59.71 ($\pm$ 4.87)	52.91 ($\pm$ 4.15)	56.01 ($\pm$ 4.62)
	0.3	58.05 ($\pm$ 4.90)	58.50 ($\pm$ 4.40)	52.83 ($\pm$ 4.03)	57.89 ($\pm$ 5.00)
	0.4	57.74 ($\pm$ 4.03)	58.42 ($\pm$ 4.15)	53.06 ($\pm$ 3.91)	59.03 ($\pm$ 5.09)
	0.5	57.74 ($\pm$ 4.03)	58.42 ($\pm$ 4.15)	53.06 ($\pm$ 3.91)	59.03 ($\pm$ 5.09)
	0.6	55.25 ($\pm$ 5.58)	58.80 ($\pm$ 4.37)	53.44 ($\pm$ 4.18)	57.21 ($\pm$ 5.40)
asl-gt	**0.2**		82.78 ($\pm$ 1.31)	82.94 ($\pm$ 1.07)	82.30 ($\pm$ 1.52)
	0.3		73.16 ($\pm$ 1.33)	73.14 ($\pm$ 1.43)	73.17 ($\pm$ 1.32)
	0.4		53.38 ($\pm$ 1.11)	53.40 ($\pm$ 1.11)	53.36 ($\pm$ 1.05)
	0.5		33.67 ($\pm$ 1.43)	33.70 ($\pm$ 1.41)	33.68 ($\pm$ 1.43)
	0.6		27.31 ($\pm$ 0.72)	27.31 ($\pm$ 0.72)	27.31 ($\pm$ 0.72)
context	**0.7**		53.47 ($\pm$ 6.80)	37.36 ($\pm$ 3.73)	51.53 ($\pm$ 6.93)
	0.8		48.75 ($\pm$ 7.29)	48.89 ($\pm$ 6.34)	47.92 ($\pm$ 5.94)
	0.9		26.25 ($\pm$ 4.08)	27.08 ($\pm$ 3.78)	26.53 ($\pm$ 4.27)

Table 6 Mean accuracy (in %) with respect to σ_{min} (minimal support threshold) and the different methods proposed in [Fradkin and Mörchen, 2015] for different datasets. The methods *BIDE-D* and *BIDE-DC* extract discriminant sequential patterns. *BIDE* and *BIDE-C* extract frequent sequential patterns. Support thresholds used to obtain results from context are much higher than for the other datasets.

difficulty discriminant. The results obtained for *BIDE-D* are due to the selection of the best discriminant patterns and not to a minimal growth rate threshold. Thereby, even if there is no discriminant sequential patterns in *asl-gt*, the *BIDE-D* approaches extract some patterns to use in the classifier.

On *asl-bu DCM* performs slightly worse than *BIDE-D*. The accuracy using $g_{min} = 2$ is equivalent from $\sigma_{min} = 0.4$ to $\sigma_{min} = 0.6$, whereas the number of extracted patterns is reduced from more than $30,000$ to $1,600$. DCM's accuracy exhibits better performances than *BIDE-D* in terms of standard deviation. We can notice that *asl-bu* is a difficult type of dataset for chronicle mining due to multiple occurrences of the same event types. As a result *DCM* did not extract any discriminant patterns for $\sigma_{min} = 0.2$ or $\sigma_{min} = 0.3$. It takes too much time and does not finish for these parameters within the 12 hour timeout.

Finally, on *context*, DCM outperforms *BIDE-D* in accuracy. The accuracy standard deviation is higher than for the two other datasets but is, however, similar to the one of *BIDE-D*. We note that the low results of *BIDE-D* approaches may be due to the use of a different minimum support threshold strategy for this dataset. The difference in the number of patterns extracted by the $\sigma_{min} = 0.2$ or $\sigma_{min} = 0.6$ parameters is much smaller than for *asl-bu*. The count goes from 360 patterns for $\sigma_{min} = 0.2$ and $g_{min} = 2$ to 145 for $\sigma_{min} = 0.6$ and $g_{min} = 4$.

We conclude from these results that discriminant chronicles as rules can produce similar and sometimes better accuracy than sequential patterns as features. Furthermore, our results show that it seems easier to obtain a good accuracy for datasets containing few classes (*e.g.*, *context*: 5 classes) than datasets for which the

sequences are distributed in a large set of classes (*e.g.*, *asl-gt*: 40 classes). Indeed, a large class number implies a generalization problem. A pattern discriminating a class against 39 other classes has to be more specific than a pattern discriminating a class against 4 other classes. This generalization problem comes from the unbalanced distribution between positive and negative sequences that requires to be very specific to obtain discriminant patterns. This problem is less important for the *BIDE-D* approaches because the *SVM* classifier learns combinations between discriminant patterns extracted for a subset of classes to generalize it for all classes. Thereby, the *SVM* classifier can predict a class without using discriminant pattern for this class but it is not possible with discriminant chronicles used as rules. The definition of an extracted discriminant pattern is also different for *DCM* and *BIDE-D* and affects this generalization problem. A pattern is extracted by *DCM* if its growth rate is greater than a threshold. In this case, it is possible to extract no patterns for a threshold. A pattern is extracted by *BIDE-D* if its growth rate is one of the highest. Thereby, it is ensured that some patterns will be extracted and used as features by the classifier. Finally, *DCM* is easy to set up while a cross-validation is needed to obtain the parameters of the *SVM* that use the sequential patterns as features (the standard *SVM* parameter C is a power of 10 and is tested from 10^{-3} to 10^3).

7 Case study of care pathway analytics

A care pathway designates the sequences of interactions of a patient with the healthcare system (medical procedures, biology analysis, drug deliveries, etc.). Healthcare systems collect longitudinal data about patients to manage their reimbursements. Such data are now readily available in medico-administrative databases. Huge databases, such as the SNIIRAM [Moulis et al., 2015] in France, have a better population coverage than *ad hoc* cohorts and Electronic Medical Records (EMR). Moreover it covers cares in real life situations over a long period of several years.

Medico-administrative databases are useful to answer questions about care quality improvement (*e.g.*, care practice analysis), care costs cutting and prediction, or epidemiological studies. Among epidemiological studies, pharmaco-epidemiological studies answer questions about the uses of health products, drugs or medical devices, on a real population.

The main difficulties encountered by pharmaco-epidemiology studies are that 1) epidemiologists often have to provide an hypothesis to assess, 2) they cannot handle sequences with more than two events and 3) the temporal dimension of pathways is poorly exploited.

Care pathway analytics aim at supporting epidemiologists to explore such complex information sources. It consists of using data analytics tools (visualization, data abstraction, machine learning algorithms, etc.) applied on healthcare data. More specifically, temporal pattern mining algorithms can extract interesting sequences of care events that would suggest candidate hypotheses. As the pharmaco-

epidemiological studies consist of studying care pathways with respect to the patient outcome, discriminant temporal patterns are appropriate for such analyses.

This section[4] presents the use of *DCM* to study care pathways of epileptic patients. Recent studies suggested that medication substitution (so called switch) may be associated with epileptic seizures for patients with long-term treatment with anti-epileptic medication. In [Polard et al., 2015], the authors did not find significant statistical relationship between brand-to-generic substitution and seizure-related hospitalization. The *DCM* algorithm is used to extract patterns of drug deliveries that discriminate occurrences of recent seizures. These patterns may be interesting for further investigations by statistical analysis.

7.1 Positive and negative datasets construction

This section presents the transformation of medical database into a dataset of labeled sequences. This transformation is required to apply the *DCM* algorithm and can impact the results as well as their interpretability. In fact, depending on the input data (event types, sequence lengths, etc.), the extracted patterns may change a lot. We tested several sequence encodings and the one proposed below appears to be the most interesting.

Our dataset was obtained from the SNIIRAM [Moulis et al., 2015] database which contains a lot of tables with large numbers of attributes. Two important choices are required to obtain datasets of labeled sequences: 1) the choice of the set of events on which sequences will be built and 2) the sequence labeling.

The care pathway of a patient is a collection of timestamped drugs deliveries from 2009 to 2011. All other care events have been ignored in this study. For each drug delivery, an event id is a tuple $\langle m, grp, g \rangle$ where m is the ATC[5] code of the active molecule, $g \in \{0, 1\}$ where $g = 1$ means generic drug and $g = 0$ brand-named drug and grp is the speciality group. The speciality group identifies the drug presentation (international non-proprietary name, strength per unit, number of units per pack and dosage form).

A case-crossover protocol has been adapted to label sequences. This protocol is often used in pharmaco-epidemilogical studies. It studies the possible causes of an outcome using a patient as his/her own control. In our case, the patient outcomes are seizure events of epileptic patients. According to the study of Polard et *al.* [Polard et al., 2015], we selected epileptic patients with stable treatments, *i.e.*, patients who had at least 10 anti-epileptic drugs deliveries within a year without any seizure. Epileptics seizures have been identified by hospitalization related to an epileptic

---

[4] The results presented in this section have been published in the AIME conference [Dauxais et al., 2017].

[5] ATC: Anatomical Therapeutic Chemical Classification System

event, coded G40.x or G41.x with ICD-10[6]. The total number of such patients is 8,379.

Then, each patient generates two sequences: one positive and one negative. Drugs delivered within the 90 days before induction periods yield the positive sequences and those delivered within the 90 days before the positive sequence, *i.e.*, the 90 to 180 days before induction, yield the negative sequences. A three-day induction period is defined before the first seizure of each patient.

At the end of the data preparation, the dataset has two sets of 8,379 labeled sequences, and contains 127,191 events corresponding to 1,716 different drugs.

7.2 Experiments and results

Set up with $\sigma_{min} = 5.10^{-3}$, *i.e.* 42 patients[7], and $g_{min} = 1.4$, we generated 777 discriminant chronicles that discriminate the case sequences from the control sequences of 90 days. Chronicles involved 510 different multisets and 128 different event types. Similar results were obtained with sequence duration of 60, 120 and 150 days. For comparison, 535 chronicles were extracted for the dataset containing sequences of 150 days with the same parameters. The main difference between those two datasets is that chronicles are discriminant in the 150 days dataset for higher g_{min}. For example, 487 chronicles are discriminant for $g_{min} = 1.8$. Our intuition is that regular treatments are more easily discriminated from punctual deliveries when the sequences are larger. Larger discriminant chronicles are extracted from the 150 days dataset than from the 90 days dataset and these chronicles describe more accurately the underlying discriminant behavior.

Three types of pattern are of specific interest for clinicians: (1) sequences of anti-epileptic generic and brand-named drug deliveries, (2) sequences of same anti-epileptic drug deliveries, (3) sequences with anti-epileptic drug deliveries and other drug types deliveries. According to these criteria, we selected 55 discriminant chronicles involving 16 different multisets to be discussed with clinicians. The remaining of this section focuses on chronicles related to *valproic acid* (*N03AG01* ATC code, with different presentations) because it is the most frequent anti-epileptic drug occurring in the dataset but our results contain similar chronicles related to other anti-epileptic drugs like *levetiracetam* or *lamotrigine*.

7.2.1 Taking into account time in brand-to-generic substitution

We start with patterns representing switches between different presentation of *N03AG01*. Fig. 6 illustrates all discriminant patterns that have been extracted. It is noteworthy that all chronicles have temporal constraints; this means that multisets

---

[6] ICD-10: International Classification of Diseases 10th Revision

[7] This number of patients has been initially estimated important by epidemiologists to define a population of patients with similar care sequences associated to seizures.

Fig. 6 Four discriminant chronicles describing switches between same type of valproic acid (N03AG01) generic (G 438) and brand-named (R 438). $supp(\mathscr{C}_i, \mathscr{S}^+)$ respectively for $i = 1$ to 4 equals 43, 78, 71 and 43 and $supp(\mathscr{C}_i, \mathscr{S}^-)$ equals 23, 53, 39 and 30.
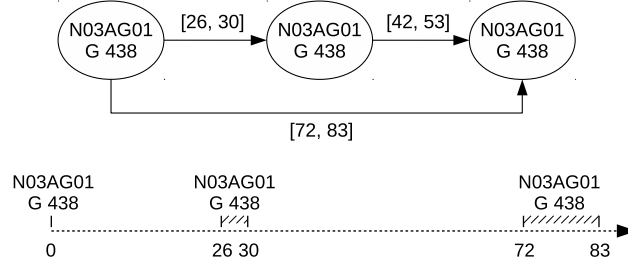

Fig. 7 Above, a chronicle describing repetitions of *valproic acid* (N03AG01) generic (G 438) and, below, its timeline representation. The chronicle is more likely related to epileptic seizures: $supp(\mathscr{C}, \mathscr{S}^+) = 50$, $supp(\mathscr{C}, \mathscr{S}^-) = 17$.

without temporal constraints are not discriminant. These results are consistent with Polard et *al.* [Polard et al., 2015] which concluded that brand-name-to-generic anti-epileptic drug substitution was not associated with an elevated risk of seizure-related hospitalization. But temporal constraints were not taken into account in the latter study. The four extracted chronicles suggest that for some small patient groups, drug switches with specific temporal constraints are more likely associated with seizures.

The first two chronicles represent delivery intervals lower than 30 days, from brand-to-generic and generic-to-brand-names respectively. The third one represents an interval between the two events that are greater than 30 days but lower than 60 days. The discriminant temporal constraints of the last one could be interpreted as [67, 90] because of the bounded duration of the study period (90 days). This chronicle represents a switch occurring more than 60 days but most of the time less than 90 days.

These behaviors may correspond to unstable treatments. In fact, anti-epileptic deliveries have to be renewed regularly in intervals of a few months, thus, a regular treatment corresponds to a delay of approximately 30 days between two anti-epileptic drug deliveries.

We next present in Figure 7 an example of a discriminant chronicle that involves three deliveries of *N03AG01* (no chronicle involves more deliveries of this anti-epileptic drug).
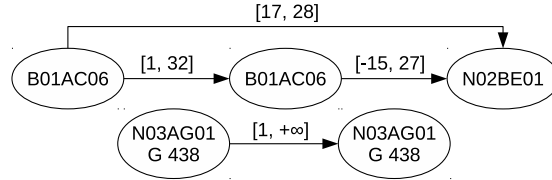
Fig. 8 A chronicle describing co-occurrences between anti-thrombosis drugs (*B01AC06*) and *valproic acid* which is more likely associated to seizures: $supp(\mathscr{C},\mathscr{S}^+) = 42$, $supp(\mathscr{C},\mathscr{S}^-) = 20$.

The growth rate of this chronicle is high (2.94). It is easy to understand and, with their discriminant temporal constraints, it can be represented on a timeline (see Figure 7, below). It is noteworthy that the timeline representation loses some information about the constraints. The first delivery is used as a starting point (t_0), but it clearly illustrates that the last delivery occurs too late after the second one (more than 30 days after). As well as previous patterns, this chronicle describes an irregularity in deliveries. More precisely, the irregularity occurs between the second and the third deliveries as described by the discriminant temporal constraints $[42,53]$ and $[72,83]$.

We conclude from observations on the previous two types of patterns that the precise numerical temporal information discovered by *DCM* is useful to identify discriminant behaviors. Analyzing pure sequential patterns does not provide enough expression power to associate switch of anti-epileptic drugs deliveries with seizures. Chronicles, specifying temporal constraints, allow us to describe the conditions under which temporal delays between anti-epileptic drugs deliveries are discriminant for epileptic seizures.

7.2.2 Example of a complex chronicle

Clinicians found the chronicle presented in Figure 8 interesting, as it does not match expert knowledge prima facie. The chronicle illustrates a potential adverse drug interaction between an anti-epileptic drug and a drug non-directly related to epilepsy, more especially aspirin (*B01AC06*), prescribed as an anti-thrombotic treatment. The discriminant temporal constraints imply that aspirin and paracetamol (*N02BE01*) are delivered within a short period (less than 28 days). There is no temporal relations between these deliveries and the deliveries of *valproic acid*. But their co-occurrence within the period of 90 days is part of the discriminatory factor.

After a deeper analysis of patient care pathways supporting this chronicle, clinicians made the hypothesis that these patients were treated for brain stroke. It is known for seriously exacerbating epilepsy and increasing seizure risk.

8 Conclusion

Chronicles mining was already studied but there was not previous work on discriminant chronicles mining, nor patterns with quantitative temporal information. The *DCM* algorithm has been proposed to solve this problem. It allows, on the one hand, to take into account temporal information and, on the other hand, to constrain the extracted chronicles with a growth rate constraint. This algorithm relies on rule learning domain which has proved to be efficient for discovering discriminant rules from quantitative attributes.

The experiments using the $Ripper_k$ algorithm showed that *DCM* is able to extract discriminant chronicles efficiently. Comparisons in terms of accuracy between *DCM* and the *BIDE-D* [Fradkin and Mörchen, 2015] algorithms showed that discriminant chronicles extracted with *DCM* are able to compete with state-of-the-art approaches without the need of classifiers. Thus, the patterns extracted by *DCM* can be interpreted by experts.

Experimental results on our case study show that *DCM* extracts a reduced number of patterns. Discriminant patterns have been presented to clinicians who conclude to their potential interestingness to explore care pathways supported by chronicles. The main interest of this experiment is to show that some interesting patterns are extracted and that clinicians do actually understand them. Contrary to black-box approaches such as *BIDE-D*, this approach provides insights to the expert about large data at hand.

At this stage of the work, our main perspective is to integrate *DCM* in a care pathway analytic tool such that extracted chronicles can easily be contextualized in care pathways and manually modified to progressively build a care sequence of interest.

An improvement perspective would be to extract temporal constraints with an approach managing multiple instances [Foulds and Frank, 2010]. In fact, in our two-step algorithm, temporal constraints are extracted from instances of a given multiset. In sequences, a multiset may occur several times due to event repetitions. As a consequence, one sequence generates several instances of temporal constraints for the same multiset. Assuming that only one instance is significant for our problem, all other instances confuse the dataset.

From the interpretability point of view, extracted pattern set remains too large. Condensed representations of chronicles mining, *e.g.*, minimal chronicles, or a post-treatment identifying similar chronicles would allow a better reading of the results.

References

Achar et al., 2012. Achar, A., Laxman, S., and Sastry, P. (2012). A unified view of the apriori-based algorithms for frequent episode discovery. *Knowledge and information systems*, 31(2):223–250.

Agrawal and Srikant, 1995. Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering*, pages 3–14. IEEE.

Allen, 1984. Allen, J. F. (1984). Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.

Alvarez et al., 2013. Alvarez, M. R., Felix, P., and Carinena, P. (2013). Discovering metric temporal constraint networks on temporal databases. *Artificial Intelligence in Medicine*, 58(3):139 – 154.

Batal et al., 2013. Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2013). A temporal pattern mining approach for classifying electronic health record data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):63.

Bay and Pazzani, 2001. Bay, S. D. and Pazzani, M. J. (2001). Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246.

Berlingerio et al., 2007. Berlingerio, M., Bonchi, F., Giannotti, F., and Turini, F. (2007). Mining clinical data with a temporal dimension: a case study. In *Proceedings of the International Conference on Bioinformatics and Biomedicine*, pages 429–436.

Bornemann et al., 2016. Bornemann, L., Lecerf, J., and Papapetrou, P. (2016). Stife: A framework for feature-based classification of sequences of temporal intervals. In *International Conference on Discovery Science*, pages 85–100. Springer.

Bringmann et al., 2011. Bringmann, B., Nijssen, S., and Zimmermann, A. (2011). Pattern-based classification: a unifying perspective. *arXiv preprint arXiv:1111.6191*.

Cohen, 1995. Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the International Conference on Machine Learning*, pages 115–123.

Concaro et al., 2009. Concaro, S., Sacchi, L., Cerra, C., Fratino, P., and Bellazzi, R. (2009). Mining healthcare data with temporal association rules: Improvements and assessment for a practical use. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 16–25.

Cram et al., 2012. Cram, D., Mathern, B., and Mille, A. (2012). A complete chronicle discovery approach: application to activity analysis. *Expert Systems*, 29(4):321–346.

Dauxais et al., 2017. Dauxais, Y., Guyet, T., Gross-Amblard, D., and Happe, A. (2017). Discriminant chronicles mining: Application to care pathways analytics. In *Proceedings of the Conference on Artificial Intelligence in Medicine*, pages 234–244. Springer.

Dechter et al., 1991. Dechter, R., Meiri, I., and Pearl, J. (1991). Temporal constraint networks. *Artificial intelligence*, 49:61–95.

Dong and Li, 1999. Dong, G. and Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of ACM SIGKDD*, pages 43–52.

Doran and Ray, 2014. Doran, G. and Ray, S. (2014). A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, 97(1):79–102.

Dousson and Duong, 1999. Dousson, C. and Duong, T. V. (1999). Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proceedings of International Conference on Artificial Intelligence*, pages 620–626.

Duivesteijn et al., 2016. Duivesteijn, W., Feelders, A. J., and Knobbe, A. (2016). Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98.

Fabrègue et al., 2014. Fabrègue, M., Braud, A., Bringay, S., Grac, C., Le Ber, F., Levet, D., and Teisseire, M. (2014). Discriminant temporal patterns for linking physico-chemistry and biology in hydro-ecosystem assessment. *Ecological Informatics*, 24:210–221.

Fabrègue et al., 2013. Fabrègue, M., Braud, A., Bringay, S., Le Ber, F., and Teisseire, M. (2013). Orderspan: Mining closed partially ordered patterns. In *International Symposium on Intelligent Data Analysis*, pages 186–197. Springer.

Foulds and Frank, 2010. Foulds, J. and Frank, E. (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(01):1–25.

Fradkin and Mörchen, 2015. Fradkin, D. and Mörchen, F. (2015). Mining sequential patterns for classification. *Knowledge and Information Systems*, 45(3):731–749.

Guyet and Quiniou, 2011. Guyet, T. and Quiniou, R. (2011). Extracting temporal patterns from interval-based sequences. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1306–1311.

Herrera et al., 2011. Herrera, F., Carmona, C. J., González, P., and Del Jesus, M. J. (2011). An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525.

Huang et al., 2012. Huang, Z., Lu, X., and Duan, H. (2012). On mining clinical pathway patterns from medical behaviors. *Artificial Intelligence in Medicine*, 56(1):35–50.

Lakshmanan et al., 2013. Lakshmanan, G. T., Rozsnyai, S., and Wang, F. (2013). Investigating clinical care pathways correlated with outcomes. In *Business process management*, pages 323–338. Springer.

Lattner et al., 2003. Lattner, A. D., Kim, S., Cervone, G., and Grefenstette, J. J. (2003). Experimental comparison of symbolic learning programs for the classification of gene network topology models. *Center for Computing Technologies–TZI*, 2:1.

Lipton, 2016. Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Mabroukeh and Ezeife, 2010. Mabroukeh, N. R. and Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Journal of Computing Survey*, 43(1):1–41.

Mannila et al., 1997. Mannila, H., Toivonen, H., and Inkeri Verkamo, A. (1997). Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3):259–289.

Mäntyjärvi et al., 2004. Mäntyjärvi, J., Himberg, J., Kangas, P., Tuomela, U., and Huuskonen, P. (2004). Sensor signal data set for exploring context recognition of mobile devices. In *Proc. of 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004)*, pages 18–23.

Mooney and Roddick, 2013. Mooney, C. H. and Roddick, J. F. (2013). Sequential pattern mining – approaches and algorithms. *ACM Journal of Computing Survey*, 45(2):1–39.

Moskovitch and Shahar, 2015. Moskovitch, R. and Shahar, Y. (2015). Fast time intervals mining using the transitivity of temporal relations. *Knowledge and Information Systems*, 42(1):21–48.

Moulis et al., 2015. Moulis, G., Lapeyre-Mestre, M., Palmaro, A., Pugnet, G., Montastruc, J.-L., and Sailler, L. (2015). French health insurance databases: What interest for medical research? *La Revue de Médecine Interne*, 36(6):411–417.

Novak et al., 2009. Novak, P. K., Lavrač, N., and Webb, G. I. (2009). Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403.

Papapetrou et al., 2005. Papapetrou, P., Kollios, G., Sclaroff, S., and Gunopulos, D. (2005). Discovering frequent arrangements of temporal intervals. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE.

Pei et al., 2002. Pei, J., Han, J., and Wang, W. (2002). Mining sequential patterns with constraints in large databases. In *Proceedings of the international conference on Information and knowledge management*, pages 18–25. ACM.

Polard et al., 2015. Polard, E., Nowak, E., Happe, A., Biraben, A., and Oger, E. (2015). Brand name to generic substitution of antiepileptic drugs does not lead to seizure-related hospitalization: a population-based case-crossover study. *Pharmacoepidemiology and drug safety*, 24(11):1161–1169.

Quiniou et al., 2001. Quiniou, R., Cordier, M., Carrault, G., and Wang, F. (2001). Application of ILP to cardiac arrhythmia characterization for chronicle recognition. In *Proceedings of International Conference on Inductive Logic Programming*, pages 220–227.

Sahuguède et al., 2018. Sahuguède, A., Fergani, S., Le Corronc, E., and Le Lann, M.-V. (2018). Mapping Chronicles to a k-dimensional Euclidean Space via Random Projections. In *14th International Conference on Automation Science and Engineering (CASE)*, page 6p. IEEE.

Santisteban and Tejada-Cárcamo, 2015. Santisteban, J. and Tejada-Cárcamo, J. (2015). Unilateral jaccard similarity coefficient. In *GSB@ SIGIR*, pages 23–27.

Starner et al., 1998. Starner, T., Weaver, J., and Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375.

Uno et al., 2004. Uno, T., Kiyomi, M., and Arimura, H. (2004). LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In *FIMI*, volume 126.

Wright et al., 2015. Wright, A. P., Wright, A. T., McCoy, A. B., and Sittig, D. F. (2015). The use of sequential pattern mining to predict next prescribed medications. *Journal of Biomedical Informatics*, 53:73–80.