GROSSO: Mining Statistically Robust Patterns from a Sequence of Datasets

Andrea Tonon
Department of Information Engineering
University of Padova
Padova, Italy
andrea.tonon@dei.unipd.it

Fabio Vandin
Department of Information Engineering
University of Padova
Padova, Italy
fabio.vandin@unipd.it

Grosso: (Italian adj.) large, big, robust.

***Abstract*—Pattern mining is a fundamental data mining task with applications in several domains. In this work, we consider the scenario in which we have a sequence of datasets generated by potentially different underlying generative processes, and we study the problem of mining *statistically robust patterns*, which are patterns whose probabilities of appearing in transactions drawn from such generative processes respect well defined conditions. Such conditions define the patterns of interest, describing the evolution of their probabilities through the datasets in the sequence, which may, for example, increase, decrease, or stay stable, through the sequence. Due to the stochastic nature of the data, one cannot identify the exact set of the statistically robust patterns analyzing a sequence of samples, i.e., the datasets, taken from the generative processes, and has to resort to approximations. We then propose GROSSO, an algorithm to find a rigorous approximation of the statistically robust patterns that does not contain false positives with high probability. We apply our framework to the mining of statistically robust sequential patterns. Our extensive evaluation on pseudo-artificial and real data shows that GROSSO provides high-quality approximations for the problem of mining statistically robust sequential patterns.**

***Index Terms*—statistically robust patterns, sequential patterns, VC-dimension, statistically-sound pattern mining**

I. INTRODUCTION

Frequent pattern mining [1] is one of the fundamental tasks in data mining, and requires to identify all patterns appearing in fractions at least θ of all transactions from a transactional dataset. Several variants of the problem have explored different types of patterns (from itemsets [2] to sequential patterns [3], to subgroups [4], to graphlets [5]) relevant to applications ranging from market basket analysis to recommendation systems to spam detection.

In several real applications, a pattern is studied in the context of a *sequence of datasets*, where the sequence is given, for example, from the collection of the data at different time points. For example, in market basket analysis, it is natural to study the patterns (e.g., itemsets) in datasets obtained from transactions in different weeks or months. In almost all applications, one can assume that each dataset is obtained from a *generative process* on transactions, which generates transactions according to some probability distribution, as assumed by *statistically-sound pattern mining* [6]. Consider, for example, a series of n surveys performed in n different time intervals in a supermarket, where we collect the data of the receipts of the costumers. The goal of such surveys is to infer information on how the behavior of the entire customers population evolves, but, obviously, it is impossible to collect the receipts of the whole population. Thus, our datasets only represent a collection of samples from the whole population.

In such a scenario, patterns of interest are the ones whose probability of appearing in a transaction follows some well-specified trend (e.g., it increases, decreases, or is constant across datasets). In the survey example above, we may be interested in finding sequences of purchases (i.e., sequential patterns) which become more and more common in time to understand how the customers' behavior changes over time. However, the identification of such patterns is extremely challenging, since the underlying probability distributions are unknown and the observed frequencies of the patterns in the data only approximately reflect such probabilities, considering the same trends at the level of observed frequencies leads to reporting several false positives. This problem is exacerbated by the huge number of potential candidates, which poses a severe *multiple hypothesis correction problem* [7]. In addition, techniques developed for significant pattern mining [6] or for statistically emerging pattern mining [8] can only be applied to (a sequence of) two datasets.

To address such challenges, in this work we introduce a novel framework to identify *statistically robust patterns* from a sequence of datasets, i.e., patterns whose probability of appearing in transactions follows a well-specified trend, while providing guarantees on the quality of the reported patterns in terms of false positives.

A. Our Contributions

In this work, we introduce the problem of mining *statistically robust patterns* from a *sequence of datasets*. In this regard, our contributions are:

- We define the problem of mining statistically robust patterns, and define an approximation of such patterns that does not contain *false positives*. We also describe three general types of patterns (emerging, descending, and stable) which are of interest in most scenarios.
- We introduce an algorithm, GROSSO, to obtain a rigorous approximation, without false positives, of the statistically

robust patterns from a sequence of datasets with probability at least $1 - \delta$, where δ is a confidence parameter set by the user. Our strategy is based on the concept of maximum deviation and can employ any uniform convergence bound. We show how such strategy can be used to approximate the three types of statistically robust patterns we introduced.
- We apply the general framework of statistically robust patterns to mine sequential patterns. We also introduce a novel algorithm to compute an upper bound on the capacity of a sequence that can be used to bound the maximum deviation using the statistical learning concept of VC-dimension, which may be of independent interest.
- We perform an extensive experimental evaluation, mining statistically robust sequential patterns from pseudo-artificial and real data. Our evaluation proves that relying on frequency alone leads to several spurious discoveries, while GRoSSo provides high-quality approximations.

B. Related Works

We now discuss the relationship of our work to prior art on significant pattern mining, emerging pattern mining, and robust pattern mining, which are the areas most related to our work. We also focus on works that considered sequential pattern mining, which is the application of our framework that we present in this paper, and that use concepts from statistical learning theory, as done in our work.

In significant pattern mining the dataset is seen as a sample from an unknown distribution and one is interested in finding patterns significantly deviating from an assumed null distribution (or hypothesis). Many variants and algorithms have been proposed for the problem. We point interested reader to the survey [6] and the recent works [9]–[11]. Few works have been proposed to mine statistically significant sequential patterns [12]–[14]. These methods are orthogonal to our approach, which focuses on finding patterns whose frequencies with respect to (w.r.t.) underlying generative distributions respect well defined conditions through a sequence of datasets.

The first work that proposed the problem of mining emerging patterns is [15]. To the best of our knowledge, the only work that considers the problem of finding emerging patterns considering a data generative process and provides statistical guarantees is [8]. However, the proposed approach only works with two datasets and only finds patterns with significant differences in the two datasets. Instead, our approach describes more general trends of the probabilities of the patterns and considers more than two datasets, and it is unclear whether the approach of [8] can be modified to work in our scenario.

Since the introduction of the frequent sequential pattern mining problem [3], several algorithms have been proposed for this task (e.g., [16]–[18]). Reference [19] is the first work that applies the statistical learning theory concept of VC-dimension to sequential patterns, and it provides the first computable efficient upper bound on the empirical VC-dimension of sequential patterns, based on the notion of *capacity* of a sequence. In this work, we propose a tighter upper bound on the capacity of a sequence to compute it and we apply it in a different scenario.

More recently, [20] provides a sampling based algorithm to compute approximations for the frequent sequential patterns problem, based on an upper bound to the VC-dimension of sequential patterns. It is also the first work to consider the problem of mining *true frequent sequential patterns*, that are frequent sequential patterns w.r.t. an underlying generative process. They propose two approaches to compute approximations of such problem: one based on the empirical VC-dimension and the other based on the Rademacher complexity. While we use a general framework similar to the one proposed by [20], we consider the problem of mining statistically robust patterns in a sequence of datasets, that is a different task. Reference [21] is the first work that considers the extraction of frequent patterns w.r.t. an underlying generative process, based on the concept of empirical VC-dimension of itemsets but their solution is tailored to itemsets and, thus, not applicable to sequential patterns.

Few works have been proposed to mine robust patterns, where the robustness is usually defined by constraints between the relation of the observed frequency of a pattern in a dataset and the frequencies of its sub- or super-patterns. For example, [22] defines robust patterns as patterns for which, by removing some of their sub-patterns, the ratio between its original frequency and the frequency of the resulting pattern in a dataset is greater than a user defined parameter. Reference [23] introduces a space of rules patterns model and it defines a Bayesian criterion for evaluating the interest of sequential patterns for mining sequential rule patterns for classification purpose. Differently from our work, these contributions focus on a single dataset and do not consider a dataset as a collection of samples from an unknown generative process.

II. Preliminaries

We now provide the definitions and the concepts used throughout the paper. We start by introducing the task of pattern mining and defining the problems of mining frequent and true frequent patterns. Then, we formally define the concept of maximum deviation required by our strategy to find an approximation of the statistically robust patterns.

A. Pattern Mining

Let a dataset $\mathcal{D} = \{\tau_1, \tau_2, ..., \tau_m\}$ be a finite bag of $|\mathcal{D}| = m$ *transactions*, where each transaction is an element from a domain $\mathbb{U}$. We assume that the elements of $\mathbb{U}$ exhibit a *poset* structure. We define a pattern p as an element of $\mathbb{U}$, potentially with some constraints. (For example, in itemset mining the domain $\mathbb{U}$ consists of all subsets of binary features called items.) A pattern p *belongs* to a transaction $\tau \in \mathcal{D}$ if and only if p is contained in τ, denoted by $p \sqsubseteq \tau$. The *support set* $T_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the set of transactions in $\mathcal{D}$ containing p: $T_{\mathcal{D}}(p) = \{\tau \in \mathcal{D} : p \sqsubseteq \tau\}$. Finally, the *frequency* $f_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the *fraction* of transactions in $\mathcal{D}$ to which p belongs: $f_{\mathcal{D}}(p) = \frac{|T_{\mathcal{D}}(p)|}{|\mathcal{D}|}$.

Given a dataset $\mathcal{D}$ and a *minimum frequency threshold* $\theta \in (0,1]$, *frequent pattern (FP) mining* is the task of reporting the set $FP(\mathcal{D},\theta)$ of all the patterns whose frequencies in $\mathcal{D}$ are at least θ, and their frequencies: $FP(\mathcal{D},\theta) = \{(p, f_{\mathcal{D}}(p)) : p \in \mathbb{U}, f_{\mathcal{D}}(p) \geq \theta\}$.

B. True Frequent Pattern Mining

In several applications, the dataset $\mathcal{D}$ is a sample of transactions independently drawn from an unknown probability distribution π on $\mathbb{U}$, that is, the dataset $\mathcal{D}$ is a finite bag of $|\mathcal{D}|$ *independent identically distributed* (i.i.d.) samples from π, with $\pi : \mathbb{U} \to [0,1]$. The *true support set* $T(p)$ of p is the set of patterns in $\mathbb{U}$ to which p belongs: $T(p) = \{\tau \in \mathbb{U} : p \sqsubseteq \tau\}$, and the *true frequency* $t_\pi(p)$ of p w.r.t. π is the probability that a transaction sampled from π contains p: $t_\pi(p) = \Pr_{\tau\sim\pi}(p \sqsubseteq \tau)$. In such a scenario, the final goal of the data mining process on $\mathcal{D}$ is to gain a better understanding of the process that generated the data, i.e., the distribution π, through the true frequencies of the patterns, which are unknown and only approximately reflected in the dataset $\mathcal{D}$. Thus, given a probability distribution π on $\mathbb{U}$ and a minimum frequency threshold $\theta \in (0,1]$, *true frequent pattern (TFP) mining* is the task of reporting the set $TFP(\pi,\theta)$ of all patterns whose true frequencies w.r.t. π are at least θ, and their true frequencies: $TFP(\pi,\theta) = \{(p, t_\pi(p)) : p \in \mathbb{U}, t_\pi(p) \geq \theta\}$. Note that, given a finite number of random samples from π, the dataset $\mathcal{D}$, it is not possible to find the exact set $TFP(\pi,\theta)$, and one has to resort to approximations of $TFP(\pi,\theta)$.

C. Maximum Deviation

Let $\mathcal{X}$ be a domain set and let $\mathcal{P}$ be a probability distribution on $\mathcal{X}$. Let $\mathcal{G}$ be a set of functions from $\mathcal{X}$ to $[0,1]$. Given a function $g \in \mathcal{G}$, the *expectation* of g is defined as $\mathbb{E}(g) = \mathbb{E}_{x\sim\mathcal{P}}[g(x)]$, with $x \in \mathcal{X}$, and, given a sample A of $|A|$ elements drawn from $\mathcal{P}$, the *empirical average* of g on A is defined as $E(g,A) = \frac{1}{|A|}\sum_{x_i\in A} g(x_i)$. The *maximum deviation* is defined as the largest difference, over all functions $g \in \mathcal{G}$, between the expectation of g and its empirical average on a sample A, that is, $\sup_{g\in\mathcal{G}} |\mathbb{E}(g) - E(g,A)|$.

In the TFP mining task, one is interested in finding good estimates for $t_\pi(p)$ simultaneously for each pattern $p \in \mathbb{U}$. In such a scenario, the true frequency $t_\pi(p)$ and the frequency $f_{\mathcal{D}}(p)$ of a pattern p on $\mathcal{D}$ represent, respectively, the expectation and the empirical average of a function associated with p, since $t_\pi(p) = \mathbb{E}_{\tau\sim\pi}[\mathbb{1}_\tau(p)]$ and $f_{\mathcal{D}}(p) = \frac{1}{|\mathcal{D}|}\sum_{\tau_i\in\mathcal{D}} \mathbb{1}_{\tau_i}(p)$, with $\mathbb{1}_\tau(p)$ the indicator function that assumes the value 1 if and only if $p \sqsubseteq \tau$. Thus, in the TFP scenario the maximum deviation is $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)|$, and one is interested in finding probabilistic upper bounds on such measure, i.e., finding a $\mu \in (0,1)$ such that (s.t.) $\Pr(\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu) \geq 1-\delta$, with a confidence parameter $\delta \in (0,1)$.

Such probabilistic upper bounds on the maximum deviation can be computed with tools from statistical learning theory, e.g., VC-dimension [25] and Rademacher complexity [26]. More common techniques, e.g., Hoeffding inequality and union bounds, instead do not provide useful results since they require to know the number of all possible patterns that can be generated from the process, which can be infinite or impractical to compute.

III. Statistically Robust Pattern Mining

In this work, we introduce the task of mining *statistically robust patterns* (SRP) from a *sequence of datasets*. Let us consider the scenario in which we have a sequence $\mathcal{D}_1^n = \{\mathcal{D}_1, \mathcal{D}_2, .., \mathcal{D}_n\}$ of n datasets, where each dataset $\mathcal{D}_i$ is a bag of $|\mathcal{D}_i|$ i.i.d. samples taken from a probability distribution π_i on $\mathbb{U}$, with $i \in [1,n]$. Let $\Pi_1^n = \{\pi_1, \pi_2, .., \pi_n\}$ denote the set of the n probability distributions and $\mathcal{T}_p = \{t_{\pi_1}(p), t_{\pi_2}(p), .., t_{\pi_n}(p)\}$ the set of the true frequencies of the pattern p w.r.t. Π_1^n. In such a scenario, we are interested in finding patterns whose true frequencies w.r.t. Π_1^n respect a well defined *condition* $cond(\mathcal{T}_p)$ that describes the evolution of their true frequencies through the sequence. For example, one may be interested in finding patterns whose true frequencies are almost the same in all the probability distributions, or patterns whose true frequencies always increase/decrease, and so on. So, given n probabilities distribution $\Pi_1^n = \{\pi_1, \pi_2, .., \pi_n\}$, a condition $cond(\mathcal{T}_p)$ on the true frequencies $\mathcal{T}_p$ that defines the patterns we are interested in, *statistically robust pattern mining* is the task of reporting the set $SRP(\Pi_1^n)$ of all patterns whose true frequencies w.r.t Π_1^n respect $cond(\mathcal{T}_p)$: $SRP(\Pi_1^n) = \{(p, \mathcal{T}_p) : p \in \mathbb{U} \wedge cond(\mathcal{T}_p) = 1\}$.

Similarly to TFP mining, from a sequence of samples (the datasets $\mathcal{D}_1^n$) it is not possible to find the exact set $SRP(\Pi_1^n)$. Thus, one has to resort to approximations. Denoting by $\mathcal{F}_p = \{f_{\mathcal{D}_1}(p), f_{\mathcal{D}_2}(p), ..., f_{\mathcal{D}_n}(p)\}$ the set of the n frequencies of p in $\mathcal{D}_1^n$, we define a *false positives free (FPF) approximation* $\mathcal{A}$ of $SRP(\Pi_1^n)$ as: $\mathcal{A} = \{(p, \mathcal{F}_p) : \exists (p, \mathcal{T}_p) \in SRP(\Pi_1^n)\}$. The approximation $\mathcal{A}$ does not contain *false positives*, that is, patterns $p \notin SRP(\Pi_1^n)$.

We define three general types of patterns that can be described by the SRPs framework, and that we consider in the rest of this work.

1) Emerging Patterns (EP): these are patterns whose true frequencies always increase over the sequence, i.e., patterns p for which $t_{\pi_{i+1}}(p) > t_{\pi_i}(p) + \varepsilon$, for all $i \in [1,n)$, for some given *emerging threshold* $\varepsilon \in [0,1)$. Formally, given an emerging threshold $\varepsilon \in [0,1)$, we define the *emerging condition* $cond^E(\mathcal{T}_p)$ as:

$$cond^E(\mathcal{T}_p) = \begin{cases} 1 & \text{if } t_{\pi_{i+1}}(p) > t_{\pi_i}(p) + \varepsilon, \forall i \in [1,n) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

2) Descending Patterns (DP): these are patterns p whose true frequencies always decrease over the sequence, i.e., patterns p for which $t_{\pi_i}(p) > t_{\pi_{i+1}}(p) + \varepsilon$ for all $i \in [1,n)$, for some given emerging threshold $\varepsilon \in [0,1)$. Formally, given an emerging threshold $\varepsilon \in [0,1)$, we define the *descending condition* $cond^D(\mathcal{T}_p)$ as:

$$cond^D(\mathcal{T}_p) = \begin{cases} 1 & \text{if } t_{\pi_i}(p) > t_{\pi_{i+1}}(p) + \varepsilon, \forall i \in [1,n) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

3) Stable Patterns (SP): these are patterns whose true frequencies in the n probability distributions are above a minimum frequency threshold θ and do not change too much. In particular, we consider patterns p for which $|t_{\pi_i}(p) - t_{\pi_j}(p)| \leq \alpha$ and $t_{\pi_i}(p) \geq \theta$ for all $i \neq j \in [1, n]$, for some given *error threshold* $\alpha \in (0, 1)$ and a minimum frequency threshold $\theta \in (0, 1)$. Formally, given an error threshold $\alpha \in (0, 1)$ and minimum frequency threshold $\theta \in (0, 1)$, we define the *stability condition* $cond^S(\mathcal{T}_p)$ as:

$$cond^S(\mathcal{T}_p) = \begin{cases} 1 & \text{if } |t_{\pi_i}(p) - t_{\pi_j}(p)| \leq \alpha \wedge\ t_{\pi_i}(p) \geq \theta, \\ & \quad \forall\, i \neq j \in [1, n] \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Note that many more types of patterns can be described by our proposed framework. For example, one may be interested in patterns whose true frequencies in the different probability distributions have a ratio larger than a user-defined constant, or may be interested in patterns whose true frequencies are stable in some distributions and then increase/decrease in others, or that first increase and then decrease, and so on. In addition, for the EP and DP tasks, we provided general conditions to describe such patterns, while one may also consider constraints using a minimum frequency threshold θ.

IV. GROSSO: APPROXIMATING THE STATISTICALLY ROBUST PATTERNS

In this section, we describe GROSSO, our strategy to provide a rigorous approximation to the SRPs. In particular, GROSSO aims to find an approximation that does not contain false positives (i.e., a FPF approximation, see Section III) with high probability. (GROSSO can be extended to find approximations with guarantees on false negatives; the details will appear in the journal version of this work.) We then show how to apply such strategy to mine approximations to the three types of SRPs we defined in the previous section.

For a fixed $cond(\mathcal{T}_p)$ that defines the SRPs we are interested in, and given the sequence $\mathcal{D}_1^n$ of n datasets and a confidence parameter $\delta \in (0, 1)$, we start computing an upper bound μ_i on the maximum deviation w.r.t π_i for each dataset $\mathcal{D}_i$, i.e., $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$, with $i \in [1, n]$. Each upper bound is computed using confidence δ/n, thus $\Pr(\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i) \geq 1 - \delta/n$, $\forall\, i \in [1, n]$. We denote by $\mu_1^n = \{\mu_1, \mu_2, .., \mu_n\}$ the set of the n upper bounds on the maximum deviations. Since $cond(\mathcal{T}_p)$ considers the true frequencies $\mathcal{T}_p$, which are unknown, we need to define a new condition $cond(\mathcal{F}_p, \mu_1^n)$ on the frequencies $\mathcal{F}_p$ and on the upper bounds μ_1^n. Such new condition takes into account the uncertainty of the data in our samples, i.e., the datasets, and, for a pattern p, it must be $cond(\mathcal{T}_p) = 0 \implies cond(\mathcal{F}_p, \mu_1^n) = 0$ if $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$ holds $\forall i \in [1, n]$. Then, we aim to find a starting set of possible candidates. For each dataset $\mathcal{D}_i$, we compute the minimum frequency threshold $\tilde{\theta}_i$ which the patterns must have in such dataset to verify $cond(\mathcal{F}_p, \mu_1^n)$. We then mine the dataset $\mathcal{D}_k$, where $k = \arg\max_{i \in [1,n]} \tilde{\theta}_i$, with the corresponding minimum frequency threshold $\tilde{\theta}_k$, obtaining the set $\mathcal{B} = FP(\mathcal{D}_k, \tilde{\theta}_k)$ of the starting candidates. The idea is to mine the dataset with the highest minimum frequency threshold in order to obtain a starting set of possible candidates that is as small as possible. Finally, we explore the remaining datasets. For each $\mathcal{D}_i \in \mathcal{D}_1^n / \mathcal{D}_k$ and for each $p \in \mathcal{B}$, we compute its frequency $f_{\mathcal{D}_i}(p)$ in $\mathcal{D}_i$ and check whether $cond(\mathcal{F}_p, \mu_1^n) = 1$, considering the frequencies that have already been computed. If $cond(\mathcal{F}_p, \mu_1^n) = 0$, we cannot prove that $cond(\mathcal{T}_p) = 1$, and we remove such pattern from the set of the possible candidates. Algorithm 1 shows the pseudocode of GROSSO.

Algorithm 1: GROSSO: find an approximation $\mathcal{A}$ of $SRP(\Pi_1^n)$.

Data: Datasets $\mathcal{D}_1^n$, $\delta \in (0, 1)$.
Result: Set $\mathcal{A}$ that is a FPF approx. of $SRP(\Pi_1^n)$ with probability $\geq 1 - \delta$.

```
foreach D_i ∈ D_1^n do
    μ_i ← computeMaxDev(D_i, δ/n);
    θ̃_i ← minimum frequency threshold for D_i;
B ← FP(D_k, θ̃_k), with k = arg max_{i∈[1,n]} θ̃_i;
A ← ∅;
foreach (p, f_{D_k}(p)) ∈ B do
    F_p ← empty array of n elements;
    F_p[k] ← f_{D_k}(p); /* F_p[k]: k-th element of F_p */
    A ← A ∪ (p, F_p);
foreach D_i ∈ D_1^n / D_k do
    foreach (p, F_p) ∈ A do
        F_p[i] ← computeFrequency(D_i, p);
        if cond(F_p, μ_i^n) = 0 then
            A ← A/(p, F_p);
return A;
```

Theorem 1. *The set $\mathcal{A}$ returned by* GROSSO *is a FPF approximation of $SRP(\Pi_1^n)$ with probability $\geq 1 - \delta$.*

Proof. From the definition of $cond(\mathcal{F}_p, \mu_1^n)$, we know that $cond(\mathcal{T}_p) = 0 \implies cond(\mathcal{F}_p, \mu_1^n) = 0$ if $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$ holds $\forall\, i \in [1, n]$. In such a scenario, only the patterns $p \in SRP(\Pi_1^n)$ can appear in $\mathcal{A}$, and thus $\mathcal{A}$ is a FPF approximation of $SRP(\Pi_1^n)$. Now, let us define the event E_i as the event in which $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| > \mu_i$, with $i \in [1, n]$. From the choice of the confidence parameter used to compute the upper bounds on the maximum deviation, we know that $\Pr(E_i) < \delta/n$. So, we have $\Pr(\exists i \in [1, n] : \sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| > \mu_i) = \Pr(\cup_{i=1}^n E_i) \leq \sum_{i=1}^n \Pr(E_i) < \delta$. Thus, the set $\mathcal{A}$ returned by GROSSO is a FPF approximation of $SRP(\Pi_1^n)$ with probability $\geq 1 - \delta$, which concludes our proof. □

A. Approximating the EP

Now, we apply the strategy defined above to find an approximation of the EP. Starting from $cond^E(\mathcal{T}_p)$ (Equation 1), we

define $cond^E(\mathcal{F}_p, \mu_1^n)$ as:

$$cond^E(\mathcal{F}_p, \mu_1^n) = \begin{cases} 1 & \text{if } f_{\mathcal{D}_{i+1}}(p) - \mu_{i+1} - (f_{\mathcal{D}_i}(p) + \mu_i) > \varepsilon, \\ & \quad \forall\, i \in [1, n) \\ 0 & \text{otherwise.} \end{cases}$$

For a given $i \in [1, n)$, such condition represents the scenario in which $t_{\pi_{i+1}}(p)$ and $t_{\pi_i}(p)$ assume the values $f_{\mathcal{D}_{i+1}}(p) - \mu_{i+1}$ and $f_{\mathcal{D}_i}(p) + \mu_i$, respectively, that are the values at which their distance is minimum over all possible values that they can assume. Only if such condition is true, we can prove that $t_{\pi_{i+1}}(p) > t_{\pi_i}(p) + \varepsilon$. Then, starting from such condition, we compute the minimum frequency threshold for each dataset. Since it must be $f_{\mathcal{D}_2}(p) - \mu_2 > f_{\mathcal{D}_1}(p) + \mu_1 + \varepsilon$ and $f_{\mathcal{D}_3}(p) - \mu_3 > f_{\mathcal{D}_2}(p) + \mu_2 + \varepsilon$, and thus $f_{\mathcal{D}_3}(p) > f_{\mathcal{D}_1}(p) + 2 \cdot \varepsilon + \mu_1 + \mu_3 + 2 \cdot \mu_2$, iterating such reasoning for all the n datasets and considering $f_{\mathcal{D}_1}(p) \geq 0$, we obtain the minimum frequency threshold $\tilde{\theta}_n^E = (n-1) \cdot \varepsilon + \mu_1 + \mu_n + \sum_{i=2}^{n-1} 2 \cdot \mu_i$ for the dataset $\mathcal{D}_n$, the highest over all the n datasets. Thus, the set $FP(\mathcal{D}_n, \tilde{\theta}_n^E)$ provides the starting candidates. Finally, starting from $\mathcal{D}_{n-1}$ and ending with $\mathcal{D}_1$, we analyze the remaining datasets and check whether the candidates verify $cond^E(\mathcal{F}_p, \mu_1^n)$.

Note that if one is interested in patterns with a true frequency above a value $\theta \in (0, 1)$, i.e., $t_{\pi_i}(p) \geq \theta$, $\forall\, i \in [1, n]$, the following strategy can be used to reduce the set of starting candidates. Since we require that $f_{\mathcal{D}_1}(p) \geq \theta + \mu_1$ to discard possible false positives, a factor $\theta + \mu_1$ must be added to $\tilde{\theta}_n^E$. Instead, if one is interested in patterns p with $t_{\pi_n}(p) \geq \theta$, the minimum frequency for dataset $\mathcal{D}_n$ is $\tilde{\theta}_n^E = \max\{(n-1) \cdot \varepsilon + \mu_1 + \mu_n + \sum_{i=2}^{n-1} 2 \cdot \mu_i,\ \theta + \mu_n\}$.

Theorem 2. $cond^E(\mathcal{T}_p) = 0 \implies cond^E(\mathcal{F}_p, \mu_1^n) = 0$.

Proof. Let us consider that $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$, $\forall\, i \in [1, n]$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i, f_{\mathcal{D}_i}(p) + \mu_i]$, $\forall\, i \in [1, n]$. Let p' be a pattern s.t. $cond^E(\mathcal{T}_{p'}) = 0$. From Equation 1, there is at least a couple of consecutive distribution π_j, π_{j+1}, with $j \in [1, n)$, s.t. $t_{\pi_{j+1}}(p') \leq t_{\pi_j}(p') + \varepsilon$. Since we know that $t_{\pi_{j+1}}(p') \in [f_{\mathcal{D}_{j+1}}(p') - \mu_{j+1}, f_{\mathcal{D}_{j+1}}(p') + \mu_{j+1}]$ and that $t_{\pi_j}(p') \in [f_{\mathcal{D}_j}(p') - \mu_j, f_{\mathcal{D}_j}(p') + \mu_j]$, the condition $f_{\mathcal{D}_{j+1}}(p') - \mu_{j+1} - (f_{\mathcal{D}_j}(p') + \mu_j) > \varepsilon$, cannot be verified for such p', and thus $cond^E(\mathcal{F}_p, \mu_1^n) = 0$, which concludes our proof. □

B. Approximating the DP

Using the same approach proposed to approximate the EP, it is possible to approximate the DP. Starting from $cond^D(\mathcal{T}_p)$ (Equation 2), we define $cond^D(\mathcal{F}_p, \mu_1^n)$ as:

$$cond^D(\mathcal{F}_p, \mu_1^n) = \begin{cases} 1 & \text{if } f_{\mathcal{D}_i}(p) - \mu_i - (f_{\mathcal{D}_{i+1}}(p) + \mu_{i+1}) > \varepsilon, \\ & \quad \forall\, i \in [1, n) \\ 0 & \text{otherwise.} \end{cases}$$

Iterating such condition for all the n datasets, we obtain the minimum frequency threshold $\tilde{\theta}_1^D = \tilde{\theta}_n^E$ for the dataset $\mathcal{D}_1$, that is the highest over all the n datasets. Thus, the set $FP(\mathcal{D}_1, \tilde{\theta}_1^D)$ provides the starting candidates. Finally, starting from $\mathcal{D}_2$ and ending with $\mathcal{D}_n$, we analyze the remaining datasets and check whether the candidates verify $cond^D(\mathcal{F}_p, \mu_1^n)$. In the case of a minimum frequency threshold $\theta \in (0, 1)$, reasoning analogous to the EP can be applied.

Theorem 3. $cond^D(\mathcal{T}_p) = 0 \implies cond^D(\mathcal{F}_p, \mu_1^n) = 0$.

The proof is analogous to the proof of Theorem 2.

C. Approximating the SP

Finally, we apply the strategy defined above to find an approximation of the SP. Starting from $cond^S(\mathcal{T}_p)$ (Equation 3), we define $cond^S(\mathcal{F}_p, \mu_1^n)$ as:

$$cond^S(\mathcal{F}_p, \mu_1^n) = \begin{cases} 1 & \text{if } f_{\mathcal{D}_i}(p) + \mu_i - (f_{\mathcal{D}_j}(p) - \mu_j) \leq \alpha \\ & \quad \wedge\, f_{\mathcal{D}_j}(p) + \mu_j - (f_{\mathcal{D}_i}(p) - \mu_i) \leq \alpha, \\ & \quad \wedge\, f_{\mathcal{D}_i}(p) - \mu_i \geq \theta, \\ & \quad \forall\, i \neq j \in [1, n] \\ 0 & \text{otherwise.} \end{cases}$$

Given $i \neq j \in [1, n]$, the first two conditions represent the scenario in which $t_{\pi_i}(p)$ and $t_{\pi_j}(p)$ assume the values $f_{\mathcal{D}_i}(p) - \mu_i$ and $f_{\mathcal{D}_j}(p) + \mu_j$, respectively, if $f_{\mathcal{D}_i}(p) < f_{\mathcal{D}_j}(p)$, or respectively the values $f_{\mathcal{D}_j}(p) - \mu_j$ and $f_{\mathcal{D}_i}(p) + \mu_i$ if $f_{\mathcal{D}_j}(p) < f_{\mathcal{D}_i}(p)$, that are the values at which their distance is maximum over all possible values that they can assume. Only if such conditions are true, we can prove that $|t_{\pi_i}(p) - t_{\pi_j}(p)| \leq \alpha$. The third condition, instead, represents the scenario in which $t_{\pi_i}(p)$ assumes the value $f_{\mathcal{D}_i}(p) - \mu_i$, that is the minimum value that it can assume. Only if such condition is true, we can prove that $t_{\pi_i}(p) \geq \theta$. The only condition that affects the minimum frequency thresholds $\tilde{\theta}_i^S$ is $f_{\mathcal{D}_i}(p) \geq \theta + \mu_i$, $\forall\, i \in [1, n]$. So, we have $\tilde{\theta}_i^S = \theta + \mu_i$, $\forall\, i \in [1, n]$, and the set $FP(\mathcal{D}_k, \tilde{\theta}_k^S)$, with $k = \arg\max_{i \in [1,n]} \tilde{\theta}_i^S$, provides the starting candidates. Finally, we analyze the remaining datasets $\mathcal{D}_i \in \mathcal{D}_1^n / \mathcal{D}_k$ and check whether the candidates verify $cond^S(\mathcal{F}_p, \mu_1^n)$.

Theorem 4. $cond^S(\mathcal{T}_p) = 0 \implies cond^S(\mathcal{F}_p, \mu_1^n) = 0$.

Proof. Let us consider that $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$, $\forall\, i \in [1, n]$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i, f_{\mathcal{D}_i}(p) + \mu_i]$, $\forall\, i \in [1, n]$. Let p' be a pattern s.t. $cond^S(\mathcal{T}_{p'}) = 0$. From Equation 3, there is at least a distribution π_i, with $i \in [1, n]$ s.t. $t_{\pi_i}(p') < \theta$ and/or there is at least a couple of distributions π_k, π_j, with $k \neq j \in [1, n]$, s.t. $|t_{\pi_j}(p') - t_{\pi_k}(p')| > \alpha$. First, let us consider the case in which there is a distribution π_i, with $i \in [1, n]$, s.t. $t_{\pi_i}(p') < \theta$. Since we know that $t_{\pi_i}(p') \in [f_{\mathcal{D}_i}(p') - \mu_i, f_{\mathcal{D}_i}(p') + \mu_i]$, the condition $f_{\mathcal{D}_i}(p') - \mu_i \geq \theta$ cannot be verified, and thus $cond^S(\mathcal{F}_p, \mu_1^n) = 0$. Now, let us consider the case in which there is a couple of distributions π_k, π_j, with $k \neq j \in [1, n]$ s.t. $|t_{\pi_j}(p') - t_{\pi_k}(p')| > \alpha$. Since we know that $t_{\pi_j}(p') \in [f_{\mathcal{D}_j}(p') - \mu_j, f_{\mathcal{D}_j}(p') + \mu_j]$ and that $t_{\pi_k}(p') \in [f_{\mathcal{D}_k}(p') - \mu_k, f_{\mathcal{D}_k}(p') + \mu_k]$, the condition $f_{\mathcal{D}_j}(p') + \mu_j - (f_{\mathcal{D}_k}(p') - \mu_k) \leq \alpha$ cannot be verified, if $f_{\mathcal{D}_j}(p') > f_{\mathcal{D}_k}(p')$, while the condition $f_{\mathcal{D}_k}(p') + \mu_k - (f_{\mathcal{D}_j}(p') - \mu_j) \leq \alpha$ cannot be verified if $f_{\mathcal{D}_j}(p') < f_{\mathcal{D}_k}(p')$, and thus $cond^S(\mathcal{F}_p, \mu_1^n) = 0$, which concludes our proof. □

V. Application: Mining Statistically Robust Sequential Patterns

In this section, we introduce the task of sequential pattern mining, as a concrete realization of the general framework of pattern mining we introduced in Section II-A. Then, we introduce the statistical learning theory concept of VC-dimension and we apply it to sequential patterns. We introduce a novel algorithm to compute an upper bound on the capacity of a sequence and we use such algorithm to compute an upper bound on the empirical VC-dimension of sequential patterns. Finally, we provide a VC-dimension based strategy to bound the maximum deviation of the true frequencies of sequential patterns, which can be used in the SRP mining scenario.

A. Sequential Pattern Mining

Let $\mathcal{I} = \{i_1, i_2, ..., i_p\}$ be a finite set of items. An *itemset* X is a non-empty subset of $\mathcal{I}$, i.e., $X \subseteq \mathcal{I}$, $X \neq \emptyset$. A *sequential pattern* (or *sequence*) $s = \langle S_1, S_2, ..., S_k \rangle$ is a finite ordered sequence of itemsets, with $S_i \subseteq \mathcal{I}$, $S_i \neq \emptyset$ for all $i \in [1, k]$. We say that such sequence s is *built on* $\mathcal{I}$ and we denote by $\mathbb{S}$ the set of all such sequences. The *length* $|s|$ of s is the number of itemsets in s. The *item-length* $||s||$ of s is the sum of the sizes of the itemsets in it, i.e., $||s|| = \sum_{i=1}^{|s|} |S_i|$, where the size $|S_i|$ of an itemset S_i is the number of items in it. A sequential pattern $y = \langle Y_1, Y_2, ..., Y_a \rangle$ is a *subsequence* of an other sequential pattern $w = \langle W_1, W_2, ..., W_b \rangle$, denoted by $y \sqsubseteq w$, if and only if there exists a sequence of naturals $1 \leq i_1 < i_2 < ... < i_a \leq b$ s.t. $Y_1 \subseteq W_{i_1}, Y_2 \subseteq W_{i_2}, ..., Y_a \subseteq W_{i_a}$. Note that an item can occur only once in an itemset, but it can occur multiple times in different itemsets of the same sequence. The *capacity* $c(s)$ of a sequence s is the number of distinct subsequences of s: $c(s) = |\{a : a \sqsubseteq s\}|$.

Example 1. *The sequential pattern $y = \langle \{2, 6, 7\}, \{2\} \rangle$ has length $|y| = 2$, item-length $||y|| = 4$ and capacity $c(s) = 14$. It is a subsequence of $w = \langle \{2, 4, 6, 7\}, \{8, 7\}, \{2, 7\} \rangle$ but not of $z = \langle \{2\}, \{6, 7\}, \{2, 7\} \rangle$ or $q = \langle \{2\}, \{2, 6, 7\} \rangle$.*

B. VC-Dimension of Sequential Patterns

The Vapnik-Chervonenkis (VC) dimension [25], [27] of a space of points is a measure of the complexity or expressiveness of a family of indicator functions, or, equivalently of a family of subsets, defined on that space. A finite bound on the VC-dimension of a structure implies a bound of the number of random samples required to approximately learn that structure.

We define a range space as a pair $(X, \mathcal{R})$, where X is a finite or infinite set and $\mathcal{R}$, the *range set*, is a finite or infinite family of subsets of X. The members of X are called *points* while the members of $\mathcal{R}$ are called *ranges*. Given $A \subseteq X$, we define the *projection* of $\mathcal{R}$ in A as $P_{\mathcal{R}}(A) = \{r \cap A \ : \ r \in \mathcal{R}\}$. We define 2^A as the *power set* of A, that is the set of all the possible subsets of A, including the empty set $\emptyset$ and A itself. If $P_{\mathcal{R}}(A) = 2^A$, then A is said to be *shattered by* $\mathcal{R}$. The VC-dimension of a range space is the cardinality of the largest set shattered by the ranges.

Definition 1. *Let $RS = (X, \mathcal{R})$ be a range space and $B \subseteq X$. The empirical VC-dimension $EVC(RS, B)$ of RS on B is the maximum cardinality of a subset of B shattered by $\mathcal{R}$.*

The main application of VC-dimension in statistics and learning theory is to derive the sample size needed to approximate "learn" the ranges, as defined below.

Definition 2. *Let $RS = (X, \mathcal{R})$ be a range space and let γ be a probability distribution on X. Given $\mu \in (0, 1)$, a bag B of elements sampled from X according to γ is a μ-bag of (X, γ) if for all $r \in \mathcal{R}$, $\left| \Pr_\gamma(r) - \frac{|B \cap r|}{|B|} \right| \leq \mu$.*

A μ-bag of (X, γ) can be constructed sampling points from X according to the distribution γ, as follows.

Theorem 5 ([28])**.** *Let $RS = (X, \mathcal{R})$ be a range space and let γ be a probability distribution on X. Let B a bag of $|B|$ elements sampled from X according to γ and let d be the empirical VC-dimension $EVC(RS, B)$ of RS on B. Then, given $\delta \in (0, 1)$ and $\mu = \sqrt{\frac{1}{2|B|}\left(d + \ln \frac{1}{\delta}\right)}$, the bag B is a μ-bag of (X, γ) with probability at least $1 - \delta$.*

We now define the range space of sequential patterns.

Definition 3. *Let $\mathbb{S}$ be the set of all sequences that can be built on $\mathcal{I}$ and let π be a probability distribution on $\mathbb{S}$. We define $RS = (X, \mathcal{R})$ to be a range space associated with $\mathbb{S}$ w.r.t. π s.t.: i) $X = \mathbb{S}$; ii) $\mathcal{R} = \{T(s) : s \in \mathbb{S}\}$ is a family of sets of sequential transactions s.t. for each sequential pattern s the set $T(s) = \{\tau \in \mathbb{S} : s \sqsubseteq \tau\}$ is the true support set of s.*

Given a dataset $\mathcal{D}$, that is a finite bag of transactions sampled from π, we aim to compute the empirical VC-dimension $EVC(RS, \mathcal{D})$ of the range space associated with $\mathbb{S}$ w.r.t. π on the dataset $\mathcal{D}$ in order to find a probabilistic bound $\mu \in (0, 1)$ on the maximum deviation $\sup_{s \in \mathbb{S}} |t_\pi(s) - f_{\mathcal{D}}(s)|$. In particular, given the $EVC(RS, \mathcal{D})$ and using Theorem 5, it is possible to compute a $\mu \in (0, 1)$ s.t. $\mathcal{D}$ is a μ-bag of $(\mathbb{S}, \pi)$, and, from the definitions of μ-bag (Definition 2) and of range space of sequential patterns (Definition 3), this ensures that $\sup_{s \in \mathbb{S}} |t_\pi(s) - f_{\mathcal{D}}(s)| \leq \mu$.

The exact computation of the empirical VC-dimension $EVC(RS, \mathcal{D})$ on the dataset $\mathcal{D}$ is computationally expensive. The *s-index* introduced by Servan-Schreiber et al. [19] provides an efficiently computable upper bound on $EVC(RS, \mathcal{D})$.

Definition 4 (Definition 2 [19])**.** *Let $\mathcal{D}$ be a sequential dataset. The s-index of $\mathcal{D}$ is the maximum integer d s.t. $\mathcal{D}$ contains at least d different sequential transactions with capacity at least $2^d - 1$, s.t. no one of them is a subset of another, i.e., the d transactions form an anti-chain.*

C. New Upper Bound on the Capacity

Definition 4 requires to compute the capacity of each transaction $\tau \in \mathcal{D}$. The exact capacity $c(s)$ of a sequence s can be computed using the algorithm described in [29], but it is computationally expensive and may be prohibitive for large datasets. Thus, we are interested in efficiently computable

upper bounds on $c(s)$. A first naïve bound, that we denote by $\tilde{c}_n(s) \geq c(s)$, is given by $2^{||s||} - 1$, but it may be a loose upper bound since $c(s) = 2^{||s||} - 1$ if and only if all the items contained in all the itemsets of the sequence s are different.

The second upper bound has been introduced in [19]. Such upper bound, that we denote by $\tilde{c}(s) \geq c(s)$, can be computed as follows. When s contains, among others, two itemsets A and B s.t. $A \subseteq B$, subsequences of the form $\langle C \rangle$ with $C \subseteq A$ are considered twice in $2^{||s||} - 1$, "generated" once from A and once from B. To avoid over-counting such $2^{|A|} - 1$ subsequences, [19] proposes to consider only the ones "generated" from the longest itemset that can generate them.

In this work, we introduce a novel, tighter upper bound $\hat{c}(s) \geq c(s)$. Our upper bound is based on the following observation. Let itemsets A and B be respectively the *i-th* and *j-th* itemset of the sequence s with $i < j$, that is, A comes before B in s, and let $T = A \cap B \neq \emptyset$ be their intersection. Let D be a subset of the *bag-union* of the itemsets in s that come before A, that is $D \subseteq \bigcup_{S_k \in s : k < i} S_k$, and let E be a subset of the bag-union of the itemsets in s that come after B, that is $E \subseteq \bigcup_{S_\ell \in s : \ell > j} S_\ell$. The sequences of the form $\langle DCE \rangle$, with $C \subseteq T$, are also considered twice, for the same reasons explained above. Given $a = \sum_{k=1}^{i-1} |S_k|$ the sum of the sizes of the itemsets before A in the sequence s and $b = \sum_{\ell=j+1}^{|s|} |S_\ell|$ the sum of the sizes of the ones that come after B, the number of over-counted sequences of this form is $2^a \cdot (2^{|T|} - 1) \cdot 2^b$. Note that this new formula also includes the sequences of the form $\langle C \rangle$, since D and E may be the empty set.

An algorithm to compute an upper bound $\hat{c}(s)$ based on the observation above is the following (the pseudocode is shown in Algorithm 2). Let $s = \langle S_1, S_2, ..., S_{|S|} \rangle$ be a sequence and assume to *re-label* the itemsets in s by *increasing size*, ties broken arbitrarily, i.e., following the original order. Let $\hat{s} = \langle S_1, S_2, ..., S_{|\hat{s}|} \rangle$ be the sequence in the new order, s.t. $|S_i| \leq |S_{i+1}|, \forall\, i \in [1, |\hat{s}| - 1]$. Let $N = [n_1, n_2, ..., n_{|\hat{s}|}]$ be a vector s.t. its i-th element n_i is the sum of the sizes of the itemsets that in the original ordered sequence s come before the i-th itemset of the new ordered sequence $\hat{s}$. The inputs of our algorithm are the new ordered sequence $\hat{s}$ and the vector N. First, $\hat{c}(\hat{s})$ is set to $2^{||\hat{s}||} - 1$. For each itemset $S_i \in \hat{s}$, we check whether there exists an itemset S_j, with $j > i$, s.t. the set $T_{ij} = S_i \cap S_j$ is non-empty. For such S_j, we compute the number of over-counted subsequences with the formula above. After checking the entire sequence $\hat{s}$ for a single itemset S_i, we remove the maximum number of over-counted subsequences found for such S_i. Then, we update the vector N, subtracting the size of S_i from each n_m, if the itemset m comes after the itemset i in the original ordered sequence s.

Example 2. *Consider the following sequence* $s = \langle \{1\}, \{2, 5, 7\}, \{4\}, \{2, 3, 5\}, \{1, 8\} \rangle$. *The inputs of our algorithm are* $\hat{s} = \langle \{1\}, \{4\}, \{1, 8\}, \{2, 5, 7\}, \{2, 3, 5\} \rangle$ *and* $N = [0, 4, 8, 1, 5]$. *The naïve upper bound* $\tilde{c}_n(s)$ *is* $2^{10} - 1 = 1023$. *The upper bound* $\tilde{c}(s)$ *defined in [19] is 1022, since it only removes once the sequence* $\langle \{1\} \rangle$. *The upper bound* $\hat{c}(s)$ *obtained with our algorithm is* 1010, *since we remove the sequence* $\langle \{1\} \rangle$ *but also sequences generated by the intersection of* $\{2, 5, 7\}$ *and* $\{2, 3, 5\}$ *combined with other itemsets (e.g.,* $\langle \{2, 5\}, \{1, 8\} \rangle$*).*

Algorithm 2: Computation of the upper bound $\hat{c}(\hat{s})$.

Data: Sequence $\hat{s} = \langle S_1, S_2, .., S_{|\hat{s}|} \rangle$, with the $S'_i s$ labeled as described in the text, vector $N = [n_1, n_2, .., n_{|\hat{s}|}]$, with the $n'_i s$ computed as described in the text.
Result: Upper Bound $\hat{c}(\hat{s})$ to $c(s)$.

```
t ← ||ŝ||;
ĉ(ŝ) ← 2^t − 1;
for i ← 1 to |ŝ| − 1 do
    val ← 0;
    for j ← i + 1 to |ŝ| do
        if ∃ T = S_i ∩ S_j : T ≠ ∅ then
            val ← max{val, 2^min(n_i, n_j) · (2^|T| − 1) · 2^(t − max(n_i + |S_i|, n_j + |S_j|))};
    if val ≠ 0 then
        ĉ(ŝ) ← ĉ(ŝ) − val;
        t ← t − |S_i|;
        for m ← i + 1 to |ŝ| do
            if n_m > n_i then
                n_m ← n_m − |S_i|;
return ĉ(ŝ);
```

D. Bound to the Maximum Deviation

The following theorem allows to compute an upper bound on the maximum deviation using the s-index defined above.

Theorem 6 ([20])**.** *Let* $\mathcal{D}$ *be a finite bag of* $|\mathcal{D}|$ *i.i.d. samples from a probability distribution* π *on* $\mathbb{S}$ *and let* $\delta \in (0, 1)$. *Let* d *be the s-index of* $\mathcal{D}$. *If* $\mu = \sqrt{\frac{1}{2|\mathcal{D}|}\left(d + \ln \frac{1}{\delta}\right)}$, *then* $\sup_{s \in \mathbb{S}} |t_\pi(s) - f_{\mathcal{D}}(s)| \leq \mu$ *with probability at least* $1 - \delta$.

Theorem 6 shows how to compute an upper bound on the maximum deviation for the sequential patterns using the empirical VC-dimension. It requires to compute the s-index of the dataset $\mathcal{D}$, that can be computed using our algorithm to obtain upper bounds on the capacities of the transactions of $\mathcal{D}$. With such bound on the maximum deviation we can use GRoSSo to find FPF approximations of the statistically robust sequential patterns.

VI. Experimental Evaluation

In this section, we report the results of our experimental evaluation, on multiple pseudo-artificial and real datasets, to assess the performance of GRoSSo for approximating the statistically robust sequential patterns. To bound the maximum deviations, as required by GRoSSo, we use Theorem 6 where the upper bound to the capacity is computed using our algorithm (see Section V-C).

The goals of the evaluation are the following: i) assess the performance of our algorithm to compute an upper bound on

the capacity $c(s)$ of a sequence s, comparing our upper bound with the naïve bound and with the one proposed by [19] (see Section V-C); ii) assess the performance of GROSSO on pseudo-artificial datasets, checking whether, with probability $1-\delta$, the set of patterns returned by GROSSO does not contain false positives; iii) assess the performance of GROSSO on real datasets. Since this is the first work that considers the problem of mining SRPs, there are not methods to compare with.

A. Implementation, Environment, and Datasets

We implemented GROSSO for mining statistically robust sequential patterns and our algorithm to compute an upper bound on the capacity of a sequence in Java. To mine the frequent sequential patterns, we used the PrefixSpan [17] implementation provided by the SPMF library [30]. We performed all experiments on the same machine with 512 GB of RAM and 2 Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.3GHz, using Java 1.8.0_201. Our open-source implementation and the code developed for the tests and to generate the datasets are available in [24]. In all experiments, we fixed $\delta = 0.1$.

To obtain sequences of datasets, we generated multiple datasets starting from the Netflix Prize data,[1] which contains over 100 million ratings from 480 thousand randomly-chosen anonymous Netflix customers over 17 thousand movie titles collected between October 1998 and December 2005.

To generate a single dataset, we collected all the movies that have been rated by the users in a given time interval (e.g., in 2004). Each transaction is the temporal ordered sequence of movies rated by a single user, with the movies sorted by ratings' date. Movies rated by such user in the same day form an itemset and each movie is represented by its year of release. Considering consecutive time intervals, we obtained a sequence of datasets, where each dataset only contains data generated in a single time interval. From the original data we removed movies which year of release is not available and movies that have been rated in a year that is antecedent to their year of release. The latter are due to one of the perturbation introduced in the data to preserve the privacy of the users. [2]

We considered the data collected between January 2003 and December 2005. For each year 2004 and 2005, we generated two types of sequences: the first one composed by 4 datasets, e.g., 2004(Q1-Q4) (each dataset contains the data generated in 3 months), and the second one composed by 3 datasets, e.g., 2004(T1-T3) (each dataset contains the data generated in 4 months). Finally, we generated another sequence of datasets, 2003-2005, considering the entire data between 2003 and 2005 (each dataset contains the data generated in one year). The characteristics of the generated datasets are reported in Table I.

B. Upper Bound to the Capacity

In this section, we report the results of our algorithm (see Section V-C), which computes the upper bound $\hat{c}(s)$ on the capacity of a sequence, and compare it with the naïve upper bound $\tilde{c}_n(s) = 2^{||s||} - 1$, and the upper bound $\tilde{c}(s)$ from [19] (see Section V-C). Table I shows the averages (over all transactions) of the relative differences between our novel upper bound $\hat{c}(s)$ and the previously proposed ones, which, for a dataset $\mathcal{D}$, are computed as: $\Delta_{no}(\%) = \frac{1}{|\mathcal{D}|}\sum_{\tau\in\mathcal{D}}\left(\frac{\tilde{c}_n(\tau)-\hat{c}(\tau)}{\tilde{c}_n(\tau)}\right)\cdot 100$ and $\Delta_{po}(\%) = \frac{1}{|\mathcal{D}|}\sum_{\tau\in\mathcal{D}}\left(\frac{\tilde{c}(\tau)-\hat{c}(\tau)}{\tilde{c}(\tau)}\right)\cdot 100$. In all the datasets, our novel bound is (on average) tighter than the other bounds, with a maximum improvement of 13.81% on the naïve method and 13.37% on the method proposed by [19].

TABLE I
DATASETS CHARACTERISTICS AND COMPARISON OF THE UPPER BOUNDS ON THE CAPACITY. $|\mathcal{D}|$: NUMBER OF TRANSACTIONS; $|\mathcal{I}|$: TOTAL NUMBER OF ITEMS; AVG $||\tau||$: AVERAGE TRANSACTION ITEM-LENGTH, $\Delta_{no}(\%)$ AND $\Delta_{po}(\%)$: THE AVERAGE RELATIVE DIFFERENCES BETWEEN OUR UPPER BOUND ON THE CAPACITY AND THE PREVIOUSLY PROPOSED ONES.

Dataset $\mathcal{D}$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	Avg. $\lvert\lvert\tau\rvert\rvert$	$\Delta_{no}(\%)$	$\Delta_{po}(\%)$
2004Q1	132,907	93	24.2	11.42	10.55
2004Q2	165,428	93	23.5	11.61	10.76
2004Q3	184,109	93	24.7	9.18	8.48
2004Q4	218,151	93	24.9	9.77	9.00
2005Q1	266,799	94	26.2	12.31	11.34
2005Q2	291,627	94	25.3	12.15	11.14
2005Q3	315,316	94	24.7	8.67	7.86
2005Q4	295,797	94	19.9	7.89	6.74
2004T1	152,657	93	29.2	11.64	10.94
2004T2	184,202	93	30.3	11.64	10.96
2004T3	229,929	93	30.6	9.71	9.09
2005T1	290,287	94	32.0	13.03	12.17
2005T2	331,117	94	31.4	11.14	10.38
2005T3	326,668	94	25.7	8.53	7.61
2003Y	117,497	92	51.6	13.81	13.37
2004Y	259,407	93	65.9	11.91	11.54
2005Y	451,435	94	62.2	12.07	11.71

C. Results with Pseudo-Artificial Datasets

We performed an experimental evaluation of GROSSO using pseudo-artificial datasets. We considered the 2005(T1-T3) sequence of datasets as *ground truth* for the sequential patterns, and we generated random datasets taking random samples from each of the datasets in the sequence. In such a way, we know the true frequencies of the sequential patterns (the probability that a pattern belongs to a transaction sampled from a dataset is exactly the frequency that such pattern has in that dataset). Then, we executed GROSSO on the pseudo-artificial datasets and, by knowing the true frequencies of the patterns, we can assess its performance in terms of false positives and of correctly reported patterns. Since it is not feasible to obtain all the statistically robust sequential patterns due to the gargantuan number of candidates to consider in such datasets, for the EP and DP scenario, we only considered patterns with true frequency above a minimum threshold θ in the last and first dataset, respectively, while for the SP with true frequency above θ in all the datasets, as defined in Section III.

From each of the three original datasets, 2005T1, 2005T2 and 2005T3, we generated a random dataset with the same size of the corresponding original one, obtaining a sequence of three random datasets. From such sequence, we mined the set of statistically robust sequential patterns without considering

[1] https://www.kaggle.com/netflix-inc/netflix-prize-data
[2] https://en.wikipedia.org/wiki/Netflix_Prize

TABLE II

RESULTS ON PSEUDO-ARTIFICIAL DATASETS FOR EP AND DP. THE TABLES REPORT THE NAME OF THE SEQUENCES OF DATASETS $\mathcal{D}_1^n$, AND THE PARAMETERS ε AND θ. $|GT|$: NUMBER OF SRP FOUND IN THE GROUND TRUTH; T.FP$_f$: PERCENTAGE OF TIMES THAT THE SRP MINED USING OBSERVED FREQUENCIES CONTAIN FALSE POSITIVES; T.FP$_g$: THE PERCENTAGE OF TIMES THAT THE SRP MINED USING GROSSO CONTAIN FALSE POSITIVES; $|\mathcal{A}|/|GT|$: THE AVERAGE RATIO OVER FIVE RANDOM SEQUENCES OF SAMPLES.

Datasets $\mathcal{D}_1^n$	ε	θ	EP				DP			
			$\|GT\|$	T.FP$_f$	T.FP$_g$	$\|\mathcal{A}\|/\|GT\|$	$\|GT\|$	T.FP$_f$	T.FP$_g$	$\|\mathcal{A}\|/\|GT\|$
$\mathcal{S}_1^n$	0.01	0.3	18	0%	0%	0.46	245	60%	0%	0.28
		0.2	104	0%	0%	0.21	2439	100%	0%	0.08
$\mathcal{S}_1^{n\,x2}$	0.01	0.3	18	0%	0%	0.62	245	60%	0%	0.48
		0.2	104	20%	0%	0.38	2439	100%	0%	0.23
$\mathcal{S}_1^{n\,x3}$	0.01	0.3	18	0%	0%	0.67	245	60%	0%	0.58
		0.2	104	60%	0%	0.43	2439	100%	0%	0.34

TABLE III

RESULTS ON PSEUDO-ARTIFICIAL DATASETS FOR SP. THE TABLE SHOWS THE PARAMETERS θ AND α. SEE TABLE II FOR THE MEANING OF THE OTHER VALUES.

Datasets $\mathcal{D}_1^n$	α	θ	$\|GT\|$	T.FP$_f$	T.FP$_g$	$\|\mathcal{A}\|/\|GT\|$
$\mathcal{S}$	0.1	0.3	42	60%	0%	0.02
		0.2	430	100%	0%	0.06
$\mathcal{S}^{x2}$	0.1	0.3	42	40%	0%	0.29
		0.2	430	100%	0%	0.30
$\mathcal{S}^{x3}$	0.1	0.3	42	40%	0%	0.49
		0.2	430	100%	0%	0.46

the uncertain of the data, i.e., directly using Equation 1, Equation 2, or Equation 3, using the observed frequencies of the patterns in the random datasets. This allows us to verify whether the set of sequential patterns obtained considering only the frequencies (i.e., without taking the uncertainty into account) results in false positives.

We then ran GROSSO on the sequence of random datasets to mine a FPF approximation of the statistically robust sequential patterns, and checked whether the returned approximation contained false positives. We also reported what fraction of statistically robust sequential patterns is reported by GROSSO. (For both GROSSO and the observed frequency-based approach above, we only considered patterns with frequency greater than θ as explained above, matching our ground truth.)

Table II shows the average results, over five different random sequences denoted by $\mathcal{S}_1^n$, for mining EP and DP with $\varepsilon \in \{0, 0.01, 0.05\}$, while Table III shows the average results for mining SP with $\alpha \in \{0.05, 0.1\}$. We repeated the entire procedure with five sequences of random datasets, denoted by $\mathcal{S}_1^{n\,2x}$, where each random dataset had size twice the original one, and then with five sequences of random datasets, denoted by $\mathcal{S}_1^{n\,3x}$, with size three times the original one. For all the experiments, we used $\theta \in \{0.2, 0.3\}$. Due to space constraints, only a representative subset of the results is shown in Tables II and III. Other results are analogous.

The results show that, for almost all parameters, the sets of patterns mined in the pseudo-artificial datasets only considering the observed frequency of the patterns (i.e., without considering the uncertainty) contain false positives with high probability, in particular for the DP and SP scenario. In addition, such probability increases with a lower θ, and thus with a large number of patterns. Instead, the patterns returned by GROSSO do not contain false positives in all the runs and with all the parameters. The results are even better than the theoretical guarantees, since theory guarantees us a probability at least $1 - \delta = 0.9$ of obtaining a set without false positives. Let us note that GROSSO does not provide guarantees on the false negatives, i.e., on patterns that are SRPs but that are not returned, and in some case, the percentage of reported SRPs is small, in particular for the SP. However, such percentage increases with larger datasets. For the EP and DP, the results obtained with $\varepsilon = 0$ are very close to the ones reported by Table II, in many cases even better, while with $\varepsilon = 0.05$ GROSSO reported a lower percentage (between 0.003 and 0.23) of statistically robust sequential patterns, in particular for the DP scenario. For the SP instead, using $\alpha = 0.05$, we found only few real SRPs in the original data, and GROSSO did not report any of them.

For the EP and DP scenario, we also performed an other type of experiment to verify the absence of false positives in the output of GROSSO. We generated a random sequence of datasets taking three random samples from the same original dataset 2005T1. In such a way, the random sequence did not contain any EP and DP, since each pattern had the same true frequency in all the datasets. Then, we executed GROSSO on such sequence using $\theta = 0$ and $\varepsilon = 0$. Note that this choice of parameters is the most challenging scenario, since we searched for all the EP and DP we were able to find. Again, we repeated such experiment with five different random sequences where each dataset had the same size of the original one, five sequences with double size and five sequences with datasets that had three times the size of the original one. In all the runs, GROSSO correctly did not report any EP and DP.

These results show that, in general, considering the observed frequencies of the patterns is not enough to find sets of SRPs that do not contain false positives. Thus, techniques like the one introduced in this work are necessary to find large sets of SRPs without false positives. In addition, GROSSO is an effective tool to find SRPs avoiding false positives.

D. Results with Real Datasets

Here, we report the results of GROSSO for mining statistically robust sequential patterns from the Netflix datasets. For the EP and DP, we did not use any constraints on

the minimum frequency, thus we reported every statistically robust sequential patterns found in the data. Table IV shows the results for the EP and DP. In the EP scenario, for the sequences of datasets composed by four datasets (denoted by Q1-Q4), GROSSO reported only few patterns. In particular, all the emerging sequential patterns returned contain the year of the dataset in which they were found, e.g., in 2004(Q1-Q4) all the EP contain the item 2004, with a frequency close to zero in the first dataset. Since during the year many more movies come out, the number of users that rates such movies increases through the year and so such patterns emerge through the sequence. We found the same result in sequences composed by three datasets (denoted by T1-T3) but in this case GROSSO reported many more patterns, in particular for the 2004 sequence, since now we were considering the emerging condition only in three datasets, and thus patterns with such "emerging behavior" are easier to discover.

GROSSO did not report any DP in all the datasets using $\varepsilon = 0.05$. Observing the patterns found on 2005(T1-T3), we noted that the maximum absolute difference $\max_{s \in \mathcal{A}} |f_{\mathcal{D}_1}(s) - f_{\mathcal{D}_n}(s)|$ over all the returned patterns between the frequency of a pattern in the first dataset and its frequency in the last dataset was 0.26, while for the EP such difference was 0.60. Thus, while the frequencies of the EP increase a lot through the year, the frequencies of the DP decrease less, which explains why fewer descending patterns are found by GROSSO. The DP found on 2005(T1-T3) are on average larger than the EP found on the same data, and the 96% of such patterns contain the item 2004, many of them multiple times. Thus, they probably represent long sequential patterns whose frequencies decrease, since the users watch always less 2004's movies through the year 2005 and so, it is difficult for such long patterns to persist through the time.

Table V shows the results for the SP. We performed experiments varying $\theta \in \{0.2, 0.4\}$ and $\alpha \in \{0.05, 0.1\}$. With $\alpha = 0.05$, GROSSO did not report any SP for all the datasets. Almost always the SP found by GROSSO are quite short combinations of items that represents movies of the 90s or early 2000s, that precede the year of the mined sequence. It is surprising that sequential patterns that contain such "old" items are stable through the time, e.g., $\langle\{2000, 2001\}, \{1990\}\rangle$ has a maximum absolute difference between all its frequencies of 0.025 in the sequence 2003-2005(Y).

Overall, the results show that GROSSO detects various types of SRPs from real datasets, obtaining insights into the evolution of the generative process underlying the data.

TABLE IV
RESULTS ON REAL DATASETS FOR EP AND DP. $\mathcal{D}_1^n$: NAME OF THE SEQUENCES OF DATASETS; ε: EMERGING THRESHOLD; $|\mathcal{A}|$: NUMBER OF RETURNED SRP; AVG$||s||$: AVERAGE ITEM-LENGTH OF RETURNED SRP.

Datasets $\mathcal{D}_1^n$	ε	EP		DP	
		$\lvert\mathcal{A}\rvert$	Avg$\lvert\lvert s\rvert\rvert$	$\lvert\mathcal{A}\rvert$	Avg$\lvert\lvert s\rvert\rvert$
2004(Q1-Q4)	0	25	2.4	0	/
	0.01	16	2.3	0	/
	0.05	1	1.0	0	/
2005(Q1-Q4)	0	2	1.5	10	3.2
	0.01	1	1.0	2	2.5
	0.05	0	/	0	/
2004(T1-T3)	0	5213	4.6	5	3.4
	0.01	2214	4.4	0	/
	0.05	207	3.6	0	/
2005(T1-T3)	0	113	3.6	689	5.4
	0.01	48	3.3	187	4.9
	0.05	4	2.5	0	/
2003-2005(Y)	0.05	15107	5.4	14	5.5

TABLE V
RESULTS ON REAL DATASETS FOR SP. THE TABLE REPORTS MINIMUM FREQUENCY THRESHOLD θ AND ERROR THRESHOLD α. SEE TABLE IV FOR THE MEANING OF THE OTHER VALUES.

Datasets $\mathcal{D}_1^n$	α	θ	$\lvert\mathcal{A}\rvert$	Avg$\lvert\lvert s\rvert\rvert$
2004(Q1-Q4)	0.1	0.4	2	1.0
		0.2	40	1.8
2005(Q1-Q4)	0.1	0.4	0	/
		0.2	7	1.7
2004(T1-T3)	0.1	0.4	3	1.0
		0.2	146	2.2
2005(T1-T3)	0.1	0.4	1	1.0
		0.2	18	2.1
2003-2005(Y)	0.1	0.4	3	2.0
		0.2	458	3.9

VII. CONCLUSIONS

In this work, we introduced the problem of mining *statistically robust patterns* from a *sequence of datasets*, which naturally arises in several applications. We provided a general framework for such problem and described GROSSO, an algorithm to identify approximations of the SRPs with probabilistic guarantees on false discoveries, and applied it to identify statistically robust *sequential* patterns. Our extensive experimental evaluation shows that GROSSO significantly improves over the naïve approach which ignores the uncertainty in the data, and that it identifies interesting patterns in real datasets. While in our application we use the VC-dimension to bound the maximum deviation, any uniform convergence bound (e.g., from Rademacher complexity) can be used in our framework. Interesting future directions are the use of improved bounds on the maximum deviation, which may lead to higher statistical power, and to consider a streaming setting for the data.

ACKNOWLEDGMENT

Part of this work was supported by the MIUR, the Italian Ministry of Education, University and Research, under PRIN Project n. 20174LF3T8 AHeAD (Efficient Algorithms for HArnessing Networked Data) and the initiative "Departments of Excellence" (Law 232/2016), and by the Univ. of Padova under project SEED 2020 RATED-X.

REFERENCES

[1] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Min. Knowl. Discov.*, 2007.
[2] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD 1993*.
[3] R. Agrawal and R. Srikant, "Mining sequential patterns," *IEEE ICDM 1995*.
[4] W. Klösgen, "Problems for knowledge discovery in databases and their treatment in the statistics interpreter explora," *Int. J. of Intel. Sys.*, 1992.

[5] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield, "Efficient graphlet counting for large networks," *IEEE ICDM 2015*.

[6] W. Hämäläinen and G. I. Webb, "A tutorial on statistically sound pattern discovery," *Data Min. Knowl. Discov.*, 2019.

[7] L. Pellegrina, M. Riondato, and F. Vandin, "Hypothesis testing and statistically-sound pattern mining," *ACM SIGKDD 2019*.

[8] J. Komiyama, M. Ishihata, H. Arimura, T. Nishibayashi, and S.-i. Minato, "Statistical emerging pattern mining with multiple testing correction," *ACM SIGKDD 2017*.

[9] F. Llinares-López, M. Sugiyama, L. Papaxanthos, and K. Borgwardt, "Fast and memory-efficient significant pattern mining via permutation testing," *ACM SIGKDD 2015*.

[10] L. Pellegrina, M. Riondato, and F. Vandin, "Spumante: Significant pattern mining with unconditional testing," *ACM SIGKDD 2019*.

[11] L. Pellegrina and F. Vandin, "Efficient mining of the most significant patterns with permutation testing," *Data Min. Knowl. Discov.*, 2020.

[12] R. Gwadera and F. Crestani, "Ranking sequential patterns with respect to significance," *PAKDD 2010*.

[13] C. Low-Kam, C. Raïssi, M. Kaytoue, and J. Pei, "Mining statistically significant sequential patterns," *IEEE ICDM 2013*.

[14] A. Tonon and F. Vandin, "Permutation strategies for mining significant sequential patterns," *IEEE ICDM 2019*.

[15] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," *ACM SIGKDD 1999*.

[16] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," *EDBT*, 1996.

[17] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining sequential patterns by pattern-growth: The prefixspan approach," *IEEE Trans. Knowl. Data Eng.*, 2004.

[18] J. Wang, J. Han, and C. Li, "Frequent closed sequence mining without candidate maintenance," *IEEE Trans. Knowl. Data Eng.*, 2007.

[19] S. Servan-Schreiber, M. Riondato, and E. Zgraggen, "Prosecco: Progressive sequence mining with convergence guarantees," *K. Inf. Sys.*, 2019.

[20] D. Santoro, A. Tonon, and F. Vandin, "Mining sequential patterns with VC-dimension and Rademacher complexity," *Algorithms*, 2020.

[21] M. Riondato and F. Vandin, "Finding the true frequent itemsets," *SIAM SDM 2014*.

[22] F. Zhu, X. Yan, J. Han, S. Y. Philip, and H. Cheng, "Mining colossal frequent patterns by core pattern fusion," *IEEE ICDE 2007*.

[23] E. Egho, D. Gay, M. Boullé, N. Voisine, and F. Clérot, "A user parameter-free approach for mining robust sequential classification rules," *Knowl. Inf. Syst.*, 2017.

[24] https://github.com/VandinLab/gRosSo.

[25] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of Complexity*, 2015.

[26] S. Boucheron, O. Bousquet, and G. Lugosi, "Theory of classification: A survey of some recent advances," *ESAIM: prob. and stat.*, 2005.

[27] M. Mitzenmacher and E. Upfal, *Probability and computing: randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press, 2017.

[28] Y. Li, P. M. Long, and A. Srinivasan, "Improved bounds on the sample complexity of learning," *J. Comput. Syst. Sci.*, 2001.

[29] E. Egho, C. Raïssi, T. Calders, N. Jay, and A. Napoli, "On measuring similarity for sequences of itemsets," *Data Min. Knowl. Discov.*, 2015.

[30] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The spmf open-source data mining library version 2," *ECML PKDD 2016*.