

MARBLES: Mining Association Rules Buried in Long Event Sequences

Boris Cule Nikolaj Tatti Bart Goethals

Department of Mathematics and Computer Science

University of Antwerp

firstname.lastname@ua.ac.be

Abstract

Sequential pattern discovery is a well-studied field in data mining. Episodes are sequential patterns that describe events that often occur in the vicinity of each other. Episodes can impose restrictions on the order of the events, which makes them a versatile technique for describing complex patterns in the sequence. Most of the research on episodes deals with special cases such as serial and parallel episodes, while discovering general episodes is surprisingly understudied. This is particularly true when it comes to discovering association rules between them.

In this paper we propose an algorithm that mines association rules between two general episodes. On top of the traditional definitions of frequency and confidence, we introduce two novel confidence measures for the rules. The major challenge in mining these association rules is pattern explosion. To limit the output, we aim to eliminate all redundant rules. We define the class of closed association rules, and show that this class contains all non-redundant output. To make the algorithm efficient, we use further pruning steps along the way. First of all, we generate only free and closed frequent episodes from which we create candidate rules, we speed up the evaluation of the rules, and finally prune the remaining non-closed rules from the output.

1 Introduction

Discovering frequent patterns in an event sequence is an important field in data mining. Episodes, first defined by Mannila et al. [14], represent a rich class of sequential patterns, enabling us to discover events occurring in the vicinity of each other while at the same time capturing complex interactions between the events.

More specifically, a frequent episode is traditionally considered to be a set of events that reoccurs in the sequence. Gaps are allowed between the events and the order in which the events are allowed to occur is specified by the episode. The frequency of an episode is usually expressed as the number of windows of specified length in which the episode occurs, but can also be defined incorporating other concepts, such as minimal windows that contain the episode. However, it is important that the frequency is monotonically decreasing so we can use the well-known level-wise approach to mine all frequent episodes.

The order restrictions of an episode are described by a directed acyclic graph (DAG): the set of events

in a sequence covers the episode if and only if each event occurs only after all its parent events (with respect to the DAG) have occurred (see the formal definition in Section 3). Usually, only two extreme cases are considered. A parallel episode poses no restrictions on the order of the events, and a window covers the episode if the events occur in the window, in any order. In such a case, the DAG associated with the episode contains no edges. The other extreme case is a serial episode. Such an episode requires that the events occur in one, and only one, specific order in the sequence. Clearly, serial episodes are more restrictive than parallel episodes. If a serial episode is frequent, then its parallel version is also frequent.

An association rule between two episodes expresses the fact that the occurrence of one episode implies, with a high enough probability, that another episode can be found nearby. Typically, association rules are defined such that an occurrence of a smaller episode implies the occurrence of a greater episode.

The main contribution of this paper is an algorithm that mines association rules consisting of two episodes represented by DAGs. On top of that, we introduce two novel ways to define the confidence of an association rule, based on more intuitive concepts than the traditional method using sliding windows of fixed length. To reduce the size of the output, we adopt the traditional concept of closed patterns to obtain only non-redundant association rules. We present a collection of algorithms, called MARBLES, to mine all such rules, handling all three possible approaches.

So far, very little research has gone into the search for episodes based on DAGs and even less has gone into the discovery of association rules between them. In practice, such episodes have been overshadowed by parallel and serial episodes. The main reason for this is the pattern explosion. Consider the case of itemsets, where a frequent itemset of size k has 2^k frequent subsets. With an episode of size k , described by a DAG, the number of possible subepisodes is much larger, as we also have to look at all possible subsets of its edges, the

number of which grows quadratically with the number of nodes. On top of that, each association rule consists of two episodes, so the number of combinations is huge, as illustrated by the following example.

EXAMPLE 1. *Consider a sequence within which subsequence $abcd$ occurs frequently, using a fixed window of size 4. Assume that, outside these occurrences, events a , b , c and d never occur. It is easy to see that episodes $G = \{a, d\}$ and $H = a \rightarrow b \rightarrow c \rightarrow d$ will have the same frequency. The same, of course, is also true for any episode X , such that $G \subseteq X \subseteq H$. Therefore, the confidence of all association rules $X \Rightarrow Y$, where $G \subseteq X \subset Y \subseteq H$ will be equal to 1. In this case, a single simple reoccurring pattern in the sequence results in 158 association rules¹. However, just one of those rules is actually not redundant, namely $G \Rightarrow H$, as all others can be derived from that one.*

However, the advantage of episodes based on DAGs is that they allow us to capture dependencies between the events while not being too restrictive. The following example illustrates that parallel and serial episodes may be insufficient as a means of discovering all interesting association rules in a dataset.

EXAMPLE 2. *As an example we will use text data, namely inaugural speeches by presidents of the United States (see Section 7 for more details). Protocol requires the presidents to address the chief justice and the vice presidents in their speeches. Hence, we have discovered association rule $\{\text{chief}, \text{justic}, \text{vice}, \text{president}\} \Rightarrow \{\text{chief} \rightarrow \text{justic}, \text{vice} \rightarrow \text{president}\}$. This rule tells us that when these four words appear near each other, then 'chief' precedes 'justice' and 'vice' precedes 'president'. Since the actual address order varies from speech to speech, the pattern does not impose any additional restrictions. The discovered rule informs us of the frequent usage of phrases 'chief justice' and 'vice president' in each others vicinity, something we could never discover using only parallel or serial episodes.*

A popular method of reducing the size of the output in any pattern mining problem is to discover only closed patterns. A pattern is closed if there exists no superpattern with the same frequency. Some work has gone into the discovery of closed frequent episodes, but we go a step further, by introducing the concept of closed association rules. We output only the rules where the left-hand side is minimal, and the right-hand side maximal. Defining this class of association rules is not trivial, and we use a variety of computational tricks to speed up the execution of our algorithm. This

allows us to output association rules consisting of two episodes represented by DAGs, and yet keep the size of the output well under control.

Furthermore, apart from the traditional definition of the confidence of an association rule, based on the frequencies of the two episodes using a sliding window of fixed size, we introduce two additional ways to define the confidence, using either minimal windows or weighted minimal windows. We show that all three methods have their merits, and can be valuable and intuitive, depending on the nature of the input sequence and the wishes of the end user.

The rest of the paper is organised as follows: In Section 2, we discuss the most relevant related work, before presenting the main notations and concepts in Section 3. Section 4 introduces the notion of association rules using three different methods, while in Section 5 we discuss how we can limit the size of the output by eliminating redundant association rules. The algorithms that allow us to achieve this goal are presented in detail in Section 6. In Section 7 we show the results of our experiments, before presenting our conclusions in Section 8. The code of the algorithm is available online².

2 Related Work

The first attempt at discovering frequent subsequences, or serial episodes, was made by Wang et al. [22]. The dataset consisted of a number of sequences, and a pattern was considered interesting if it was long enough and could be found in a sufficient number of sequences. A complete solution to a more general problem was later provided by Agrawal and Srikant [2] using an APRIORI-style algorithm [1].

Looking for frequent general episodes in a single event sequence was first proposed by Mannila et al. [14]. The WINEPI algorithm finds all episodes that occur in a sufficient number of windows of fixed length, and generates association rules $X \Rightarrow Y$, where $X \subset Y$ and both X and Y are frequent episodes. Specific algorithms were given for the case of parallel and serial episodes, but no algorithm for detecting general episodes was provided. Tatti and Cule [18] extend the definition of an episode to be able to depict simultaneous events. They provide an algorithm for generating all frequent episodes, but no association rules.

Mannila et al. also propose MINEPI [14], an alternative interestingness measure for an episode, where the frequency is defined as the number of minimal windows that contain the episode. In this context, the authors also define association rules. Unfortunately, this frequency measure is not monotonically decreasing. How-

¹we counted this number manually

²<http://adrem.ua.ac.be/implementations>

ever, the issue can be fixed by defining frequency as the maximal number of non-overlapping minimal windows [13, 17]. Zhou et al. [24] proposed mining closed serial episodes based on the MINEPI method. However, the paper did not address the non-monotonicity issue of MINEPI. None of these follow-up papers handled the problem of association rules.

Méger and Rigotti [15] propose a method for mining association rules of the form $X \Rightarrow Y$, such that X and Y are both serial episodes, and X is a prefix of Y . Cule et al. [8] introduce an alternative interestingness measure for episodes, combining frequency with the cohesion of an episode. They further extend this work to mine association rules [7], but the method works only for parallel episodes.

A lot of research in the field of pattern discovery has gone into eliminating redundancy. An important way to tackle this problem is by outputting only closed patterns. Within sequence mining, some research has gone into outputting only closed subsequences, where a sequence is considered closed if it is not properly contained in any other sequence which has the same frequency. Yan et al. [23], Tzvetkov et al. [20], and Wang and Han [21] proposed methods for mining such closed patterns, while Garriga [6] further reduced the output by post-processing it and representing the patterns using partial orders. Harms et al. [12], meanwhile, experiment with closed serial episodes. In another attempt to trim the output, Garofalakis et al. [9] proposed a family of algorithms called SPIRIT which allow the user to define regular expressions that specify the language that the discovered patterns must belong to.

Pei et al. [16], and Tatti and Cule [19] considered restricted versions of the general problem setup of finding frequent episodes. The former approach assumes a dataset of sequences where the same label can occur only once. Hence, an episode can contain only unique labels. The latter pointed out the problem of defining a proper subset relationship between general episodes and tackled it by considering only strict episodes, where two nodes having the same label had to be connected by a path. In our work, we adopt the latter approach, extend it by allowing events in the sequence to take place at the same time, and build on it further in order to mine association rules.

Further interestingness measures for episodes, either statistically motivated or aimed at removing bias towards smaller episodes, were made by Garriga [5], Gwadera et al. [10, 11], Calders et al. [4], and Tatti [17]. All these methods, however, were limited to finding interesting episodes, and stopped short of discovering association rules between them.

Tackling redundancy within association rules, and

not only within the patterns they consist of, has been done within the field of frequent itemset mining, but not within episode mining. Bastide et al. [3] define rules as non-redundant if they consist of a minimal antecedent and a maximal consequent. We present a similar approach, dealing with the complexity of terms such as minimal and maximal in the context of episodes.

3 Preliminaries

In this section, we introduce the basic concepts that we will use throughout the paper. First we will describe our dataset.

DEFINITION 1. *We define a sequence event $e = (id(e), lab(e), ts(e))$ as a tuple consisting of three entries, a unique id number $id(e)$, a label $lab(e)$ coming from an alphabet Σ , and a time stamp integer $ts(e)$. We will assume that if $id(e) > id(f)$, then $ts(e) \geq ts(f)$. A sequence is a collection of sequence events ordered by their ids.*

Note that we are allowing multiple events to have the same time stamp even when their labels are equivalent. For the sake of simplicity, we will use the notation $s_1 \cdots s_N$ to mean a sequence $((1, s_1, 1), \dots, (N, s_N, N))$.

DEFINITION 2. *Given a sequence s and two integers i and j we define a subsequence $s[i, j] = s_i, \dots, s_j$ containing all events occurring between i and j . We define the length of subsequence $s[i, j]$ as*

$$len(s[i, j]) = ts(s_j) - ts(s_i) + 1.$$

Our next step is to define the patterns we are interested in.

DEFINITION 3. *An episode G is represented by a directed acyclic graph with labelled nodes, that is, $G = (V, E, lab)$, where $V = (v_1, \dots, v_K)$ is the set of nodes, E is the set of directed edges, and lab is the function $lab : V \rightarrow \Sigma$, mapping each node v_i to its label.*

When there is no danger of confusion, we will use the same letter to denote an episode and its graph.

DEFINITION 4. *A node n in an episode graph is a descendant of a node m if there is a path from m to n . In that case, node m is an ancestor of node n .*

We are now ready to give a precise definition of an occurrence of a pattern in a sequence.

DEFINITION 5. *Given a sequence s and an episode G we say that s covers G , or G occurs in s , if there is an injective map f mapping each node v_i to a valid*

index such that the node v_i in G and the corresponding sequence element $s_{f(v_i)}$ have the same label, $s_{f(v_i)} = \text{lab}(v_i)$, and that if there is an edge (v_i, v_j) in G , then we must have $f(v_i) < f(v_j)$. In other words, the parents of v_j must occur in s before v_j . If the mapping f is surjective, that is, all events in s are used, we will say that s is an instance of G .

An example of a sequence covering an episode is given in Figure 1.

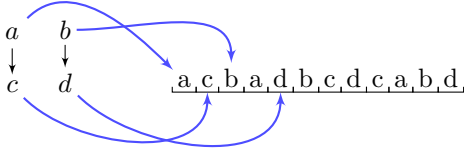


Figure 1: An example of an episode covered by a sequence.

In order to be able to discover association rules, we must first be able to compare episodes using some sort of a subset relationship.

DEFINITION 6. Given two episodes G and H , we say that G is a subepisode of H , denoted $G \subseteq H$, if the DAG describing episode G is a subgraph of the DAG describing episode H .

DEFINITION 7. Given two episodes G and H , such that $G \subseteq H$, we can express an association rule $G \Rightarrow H$. We call G the head of the rule, and H the tail of the rule.

4 Association Rules

In this section we present three possible methods to measure the frequency of an episode and the confidence of an association rule. The first one uses a sliding window of fixed size, the second is based on disjoint minimal windows, while the third one introduces the concept of weighted minimal windows.

4.1 Using Fixed Windows We start off by defining frequency of an episode in the traditional manner, based on windows of fixed length. This definition corresponds to the definition used in WINEPI [14]. The frequency is monotonically decreasing which allows us to do effective pruning while discovering frequent episodes.

DEFINITION 8. Given a window size ρ and an episode s , we define the fixed-window frequency of an episode G in s , denoted $fr_f(G; s)$, to be the number of windows of size ρ in s covering the episode,

$$fr_f(G; s) = |\{s[i, i + \rho - 1] \mid s[i, i + \rho - 1] \text{ covers } G\}|.$$

We will use $fr_f(G)$ whenever s is clear from the context. An episode is σ -frequent (or simply frequent) if its frequency is higher than or equal to some given threshold σ .

In this context, association rules can be defined in the traditional manner.

DEFINITION 9. Given a window size ρ and episodes X and Y , such that $X \subseteq Y$, we define the fixed-window confidence of the association rule $X \Rightarrow Y$, denoted $c_f(X \Rightarrow Y)$, to be the ratio of their respective frequencies,

$$c_f(X \Rightarrow Y) = \frac{fr_f(Y)}{fr_f(X)}.$$

Informally, we can interpret this definition as follows: $c_f(X \Rightarrow Y)$ is the percentage of windows that contain X that also contain Y . In other words, if we encounter a window that contains X , $c_f(X \Rightarrow Y)$ represents the probability that the window also contains Y .

4.2 Using Minimal Windows Using a sliding window of fixed length has some drawbacks, particularly in the context of association rules, as can be seen in the following examples.

EXAMPLE 3. Consider a sequence within which subsequence $abcd$ occurs frequently, using a fixed window of size 4. Assume that, outside these occurrences, events a, b, c and d never occur. The fact that $fr_f(\{a, d\}) = fr_f(a \rightarrow b \rightarrow c \rightarrow d)$ implies that $c_f(\{a, d\} \Rightarrow a \rightarrow b \rightarrow c \rightarrow d) = 1$. However, $c_f(\{b, c\} \Rightarrow a \rightarrow b \rightarrow c \rightarrow d) = \frac{1}{3}$. Intuitively, though, looking at the pattern, we note that every occurrence of $\{b, c\}$, just like every occurrence of $\{a, d\}$, implies an occurrence of $a \rightarrow b \rightarrow c \rightarrow d$. The confidences of the two rules do not reflect that.

EXAMPLE 4. Consider sequence s_1 within which subsequence $abxycd$ occurs frequently, and sequence s_2 within which subsequence $axbcyd$ occurs frequently, where x and y are noise events and are not part of the patterns. Assume we are using a fixed window of size 6, and that, outside these occurrences, events a, b, c and d never occur. As in the previous example, $c_f(\{a, d\} \Rightarrow a \rightarrow b \rightarrow c \rightarrow d) = 1$, but $c_f(\{b, c\} \Rightarrow a \rightarrow b \rightarrow c \rightarrow d)$ is now equal to $\frac{3}{5}$ in s_1 and $\frac{1}{5}$ in s_2 . Once again, the confidence values are not intuitive.

The problem with basing the definition of confidence of an association rule on the fixed-window frequencies of the two episodes is that this approach focuses on the occurrences of the episodes in the windows, rather than on their occurrences in the sequence. In the

above examples, while there are some windows that contain $\{b, c\}$ and not $a \rightarrow b \rightarrow c \rightarrow d$, it still holds that *every* occurrence of $\{b, c\}$ in the sequence can be found within an occurrence of $a \rightarrow b \rightarrow c \rightarrow d$. Our first step towards addressing this issue is to consider a different definition of the frequency of an episode in a sequence.

DEFINITION 10. *Given a sequence s and an episode G , a window $s[a, b]$ is called a minimal window of G in s , if $\text{len}(s[a, b]) \leq \rho$, $s[a, b]$ covers G , and if no proper subwindow of $s[a, b]$ covers G . We define $b(s[a, b]) = \text{ts}(s_a)$ as the beginning of the window, and $e(s[a, b]) = \text{ts}(s_b)$ as its end. We denote the set of all minimal windows of G in s with $\text{mw}(G; s)$, or simply $\text{mw}(G)$, when s is known from the context. Given a set of minimal windows W , we define a function $\text{dis}(W)$ to be equal to 1 if all windows in W are pairwise disjoint, and 0 otherwise.*

An attempt was made to define frequency of an episode as the number of its minimal windows [14]. However, this measure proved to be non-monotonic. In order to satisfy the downward-closed property, we need to consider only the non-overlapping windows [13, 17].

DEFINITION 11. *The disjoint-window frequency of an episode G in a sequence s , denoted $\text{fr}_m(G)$, is defined as the maximal number of non-overlapping minimal windows within s that contain episode G . Formally,*

$$\text{fr}_m(G) = \max \{ |W| \mid W \subseteq \text{mw}(G), \text{dis}(W) = 1 \}.$$

A naive way to define the confidence of an association rule $X \Rightarrow Y$ in the disjoint-window context would be to, once again, compute the ratio between the frequencies of Y and X . The following example shows that this might not be very intuitive either.

EXAMPLE 5. *Consider sequence s_1 within which subsequence $abcxyd$ occurs 100 times, and sequence s_2 within which subsequence $abcbed$ occurs 100 times, where x and y are noise events and are not part of the patterns. Assume that, outside these occurrences, events a, b, c and d never occur. Denote $G = \{b, c\}$ and $H = a \rightarrow b \rightarrow c \rightarrow d$. In s_1 , $\text{fr}_m(G) = 100$ and $\text{fr}_m(H) = 100$. Therefore, $\frac{\text{fr}_m(G)}{\text{fr}_m(H)} = 1$. However, in s_2 , $\text{fr}_m(G) = 200$ and $\text{fr}_m(H) = 100$, and $\frac{\text{fr}_m(G)}{\text{fr}_m(H)} = 0.5$. Clearly, in both sequences, each occurrence of G implies an occurrence of H , and the results for s_2 are not satisfactory.*

While defining the frequency of an episode using minimal windows solved some of the problems inherent in the fixed-window method, we clearly still need to find

a better way to define the confidence of an association rule. Intuitively, we wish the confidence of rule $X \Rightarrow Y$ to express the probability of encountering a minimal window of Y having encountered a minimal window of X . More formally, we wish to know what percentage of minimal windows of X are contained within minimal windows of Y . However, in order to do this, we are forced to drop the constraint that the minimal windows in question must be disjoint. The reasoning behind this decision is shown in the following example.

EXAMPLE 6. *Consider sequence $s = abcadbcbcd$. Denote $G = \{b, c\}$ and $H = a \rightarrow b \rightarrow c \rightarrow d$. The disjoint-window frequency of H is 1, but the sequence contains two overlapping minimal windows of H . There are three minimal windows of G , and each of them is contained within a minimal window of H . However, if we were to only use non-overlapping windows of H , we would be faced with two problems. First of all, the confidence of rule $G \Rightarrow H$ would depend on our choice of disjoint minimal windows — if we chose the first minimal window of H , $s[1, 5]$, we would find two occurrences of G outside it and the confidence of the rule would be $\frac{1}{3}$, whereas if we chose the second minimal window of H , $s[4, 10]$, we would find just one occurrence of G outside it, and the confidence would be $\frac{2}{3}$. More importantly, whichever choice we made, we would not be able to get the correct result, showing that every occurrence of G is contained within an occurrence of H .*

Now that we have seen that we cannot define the confidence of an association rule using either the disjoint-window frequencies, or the containment of the disjoint occurrences, of the two episodes, we are ready to present a definition that corresponds exactly to our intuition.

DEFINITION 12. *Given episodes X and Y , such that $X \subset Y$, and a minimal window $s[a, b]$ of episode X . Assume there exists a minimal window $s[c, d]$ of Y such that $c \leq a$ and $b \leq d$, then we define the minimal-extensibility of occurrence $s[a, b]$ of X into an occurrence of Y as*

$$\text{ext}_m(s[a, b], X, Y) = 1.$$

If there exists no such minimal window of Y , we define $\text{ext}_m(s[a, b], X, Y) = 0$.

DEFINITION 13. *Given episodes X and Y , such that $X \subset Y$, we define the minimal-window confidence of the association rule $X \Rightarrow Y$, denoted $c_m(X \Rightarrow Y)$, to be the proportion of minimal windows of X that are contained within a minimal window of Y ,*

$$c_m(X \Rightarrow Y) = \frac{1}{|\text{mw}(X)|} \sum_{w \in \text{mw}(X)} \text{ext}_m(w, X, Y).$$

4.3 Weighted Minimal Windows The problem with using minimal windows is that they do not take the cohesion of a pattern into account. A minimal window of size 2 will make the same contribution towards the frequency of an episode as a minimal window of size 10. Similarly, using only minimal windows, the confidence of the association rule $X \Rightarrow Y$ would depend solely on the inclusion of minimal windows of X inside minimal windows of Y , regardless of how much the window would need to be expanded in order to find the whole of Y . The following example illustrates these problems further.

EXAMPLE 7. Consider sequences $s_1 = axbycd$ and $s_2 = abxycd$. Episode $b \rightarrow c$ occurs once in each sequence, so its disjoint-window frequency would be equal to 1 in each sequence. However, the implied pattern an occurrence of a is followed by an occurrence of c is clearly stronger in s_1 , where b is immediately followed by a c . In both sequences, the minimal window of $b \rightarrow c$ is contained within a minimal window of $a \rightarrow b \rightarrow c \rightarrow d$, and the confidence of the rule $b \rightarrow c \Rightarrow a \rightarrow b \rightarrow c \rightarrow d$ using minimal windows would therefore be equal to 1 for both sequences. However, the meaning of the rule, an occurrence of episode $b \rightarrow c$ is likely to be contained within a nearby occurrence of $a \rightarrow b \rightarrow c \rightarrow d$, can be seen more clearly in s_2 . In s_1 , we must look much further before we encounter the whole of the larger episode.

To address these problems, we must first modify the definition of the frequency of an episode in a sequence.

DEFINITION 14. The total weight of a set of windows W in a sequence s , denoted $tw(W)$, is defined as

$$tw(W) = \sum_{w \in W} \frac{1}{len(w)}.$$

The weighted-window frequency of an episode G in a sequence s , denoted $fr_w(G)$, is defined as

$$fr_w(G) = \max \{tw(W) \mid W \subseteq mw(G), dis(W) = 1\}.$$

Informally, the shorter the window, the greater its contribution towards the overall frequency of the episode. Note that, once again, we need to consider only non-overlapping windows in order to satisfy the downward-closed property. However, this time, as the following example illustrates, we need to be much more careful as to which windows we choose, hence the need to maximise the sum of the windows' inverse lengths, rather than simply their number.

EXAMPLE 8. Consider sequence $s = abxywzcbaxywz$ and parallel episode $X = \{a, b, c\}$. s contains three

minimal windows of this episode, namely $s[1, 7]$, $s[7, 9]$ and $s[8, 14]$. Note that the second minimal window overlaps with both the first and the third, so the disjoint-window frequency of X would be equal to 2. The two windows that would count towards this frequency would be $s[1, 7]$ and $s[8, 14]$. However, if we want to maximise the sum of the inverse lengths of the windows, we cannot simply take the maximal number of non-overlapping windows and then compute this sum. In our example, we have two possible sets of non-overlapping windows, $s[1, 7]$ and $s[8, 14]$, and $s[7, 9]$ alone. In the first case, we have two windows of size 7, and the resulting sum would be $\frac{2}{7}$. In the second case, we have one window of size 3, and the sum would, therefore, be equal to $\frac{1}{3}$. We see that by choosing a smaller number of non-overlapping windows, we actually get a higher weighted-window frequency for the episode. We conclude that $fr_w(X) = \frac{1}{3}$.

As was already shown in Example 7, we also need to take the various lengths of the minimal windows into account when computing the confidence of association rules. For reasons similar to those discussed in Section 4.2, we once again cannot take only the disjoint minimal windows into account. Intuitively, we want the confidence of rule $X \Rightarrow Y$ to correspond to the likelihood of encountering the whole of Y once we have encountered X . The nearer the whole of Y is to X on average, the higher the confidence should be.

DEFINITION 15. Given episodes X and Y , such that $X \subset Y$, and a minimal window $s[a, b]$ of episode X . Assume there exists a minimal window $s[c, d]$ of Y such that $c \leq a$ and $b \leq d$, and that this is the smallest window that contains both Y and $s[a, b]$, then we define the weighted-extensibility of occurrence $s[a, b]$ of X into an occurrence of Y as

$$ext_w(s[a, b], X, Y) = \frac{len(s[a, b])}{len(s[c, d])}.$$

If there exists no such minimal window of Y , we define $ext(s[a, b], X, Y) = 0$.

DEFINITION 16. Given episodes X and Y , such that $X \subset Y$, we define the weighted-window confidence of the association rule $X \Rightarrow Y$, denoted $c_w(X \Rightarrow Y)$, to be the average weighted-extensibility of an occurrence of X into an occurrence of Y ,

$$c_w(X \Rightarrow Y) = \frac{1}{|mw(X)|} \sum_{w \in mw(X)} ext_w(w, X, Y).$$

The following example illustrates that, when extending an occurrence of X in order to find an occurrence of Y , it is important that we find the smallest such occurrence.

EXAMPLE 9. Consider sequence $s = axybca$ and episodes $X = b \rightarrow c$ and $Y = \{a, b \rightarrow c\}$. Sequence s contains one minimal window of X , namely $s[4, 5]$. This occurrence of X can be extended in search of an occurrence of Y . There are two candidate minimal windows of Y that can be considered, $s[1, 5]$ and $s[4, 6]$. If we choose the former, the extensibility of $s[4, 5]$ into Y would equal $\frac{2}{5}$, and if we use the latter, this value would rise up to $\frac{2}{3}$. Since we are interested in how far we need to look in order to extend an occurrence of X into an occurrence of Y , we clearly need to look for the smallest minimal window of Y that satisfies the conditions.

We now show what effect the new definitions would have on the patterns discussed in Example 7.

EXAMPLE 10. Consider sequences $s_1 = axbycd$ and $s_2 = abxycd$, and episodes $X = b \rightarrow c$ and $Y = a \rightarrow b \rightarrow c \rightarrow d$. First of all, we see that $fr_w(X; s_1) = \frac{1}{2}$, while $fr_w(X; s_2) = \frac{1}{4}$. However, we also see that $c_w(X \Rightarrow Y)$ equals $\frac{1}{3}$ in s_1 and $\frac{2}{3}$ in s_2 . This shows that both problems mentioned in Example 7 have been successfully addressed.

Table 1 shows an overview of the results of applying the three methods presented in Sections 4.1, 4.2 and 4.3 on the patterns and sequences discussed in Examples 7 and 10.

Pattern	s_1	s_2
$fr_f(X)$	5	3
$fr_f(Y)$	1	1
$c_f(X \Rightarrow Y)$	0.2	0.33
$fr_m(X)$	1	1
$fr_m(Y)$	1	1
$c_m(X \Rightarrow Y)$	1	1
$fr_w(X)$	0.5	0.25
$fr_w(Y)$	0.17	0.17
$c_w(X \Rightarrow Y)$	0.33	0.67

Table 1: An overview of all presented methods applied to the sequences given in Example 7. The window size used for the fixed-window method was 6.

5 Eliminating Redundancy

In this section, we will denote the frequency of an episode G with $fr(G)$, and the confidence of a rule $X \Rightarrow Y$ with $c(X \Rightarrow Y)$, regardless of which method we are using, as what follows applies to all three methods.

It is a known fact in episode mining that two different DAGs may actually represent the same episode. To

tackle this redundancy, we must impose some restrictions on the types of DAGs we will discover.

DEFINITION 17. An episode G is called *transitively closed* if for two nodes n and m , such that there exists a path from n to m , there also exists an edge from n to m .

DEFINITION 18. An episode G is called *strict* if for any two nodes v and w in G sharing the same label, there exists a path either from v to w or from w to v .

As has been shown by Tatti and Cule [19], the issue of the above mentioned redundancy is resolved within the class of transitively closed strict episodes. Therefore, we will mine only association rules consisting of strict, transitively closed episodes. In the remaining text, we consider episodes to be strict and transitively closed, unless stated otherwise.

A traditional step towards further reducing the output is to generate only closed episodes.

DEFINITION 19. An episode G is *closed* if there exists no episode H , such that $G \subset H$ and $fr(G) = fr(H)$.

However, we want to eliminate redundancy within the association rules we output, and not only among the frequent episodes. In fact, what we want to achieve is to output only the rules that give us most information. More precisely, we want the left-hand side to be minimal (the most general episode) and the right-hand side maximal (the most specific episode), among all those for which the rule holds. To achieve this, we first need to define what exactly we mean by minimal in this context.

DEFINITION 20. An episode G is *free* if there exists no episode H , such that $H \subset G$ and $fr(G) = fr(H)$.

If we wish to fully eliminate redundancy in the output, we must consider only those rules that consist of a free episode on the left-hand side, and a closed episode on the right-hand side.

DEFINITION 21. Given two association rules $R_1 = X_1 \Rightarrow Y_1$ and $R_2 = X_2 \Rightarrow Y_2$, we say that R_1 is a subset of R_2 if $X_2 \subseteq X_1$ and $Y_1 \subseteq Y_2$. In this case, we denote $R_1 \subseteq R_2$. If $X_2 \subset X_1$ or $Y_1 \subset Y_2$, we denote $R_1 \subset R_2$.

Unlike the frequency of an episode, the confidence of an association rule is not necessarily a monotonic measure, given this subset relationship between rules. The following example illustrates that, given rules R_1 and R_2 , such that $R_1 \subset R_2$, the confidence of R_1 could be smaller than, greater than, or equal to the confidence of R_2 .

EXAMPLE 11. Consider sequence $s = aacbabda$, using the minimal window method. Consider episodes $G_1 = a$, $G_2 = a \rightarrow b$, $G_3 = a \rightarrow c \rightarrow b$ and $G_4 = a \rightarrow c \rightarrow b \rightarrow d$, and rules $R_1 = G_1 \Rightarrow G_4$, $R_2 = G_2 \Rightarrow G_4$ and $R_3 = G_2 \Rightarrow G_3$. Clearly, it holds that $G_1 \subset G_2 \subset G_3 \subset G_4$, and therefore $R_3 \subset R_2 \subset R_1$. To compute the disjoint-window confidence of the three rules, we first need to identify all minimal occurrences of the four episodes in s . There are four minimal occurrences of G_1 , $s[1,1]$, $s[2,2]$, $s[5,5]$ and $s[8,8]$, two minimal occurrences of G_2 , $s[2,4]$ and $s[5,6]$, one minimal occurrence of G_3 , $s[2,4]$, and one minimal occurrence of G_4 , $s[2,7]$. For R_1 , we see that two of the four minimal occurrences of G_1 can be found within a minimal occurrence of G_4 , and therefore $c_m(R_1) = 0.5$. For R_2 , we find that both minimal occurrences of G_2 can be found within a minimal occurrence of G_4 , so $c_m(R_2) = 1$. Finally, for R_3 , we see that just one minimal occurrence of G_2 can be found within a minimal occurrence of G_3 , and $c_m(R_3) = 0.5$. To sum up, $R_3 \subset R_2 \subset R_1$, but $c_m(R_1) = c_m(R_3) < c_m(R_2)$.

Example 11 illustrates that we cannot apply the usual definition of closure to association rules. It is possible for two rules, R_1 and R_2 , such that $R_1 \subset R_2$, to have the same confidence purely *by coincidence*. However, in such a case, we cannot derive the confidence of R_1 using the confidence of R_2 , and we cannot leave R_1 out of the output. We must therefore be a little bit more careful when defining non-redundant rules.

DEFINITION 22. Given episodes X and Y , such that $X \subset Y$, the association rule $R = X \Rightarrow Y$ is not closed if there exists a rule R_1 , such that $R \subset R_1$ and $c(R) = c(R_1)$, and there exists no rule R_2 , such that $R \subset R_2 \subset R_1$ and $c(R) \neq c(R_2)$. An association rule that does not satisfy these conditions is closed.

Formally, given a sequence s , a user-chosen window size ρ , a frequency threshold σ and a confidence threshold ϕ , we will output all closed association rules of the form $X \Rightarrow Y$, where $X \subset Y$, $fr(Y) \geq \sigma$, and $c(X \Rightarrow Y) \geq \phi$.

6 MARBLES

In this section, we present our algorithms. We start off by describing the first step, mining frequent episodes, before moving on to the algorithms we developed for mining closed association rules.

6.1 Mining episodes In order to generate association rules, we first need to generate frequent episodes. As our goal is to mine only closed association rules we do not need to consider all frequent episodes. We will resort to *i*-closed episodes [19].

Due to space restrictions we will only sketch the necessary definitions. For more formal definitions, we refer the reader to Section 5 in the paper that originally introduced the concept of *i*-closure. [19].

DEFINITION 23. An *i*-closure is a function $icl(G; s) = H$ mapping an episode G to an episode H such that $G \subseteq H$. H is constructed from G in two phases. Firstly, if an event with a label l occurs in every minimal window of G and there is no node in G labelled with l , then we add a new node into G with label l . This is repeated until no new additions are possible. In the second phase, we add new edges. If in every instance of G a node v occurs before a node w , then we add an edge from v to w .

DEFINITION 24. Given a sequence s , we say that an episode G is *i*-closed if $G = icl(G; s)$. From the fundamental properties of the closure, it follows that G is the maximal episode for which the closure is equal to G . Similarly, we say that G is *i*-free (or an *i*-generator) if there is no H such that $H \subset G$ and $icl(G; s) = icl(H; s)$. In other words, G is a minimal episode that can produce $icl(G; s)$.

Note that there can be several minimal episodes that have the same *i*-closure but only one maximal episode.

The reason we are using *i*-closed episodes is that we can use them to generate closed association rules.

PROPOSITION 6.1. Let $X \Rightarrow Y$ be a closed association rule. Then X is *i*-free and Y is *i*-closed.

Proof. All three confidence measures depend only on the minimal windows of X and Y . As demonstrated in the proof of Theorem 6 in the original paper [19], if two episodes, say G and H , have the same *i*-closure, i.e., $icl(G; s) = icl(H; s)$, then they have exactly the same minimal windows. This implies that $c(X \Rightarrow Y) = c(X \Rightarrow icl(Y; s))$ proving that Y must be *i*-closed. Similarly if we have $X' \subseteq X$ such that $icl(X'; s) = icl(X; s)$, then $c(X \Rightarrow Y) = c(X' \Rightarrow Y)$ which immediately implies that $X = X'$. Consequently, X must be *i*-free.

To mine episodes we employ the MINEEPISODES algorithm given in the original paper [19]. The algorithm follows a standard BFS-approach for closed patterns by discovering first all frequent *i*-free patterns and computing the closure of each *i*-free episode. Hence, the output of this algorithm are all frequent *i*-free episodes and their closures. Typically, free patterns are suppressed from the final output but in this case we need them to generate the left-hand side of the association rules.

MINEEPISODES can either use fixed windows or minimal windows as a frequency constraint. The algorithm computes the frequency by first discovering all minimal windows and then computing the frequency using the discovered windows. We extend the algorithm to handle weighted minimal windows by introducing a dynamic program (given in Algorithm 1). The algorithm takes a list of minimal windows for an episode G and computes $fr_w(G)$.

Algorithm 1: WEIGHTFREQUENCY. Computes weighted frequency $fr_m(X)$.

input : list of minimal windows
 $V = \{v_1, \dots, v_N\}$ of episode X
output: $fr_m(X)$

- 1 $j \leftarrow 1$;
- 2 **foreach** $v_i \in V$ **do**
- 3 **while** $j \leq N$ **and** $e(v_i) \geq b(v_j)$ **do**
- 4 $j \leftarrow j + 1$;
- 5 $d_i \leftarrow j$;
- 6 $c_{N+1} \leftarrow 0$;
- 7 **for** $i = N \dots 1$ **do**
- 8 $c_i \leftarrow \max(1/\text{len}(v_i) + c_{d_i}, c_{i+1})$;
- 9 **return** c_1 ;

Consider that we have an ordered list of minimal windows $V = \{v_1, \dots, v_N\}$. We need to select a subset of V , containing disjoint windows that maximise the total weight. In order to do that, let c_i be the maximal weight of a subset of disjoint windows of $\{v_i, \dots, v_N\}$. Then it is easy to see that

$$c_i = \max(1/\text{len}(v_i) + c_d, c_{i+1}),$$

where d is the index of the next minimal window v_d that is disjoint with v_i . The left side in the *max* corresponds to using v_i in the subset and the right side corresponds to omitting v_i from the so far best disjoint collection.

The algorithm first finds d_i , the index of the next disjoint window for each v_i and then constructs the weights c_i . Both steps require $O(N)$ time and memory.

6.2 Mining association rules Now that we have mined episodes, the next step is to build association rules. In order to do that, we introduce MARBLES given in Algorithm 2. The algorithm takes as input episodes mined in the first step and builds rules from these episodes. We assume that episodes are provided in specific groups. The input consists of a list of pairs, where the first element is an i -closed episode, say X , and the second element is a list, say \mathcal{G} , of all i -free episodes that have X as their closure. The reason

for this grouping is that the confidence is equal for all rules of form $G \Rightarrow Y$, where $G \in \mathcal{G}$. In fact, $c(G \Rightarrow Y) = c(X \Rightarrow Y)$.

Of the three definitions of confidence given in Section 4, the fixed-window confidence is the only one that is monotonic. Confidence based on (weighted) minimal windows is not monotonic, hence we have to resort to an exhaustive enumeration.

Algorithm 2: MARBLES. Mines closed association rules.

input : confidence threshold ϕ , list \mathcal{X} of pairs (X, \mathcal{G}) , where X is an i -closed episode and \mathcal{G} are the i -free episodes having X as their i -closure
output: list of closed association rules

- 1 $\mathcal{R} \leftarrow \emptyset$;
- 2 **foreach** $(X_1, \mathcal{G}_1), (X_2, \mathcal{G}_2) \in \mathcal{X}$ s.t. $X_1 \subseteq X_2$ **do**
- 3 **foreach** $G \in \mathcal{G}_1$ **do**
- 4 add $R = G \Rightarrow X_2$ to \mathcal{R} ;
- 5 **if** $c(R) < \phi$ **then** mark R ;
- 6 **foreach** $R_1 \in \mathcal{R}$ **do**
- 7 $\mathcal{S} \leftarrow \emptyset$;
- 8 **foreach** $R_2 \in \mathcal{R}, R_1 \subsetneq R_2$ **do**
- 9 remove any rule S from \mathcal{S} s.t. $R_2 \subseteq S$;
- 10 **if** there is no $S \in \mathcal{S}$ s.t. $S \subseteq R_2$ **then**
- 11 add R_2 to \mathcal{S} ;
- 12 **if** there is $S \in \mathcal{S}$ s.t. $c(R_1) = c(S)$ **then**
- 13 mark R_1 ;
- 14 **return** unmarked rules from \mathcal{R} ;

The first loop in the algorithm discovers all confident association rules using i -free and i -closed episodes. This set contains all closed association rules, and possibly some redundant rules. To remove the redundant rules, for each rule R_1 , we first construct a set of its minimal superrules \mathcal{S} . If any of the rules in \mathcal{S} has the same confidence as R_1 , we mark R_1 as non-closed, and consequently purge it from output.

In order to implement this algorithm efficiently, we need an efficient technique for enumerating all superepisodes of a given episode (used in the first loop) and for enumerating all superrules of a given association rule (used in the second loop). Assume that we have a list of episodes \mathcal{E} . We begin by grouping \mathcal{E} into groups \mathcal{E}_L , where L is a multiset of labels and \mathcal{E}_L contains episodes containing exactly the labels L . We then create index lists for these groups of form $I_{l,o}$, such that $I_{l,o}$ contains \mathcal{E}_L , where l occurs in L o times. Within each group \mathcal{E}_L , for each edge e , we create an index list of episodes I_e to contain all episodes containing edge e .

When searching for superepisodes of a given episode, say G , we first find episode groups \mathcal{E}_L whose labels are a superset of or equal to the labels of G . After this is done, we then find a mapping between the labels of G and L (if there are several, we test them all). Once a mapping is found we map the edges of G and find all episodes that contain the edges of G .

To query association rules, we build a similar indexing structure, only now we group rules by their head episode. We index these groups using the head episode and within each group we index each rule with its tail.

Our final missing step in the algorithm is to actually compute the confidence. Computing confidence using fixed-windows is trivial. Hence, we focus on computing confidence based on minimal windows. The algorithms are given in Algorithm 3 and 4.

Algorithm 3: MINWINCONFIDENCE. Computes confidence $c_m(X \Rightarrow Y)$.

input : list of minimal windows
 $V = \{v_1, \dots, v_N\}$ of episode X , list of minimal windows $W = \{w_1, \dots, w_M\}$ of episode Y

output: $c_m(X \Rightarrow Y)$

```

1  $u \leftarrow 0; i \leftarrow 1;$ 
2 foreach  $v \in V$  do
3   while  $i \leq M$  and  $e(w_i) \leq e(v)$  do
4      $i \leftarrow i + 1;$ 
5   if  $b(w_i) \leq b(v)$  then  $u \leftarrow u + 1;$ 
6 return  $u/N;$ 

```

The algorithm MINWINCONFIDENCE takes as input two ordered lists of minimal windows, V for the head episode and W for the tail episode. The algorithm then enumerates windows in V and tries to find a covering window from W . Since W and V are ordered, we do not need to search the covering window from beginning but instead from the last inspected window. This brings the run-time to $O(|V| + |W|)$.

The algorithm WEIGHTCONFIDENCE is similar to MINWINCONFIDENCE. It also takes two ordered lists of minimal windows as input, V for the head episode and W for the tail episode. The only difference is that this time we do not need to simply find any covering window, but the smallest such window. This costs us an extra for-loop which brings the run-time to $O(|V| + |W| + |V|C)$, where C is the average number of windows in W covering a window $v \in V$. In practice, this number is small, making the algorithm efficient.

Algorithm 4: WEIGHTCONFIDENCE. Computes weighted confidence $c_w(X \Rightarrow Y)$.

input : list of minimal windows
 $V = \{v_1, \dots, v_N\}$ of episode X , list of minimal windows $W = \{w_1, \dots, w_M\}$ of episode Y

output: $c_w(X \Rightarrow Y)$

```

1  $c \leftarrow 0; i \leftarrow 1;$ 
2 foreach  $v \in V$  do
3   while  $i \leq M$  and  $e(w_i) \leq e(v)$  do
4      $i \leftarrow i + 1;$ 
5    $j \leftarrow i;$ 
6    $l \leftarrow \infty;$ 
7   while  $j \leq M$  and  $b(w_j) \leq b(v)$  do
8     if  $len(w) \leq l$  then
9        $l \leftarrow len(w);$ 
10       $i \leftarrow j;$ 
11      $j \leftarrow j + 1;$ 
12    $c \leftarrow c + len(w)/l;$ 
13 return  $c/N;$ 

```

7 Experiments

We tested our algorithm on two real-world datasets — a text dataset, and a train delay dataset.

The first set of experiments was done on a text dataset, *address*, consisting of the inaugural addresses by the presidents of the United States³, merged to form a single long sequence. We processed the sequence using the Porter Stemmer⁴ and removed the stop words. The resulting sequence was 62 066 words long, and contained 5 295 different items.

We used a window of size 15 for all our experiments, we kept the frequency threshold fixed and varied the confidence threshold σ . The main goal of our experiments was to demonstrate how we tackle the problem of pattern explosion. Table 2 shows how the total number of confident association rules compared with the number of closed rules we discovered, using the fixed-window, minimal-window, and weighted-window method, respectively. The results show that the reduction was significant at all thresholds. In all our experiments we pushed the frequency threshold as low as possible, until the algorithm for generating all confident association rules began to take too long or ran out of memory. This demonstrates the need to generate i -closed and i -free episodes as an intermediary step, as our algorithm for mining only the closed rules ran much

³taken from <http://www.bartleby.com/124/pres68>

⁴<http://tartarus.org/~martin/PorterStemmer/>

faster, and could handle much lower frequency thresholds.

method	σ	ϕ	closed	total
fixed	62	1.0	492	55 529
		0.8	2 525	147 483
		0.6	5 680	177 698
		0.4	9 183	198 049
		0.2	12 080	210 224
		0.0	45 079	262 050
minimal	8	1.0	469	1 299
		0.8	1 610	3 234
		0.6	4 544	7 173
		0.4	9 457	13 101
		0.2	15 627	20 553
		0.0	50 364	59 278
weighted	1	1.0	2 143	39 459
		0.8	4 209	52 903
		0.6	10 072	83 987
		0.4	17 379	106 627
		0.2	21 099	122 560
		0.0	88 098	214 560

Table 2: The comparison of the number of closed association rules and the total number of rules discovered in the *address* dataset, using the three different methods. The window size ρ used in all experiments was 15.

Our second set of experiments was conducted on a dataset consisting of trains delayed at a single railway station in Belgium. The dataset consists of actual departure times of delayed trains, coupled with train numbers, and contains 10 115 events involving 1 280 different train IDs, stretching over a period of one month. For the fixed-window method, a window of 30 minutes was chosen by a domain expert. The time stamps were expressed in seconds, so a single train being delayed on a particular day would be found in 1 800 windows. Therefore, the fixed-window frequency threshold for interesting patterns had to be set relatively high. The results of our experiments are shown in Table 3. Again, it is clearly visible that our algorithm significantly reduces the size of the output.

Note that, because of the data characteristics, the majority of the frequent episodes have a small number of nodes. Therefore, most of the rules consist of small episodes on both sides. Such rules tend to be closed, and the reduction in the size of the output mainly occurs at thresholds where association rules consisting of larger episodes begin to appear in the output.

We can also observe that the reduction depends a lot on the characteristics of the dataset. For example, in

method	σ	ϕ	closed	total
fixed	20500	1.0	2 594	6 012
		0.8	33 368	72 477
		0.6	70 905	140 986
		0.4	93 565	178 936
		0.2	93 999	179 713
		0.0	93 999	179 713
minimal	18	1.00	252	81 470
		0.95	302	82 549
		0.90	740	130 543
		0.85	789	131 572
		0.80	896	133 602
		0 – 0.60	960	135 126
weighted	0.03	1.0	4 775	47 019
		0.8	18 338	73 810
		0.6	43 935	127 543
		0.4	72 356	177 442
		0.2	92 944	222 646
		0.0	149 245	327 433

Table 3: The comparison of the number of closed association rules and the total number of rules discovered in the *trains* dataset, using the three different methods. The window size ρ used in all experiments was 1800.

the dense *address* dataset, the reduction is largest in the fixed-window setting, while in the sparse *trains* dataset, the minimal-window method gives the best reduction.

8 Conclusions

In this paper, we present a complete association rule miner for strict episodes, a large subclass of general episodes, represented by directed acyclic graphs (DAGs). We approach the problem from three angles, defining the frequency using windows of fixed size, minimal windows, and weighted minimal windows. While the first method is computationally the least demanding, we show that the other two can be more intuitive and meaningful. We define and mine association rules within all three settings.

Furthermore, we tackle the problem of pattern explosion. The pattern explosion is exponential already when it comes to itemsets, it is much more of a problem within the field of episodes, and it culminates when we attempt to generate association rules between general episodes. We define closed association rules, thus eliminating all redundant rules from the output. Our algorithms take advantage of various properties of closed rules, such as the fact that the left-hand side must consist of a free episode, and the right-hand side of a closed episode, which allows us to discover

closed association rules efficiently. Our experiments demonstrate that the reduction in the size of the output is considerable.

In future work, we intend to investigate if it is possible to extend this work to the complete class of general episodes, dropping the constraint that they must be strict. An interesting property of the confidence of an association rule is that it is not always a monotonic measure. Further research could also be dedicated to examining if it would still be possible to somehow prune some of the rules, based on some other criteria.

9 Acknowledgments

Nikolaj Tatti is supported by a Post-doctoral Fellowship of the Research Foundation – Flanders (FWO).

References

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994)*, pages 487–499, 1994.
- [2] R. Agrawal and R. Srikant. Mining sequential patterns. *11th International Conference on Data Engineering (ICDE 1995)*, 0:3–14, 1995.
- [3] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic*, pages 972–986, 2000.
- [4] T. Calders, N. Dexters, and B. Goethals. Mining frequent itemsets in a stream. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 83–92, 2007.
- [5] G. Casas-Garriga. Discovering unbounded episodes in sequential data. In *Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 83–94, 2003.
- [6] G. Casas-Garriga. Summarizing sequential data with closed partial orders. In *Proceedings of the SIAM International Conference on Data Mining (SDM 2005)*, pages 380–391, 2005.
- [7] B. Cule and B. Goethals. Mining association rules in long sequences. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-2010)*, pages 300–309, 2010.
- [8] B. Cule, B. Goethals, and C. Robardet. A new constraint for mining sets in sequences. In *Proceedings of the SIAM International Conference on Data Mining (SDM 2009)*, pages 317–328, 2009.
- [9] M. Garofalakis, R. Rastogi, and K. Shim. Mining sequential patterns with regular expression constraints. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):530–552, 2002.
- [10] R. Gwadera, M. J. Atallah, and W. Szpankowski. Markov models for identification of significant episodes. In *Proceedings of the SIAM International Conference on Data Mining (SDM 2005)*, pages 404–414, 2005.
- [11] R. Gwadera, M. J. Atallah, and W. Szpankowski. Reliable detection of episodes in event sequences. *Knowledge and Information Systems*, 7(4):415–437, 2005.
- [12] S. K. Harms, J. S. Deogun, J. Saquer, and T. Tadesse. Discovering representative episodal association rules from event sequences using frequent closed episode sets and event constraints. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2001)*, pages 603–606, 2001.
- [13] S. Laxman, P. S. Sastry, and K. P. Unnikrishnan. A fast algorithm for finding frequent episodes in event streams. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge discovery and data mining (KDD 2007)*, pages 410–419, 2007.
- [14] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.
- [15] N. Méger and C. Rigotti. Constraint-based mining of episode rules and optimal window sizes. In *Knowledge Discovery in Databases: PKDD 2004, 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 313–324, 2004.
- [16] J. Pei, H. Wang, J. Liu, K. Wang, J. Wang, and P. S. Yu. Discovering frequent closed partial orders from strings. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1467–1481, 2006.
- [17] N. Tatti. Significance of episodes based on minimal windows. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM 2009)*, pages 513–522, 2009.
- [18] N. Tatti and B. Cule. Mining closed episodes with simultaneous events. In *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2011)*, pages 1172–1180, 2011.
- [19] N. Tatti and B. Cule. Mining closed strict episodes. *Data Mining and Knowledge Discovery*, 2011. <http://dx.doi.org/10.1007/s10618-011-0232-z>.
- [20] P. Tzvetkov, X. Yan, and J. Han. Tsp: Mining top-k closed sequential patterns. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003)*, pages 347–354, 2003.
- [21] J. Wang and J. Han. Bide: Efficient mining of frequent closed sequences. *20th International Conference on Data Engineering (ICDE 2004)*, 0:79, 2004.
- [22] J. T.-L. Wang, G.-W. Chirn, T. G. Marr, B. Shapiro, D. Shasha, and K. Zhang. Combinatorial pattern discovery for scientific data: some preliminary results. *ACM SIGMOD Record*, 23(2):115–125, 1994.
- [23] X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *Proceedings of the SIAM International Conference on Data Mining (SDM 2003)*, pages 166–177, 2003.
- [24] W. Zhou, H. Liu, and H. Cheng. Mining closed episodes from event sequences efficiently. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining(1)*, pages 310–318, 2010.