Knowledge and Information Systems (2022) 64:2329–2359
https://doi.org/10.1007/s10115-022-01689-2

REGULAR PAPER

GROSSO: mining statistically robust patterns from a sequence of datasets

Andrea Tonon[1] · **Fabio Vandin**[1]

Received: 28 January 2021 / Revised: 4 April 2022 / Accepted: 9 April 2022 /
Published online: 2 August 2022

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 by 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 rigorous approximations of the statistically robust patterns that do not contain false positives or false negatives with high probability. We apply our framework to the mining of statistically robust sequential patterns and statistically robust itemsets. 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 and statistically robust itemsets.

Keywords Statistically robust patterns · Sequential pattern mining · Itemset mining · VC-dimension · Statistically sound pattern mining

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

A preliminary version of this work appeared in the proceedings of IEEE ICDM'20 [1].

✉ Fabio Vandin
fabio.vandin@unipd.it

Andrea Tonon
andrea.tonon@dei.unipd.it

[1] Department of Information Engineering, University of Padova, 35131 Padova, Italy

1 Introduction

Frequent pattern mining [2] 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 [3] to sequential patterns [4], to subgroups [5], to graphlets [6]) 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* [7]. Let us 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. As a result, 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* [8]. In addition, techniques developed for significant pattern mining [7] or for statistically emerging pattern mining [9] 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 or in terms of false negatives.

1.1 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

(see Sect. 2). We show how such a strategy can be used to approximate the three types of statistically robust patterns we introduced.
– We define an approximation of the statistically robust patterns that does not contain *false negatives*, and explain how GRoSSo can be modified to obtain such an approximation with high probability. We also discuss and prove additional guarantees that can be obtained with GRoSSo.
– 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 of sequential patterns, which may be of independent interest.
– We apply the general framework of statistically robust patterns to mine itemsets using the VC-dimension of itemsets to bound the maximum deviation.
– We perform an extensive experimental evaluation, mining statistically robust sequential patterns and itemsets from pseudo-artificial datasets. Our evaluation proves that relying on frequency alone leads to several spurious discoveries, while GRoSSo provides high-quality approximations for both data mining tasks. Finally, we analyze real datasets mining statistically robust sequential patterns, proving that GRoSSo is able to detect various type of patterns.

1.2 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 and itemset mining, which are the applications 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 [7] and the recent works [10–12]. Few works have been proposed to mine statistically significant sequential patterns [13–15]. 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 [16]. 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 [9]. 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 [9] can be modified to work in our scenario.

Frequent itemset mining [3] and frequent sequential pattern mining [4] are two fundamental data mining problems, and several algorithms have been proposed for these tasks (e.g., [17–20]). Servan-Schreiber et al. [21] are the first who apply the statistical learning theory concept of VC-dimension to sequential patterns, and they provide 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, Santoro et al. [22] provide a sampling-based algorithm to compute approximations for the frequent sequential patterns problem, based on an upper bound on the VC-dimension of sequential patterns. They are also the first who 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 a 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 [22], we consider the problem of mining statistically robust patterns in a sequence of datasets, that is a different task. Riondato and Upfal [23] are the first who apply the VC-dimension to itemsets, providing a sampling-based algorithm to compute approximations for the frequent itemsets problem. Riondato and Vandin [24] are instead the first who consider the extraction of frequent patterns w.r.t. an underlying generative process, based on the concept of empirical VC-dimension of itemsets. While in our application to mine statistically robust itemsets we use some of the results provided in these works, we consider the problem of mining itemsets in a sequence of datasets, that is a different problem.

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, Zhu et al. [25] define 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. Egho et al. [26] introduce a space of rules patterns model and they define 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.

This version of our work differs in many ways from the preliminary one that appeared in the proceedings of IEEE ICDM'20 [1]. The major changes are the following:

- We include additional proofs and pseudo-code that had been removed from the previous version due to space constraints.
- We define a *false negatives free* approximation, and we discuss how GRoSSo can be extended to obtain such an approximation (Sect. 4.4).
- We discuss additional guarantees that can be obtained with GRoSSo for both types of approximations (Sect. 4.5).
- We include another application of the general framework of statistically robust patterns, namely mining statistically robust itemsets, using the VC-dimension of itemsets to bound the maximum deviation (Sect. 6).
- We extend our experimental evaluation to include experiments on mining statistically robust itemsets from pseudo-artificial datasets, and on mining false negatives free approximations for both statistically robust sequential patterns and statistically robust itemsets (Sect. 7).

1.3 Organization of the paper

The rest of the paper is structured as follows. Sect. 2 contains the definitions and concepts used throughout this work. Our framework for statistically robust pattern mining is presented in Sect. 3. Section 4 describes our algorithm, GRoSSo to mine statistically robust patterns and provides discussions and proofs of the guarantees that can be obtained with GRoSSo. The

application of our approach for mining statistically robust *sequential* patterns is described in Sect. 5, while the application for mining statistically robust *itemsets* is described in Sect. 6. Section 7 reports the results of an extensive suite of experiments performed to evaluate the effectiveness of GRoSSo on pseudo-artificial and real datasets. Section 8 concludes the paper with some final remarks.

2 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, which is required by our strategy to find an approximation of the statistically robust patterns, and of VC-dimension, showing how it can be used to bound the maximum deviation.

2.1 Pattern mining

Let a *dataset* $\mathcal{D} = \{\tau_1, \tau_2, \ldots, \tau_m\}$ 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, that is, $T_{\mathcal{D}}(p) = \{\tau \in \mathcal{D} : p \sqsubseteq \tau\}$. Finally, the *frequency* $f_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the *fraction* of transactions in $\mathcal{D}$ to which p belongs, that is,

$$f_{\mathcal{D}}(p) = \frac{|T_{\mathcal{D}}(p)|}{|\mathcal{D}|}.$$

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, that is,

$$FP(\mathcal{D}, \theta) = \{(p, f_{\mathcal{D}}(p)) : p \in \mathbb{U}, f_{\mathcal{D}}(p) \geq \theta\}.$$

Given a generic set $\mathcal{A}$ of pairs $(p, \cdot)$, where the first element of each pair is a pattern, in the following, with an abuse of notation, we use $p \in \mathcal{A}$ to indicate that $\exists (p, \cdot) \in \mathcal{A}$, e.g., $p \in FP(\mathcal{D}, \theta) \Rightarrow \exists (p, f_{\mathcal{D}}(p)) \in FP(\mathcal{D}, \theta)$.

2.2 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, that is,

$$t_\pi(p) = \Pr_{\tau \sim \pi}(p \sqsubseteq \tau).$$

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, that is,

$$TFP(\pi, \theta) = \{(p, t_\pi(p)) : p \in \mathbb{U}, t_\pi(p) \geq \theta\}.$$

Let us 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 $TFP(\pi, \theta)$.

2.3 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* $\mathbb{E}[g]$ 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* $E(g, A)$ 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 a measure, i.e., finding a $\mu \in (0, 1)$ such that (s.t.)

$$\Pr\left(\sup_{p \in \mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu\right) \geq 1 - \delta,$$

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 [27] and Rademacher complexity [28]. 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.

2.4 VC-dimension

The Vapnik–Chervonenkis (VC) dimension [27, 29] 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 1 [30] *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 $1 - \delta$.

2.4.1 Range space of patterns

We now define the range space of patterns and prove how the VC-dimension can be used to bound the maximum deviation in the TFP scenario.

Definition 3 Let $\mathbb{U}$ be a domain and let π be a probability distribution on $\mathbb{U}$. We define $RS = (X, \mathcal{R})$ to be a range space associated with $\mathbb{U}$ w.r.t. π such that:

- $X = \mathbb{U}$;
- $\mathcal{R} = \{T(p) : p \in \mathbb{U}\}$ is a family of sets of transactions such that for each pattern p the set $T(p) = \{\tau \in \mathbb{U} : p \sqsubseteq \tau\}$ is the true support set of p.

The following theorem is a generalization of a result for sequential patterns appearing in [22]. Here, we provide it for a general pattern mining task.

Theorem 2 *Let RS be the range space associated with $\mathbb{U}$ w.r.t. π, let $\mathcal{D}$ be a finite bag of i.i.d. sample from π, and let v be the empirical VC-dimension $EVC(RS, \mathcal{D})$ of RS on $\mathcal{D}$. Then, given $\delta \in (0, 1)$ and*

$$\mu = \sqrt{\frac{1}{2|\mathcal{D}|}\left(v + \ln\frac{1}{\delta}\right)},$$

$\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu$ *with probability at least* $1 - \delta$.

Proof From Theorem 1, we know that $\mathcal{D}$ is a μ-bag of $(\mathbb{U}, \pi)$ with probability at least $1 - \delta$. Then, from Definition 2,

$$\left|\Pr_\pi(r) - \frac{|\mathcal{D} \cap r|}{|\mathcal{D}|}\right| \leq \mu$$

for all $r \in \mathcal{R}$. Given a pattern $p \in \mathbb{U}$ and its real support set $T(p)$, which is the range r_p, from the definition of range space of patterns (Definition 3) we have

$$\Pr_\pi(r_p) = t_\pi(p)$$

and

$$\frac{|\mathcal{D} \cap r_p|}{|\mathcal{D}|} = f_\mathcal{D}(p).$$

Thus, $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu$ with probability at least $1 - \delta$. □

In Sects. 5 and 6, we discuss, respectively, an efficient computable upper bound of the empirical VC-dimension of sequential patterns and of itemsets, to bound the maximum deviation of the true frequencies for the two data mining tasks.

3 Statistically robust pattern mining

In this work, we introduce the task of mining *statistically robust patterns* from a *sequence of datasets*. Let us consider the scenario in which we have a sequence $\mathcal{D}_1^n = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$ 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 $i \in \{1, \ldots, n\}$. Let $\Pi_1^n = \{\pi_1, \pi_2, \ldots, \pi_n\}$ denote the sequence of the n probability distributions and $\mathcal{T}_p = \{t_{\pi_1}(p), t_{\pi_2}(p), \ldots, t_{\pi_n}(p)\}$ the sequence 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 $\Pi_1^n = \{\pi_1, \pi_2, \ldots, \pi_n\}$, a condition $cond(\mathcal{T}_p)$ on the true frequencies $\mathcal{T}_p$ that defines the patterns we are interested in, with $cond(\mathcal{T}_p) = 1$

when the condition is satisfied and $cond(\mathcal{T}_p) = 0$ otherwise, *statistically robust pattern (SRP) 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 $cond(\mathcal{T}_p)$, that is,

$$SRP(\Pi_1^n) = \{(p, \mathcal{T}_p) : p \in \mathbb{U} \wedge cond(\mathcal{T}_p) = 1\}.$$

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 $\mathcal{F}_p = \{f_{\mathcal{D}_1}(p), f_{\mathcal{D}_2}(p), \ldots, f_{\mathcal{D}_n}(p)\}$ the sequence of the n frequencies of p in $\mathcal{D}_1^n$, we define a *false positives free (FPF) approximation* $\mathcal{A}_P$ of $SRP(\Pi_1^n)$ as

$$\mathcal{A}_P = \{(p, \mathcal{F}_p) : \exists (p, \mathcal{T}_p) \in SRP(\Pi_1^n)\}.$$

The approximation $\mathcal{A}_P$ does not contain *false positives*, that is, patterns $p \notin SRP(\Pi_1^n)$. In Sect. 4.4, we define an approximation that does not contain false negatives.

Now, 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.

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 $i \in \{1, \ldots, n-1\}$, 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

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

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 $i \in \{1, \ldots, n-1\}$, 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

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

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 $i \neq j \in \{1, \ldots, n\}$, for some given *stability threshold* $\alpha \in (0, 1)$ and a minimum frequency threshold $\theta \in (0, 1)$. Formally, given a stability threshold $\alpha \in (0, 1)$ and minimum frequency threshold $\theta \in (0, 1)$, we define the *stability condition* $cond^S(\mathcal{T}_p)$ as

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

Let us 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.

4 GROSSO: approximating the statistically robust patterns

In this section, we describe GROSSO, mininG statistically ROBuSt patterns from a Sequence Of datasets, our strategy to provide a rigorous approximation of the SRPs. In particular, GROSSO aims to find an approximation that does not contain false positives (i.e., a FPF approximation, see Sect. 3) with high probability. In Sects. 4.1–4.3 we show how to apply such a strategy to mine approximations of the three types of SRPs we defined in the previous section. GROSSO can also be modified to find approximations with guarantees on the false negatives: in Sect. 4.4, we show how to use GROSSO to find such approximations for the EP task. Finally, in Sect. 4.5 we describe additional guarantees that can be obtained with GROSSO for both types of approximations.

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

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

```
1  foreach D_i ∈ D_1^n do
2  |  μ_i ← computeMaxDev(D_i, δ/n);
3  |  θ̃_i ← minimum frequency threshold for D_i computed considering cond_P(F_p, μ_i^n);
4  B ← FP(D_k, θ̃_k), with k = arg max_{i∈{1,...,n}} θ̃_i;
5  A_P ← ∅;
6  foreach (p, f_{D_k}(p)) ∈ B do
7  |  F_p ← empty array of n elements;
8  |  F_p[k] ← f_{D_k}(p); /* F_p[k]: k-th element of F_p */
9  |  A_P ← A_P ∪ (p, F_p);
10 foreach D_i ∈ D_1^n \ D_k do
11 |  foreach (p, F_p) ∈ A_P do
12 |  |  F_p[i] ← computeFrequency(D_i, p);
13 |  |  if cond_P(F_p, μ_1^n) = 0 then
14 |  |  |  A_P ← A_P \ (p, F_p);
15 return A_P;
```

Algorithm 1 shows the pseudo-code of GROSSO. 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 $\delta \in (0, 1)$ as input, we start by 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 $i \in \{1, \ldots, n\}$ (lines 1-2). Each upper bound is computed using δ/n as confidence parameter, thus $\Pr(\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i) \geq 1 - \delta/n$, $\forall\, i \in \{1, \ldots, n\}$. We denote by $\mu_1^n = \{\mu_1, \mu_2, \ldots, \mu_n\}$ the sequence of the n upper bounds on the maximum deviations. (Such upper bounds can be computed, for example, using Theorem 2 and the VC-dimension.) Since $cond(\mathcal{T}_p)$ considers the true frequencies $\mathcal{T}_p$, which are unknown, we need to define a new condition $cond_P(\mathcal{F}_p, \mu_1^n)$ on the frequencies $\mathcal{F}_p$ and on the upper bounds μ_1^n. Such new FPF condition takes into account the uncertainty of the data in our samples, i.e., the datasets, in order to avoid false positives, and, for a pattern p, it must be $cond(\mathcal{T}_p) = 0 \implies cond_P(\mathcal{F}_p, \mu_1^n) = 0$ if $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$ holds $\forall i \in \{1, \ldots, n\}$. Figure 1 shows such conditions for the EP and SP scenarios as examples. Let us note that $cond_P(\mathcal{F}_p, \mu_1^n)$ can be also evaluated only considering a subsequence of the frequencies $\mathcal{F}_p$, and if $cond_P(\mathcal{F}_p, \mu_1^n) = 0$ for some subsequence, then there are no guarantees that $cond(\mathcal{T}_p) = 1$. Then, we aim to find a starting set of possible candidates. For

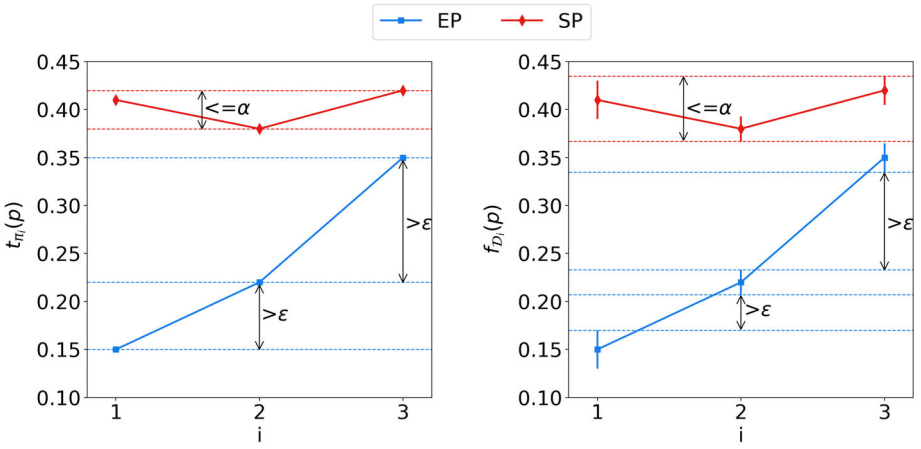

Fig. 1 FPF conditions for the EP and SP. The two panels show the $cond(\mathcal{T}_p)$ (left) and the corresponding $cond_P(\mathcal{F}_p, \mu_1^n)$ (right), which takes into account the uncertainty of the data and avoids false positives, for the EP and SP tasks, with $n = 3$ datasets

each dataset $\mathcal{D}_i$, we compute the minimum frequency threshold $\tilde{\theta}_i$ which the patterns must have in such a dataset to verify $cond_P(\mathcal{F}_p, \mu_1^n)$ (line 3). We then mine the dataset $\mathcal{D}_k$, where $k = \arg\max_{i\in\{1,\ldots,n\}} \tilde{\theta}_i$, 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 (line 4). The idea is to mine the dataset with the highest minimum frequency threshold in order to obtain a set of possible candidates that is as small as possible. Let us note that any efficient algorithm for mining frequent patterns can be used to obtain $FP(\mathcal{D}_k, \tilde{\theta}_k)$. For each pattern $p \in \mathcal{B}$, we then create an empty array $\mathcal{F}_p$ of length n, initialize its k-th element with $f_{\mathcal{D}_k}(p)$, and put the pair $(p, \mathcal{F}_p)$ in the set $\mathcal{A}_P$ containing all possible candidates (lines 6-9). Finally, we explore the remaining datasets (lines 10-14). For each $\mathcal{D}_i \in \mathcal{D}_1^n \setminus \mathcal{D}_k$ and for each pattern $p \in \mathcal{A}_P$, we compute its frequency $f_{\mathcal{D}_i}(p)$ in $\mathcal{D}_i$ and initialize the i-th element of $\mathcal{F}_p$ with $f_{\mathcal{D}_i}(p)$ (line 12), and we check whether $cond_P(\mathcal{F}_p, \mu_1^n) = 1$, considering the subsequence of the frequencies $\mathcal{F}_p$ that has already been computed. If $cond_P(\mathcal{F}_p, \mu_1^n) = 0$, there are no guarantees that $cond(\mathcal{T}_p) = 1$, and we remove such a pattern from the set of the possible candidates (lines 13-14). Then, outputs are the patterns that have not been removed from the set of the possible candidates (line 15).

Theorem 3 *The set $\mathcal{A}_P$ returned by* GROSSO *is a FPF approximation of $SRP(\Pi_1^n)$ with probability $\geq 1 - \delta$.*

Proof From the definition of $cond_P(\mathcal{F}_p, \mu_1^n)$, we know that $cond(\mathcal{T}_p) = 0 \implies cond_P(\mathcal{F}_p, \mu_1^n) = 0$ if $\sup_{p\in\mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$ holds $\forall\, i \in \{1, \ldots, n\}$. In such a scenario, only the patterns $p \in SRP(\Pi_1^n)$ can appear in $\mathcal{A}_P$, and thus $\mathcal{A}_P$ 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 $i \in \{1, \ldots, n\}$. 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 $\Pr(\exists i \in \{1, \ldots, n\} : \sup_{p\in\mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| > \mu_i) = \Pr(\cup_{i=1}^n E_i) \leq \sum_{i=1}^n \Pr(E_i) < \delta$. Thus, the set $\mathcal{A}_P$ returned by GROSSO is a FPF approximation of $SRP(\Pi_1^n)$ with probability $\geq 1 - \delta$. □

4.1 FPF approximation of the EP

Now, we apply the strategy defined above to find a FPF approximation of the EP. Starting from $cond^E(\mathcal{T}_p)$ (Equation 1), we define $cond^E_P(\mathcal{F}_p, \mu_1^n)$ as

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

For a given $i \in \{1, \ldots, n-1\}$, such a 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 a condition is true, we are guaranteed that $t_{\pi_{i+1}}(p) > t_{\pi_i}(p) + \varepsilon$. (See Fig. 1.) Then, starting from such a 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 a reasoning for all the n datasets and considering $f_{\mathcal{D}_1}(p) \geq 0$, we obtain the minimum frequency threshold $\tilde{\theta}^E_n$ for the dataset $\mathcal{D}_n$,

$$\tilde{\theta}^E_n = (n-1) \cdot \varepsilon + \mu_1 + \mu_n + \sum_{i=2}^{n-1} 2 \cdot \mu_i,$$

the highest over all the n datasets. Thus, the set $FP(\mathcal{D}_n, \tilde{\theta}^E_n)$ 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 $cond^E_P(\mathcal{F}_p, \mu_1^n)$.

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

Proof Let us consider that $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i, \forall\, i \in \{1, \ldots, n\}$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i,\, f_{\mathcal{D}_i}(p) + \mu_i], \forall\, i \in \{1, \ldots, n\}$. 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 $j \in \{1, \ldots, n-1\}$, 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 $cond^E_P(\mathcal{F}_{p'}, \mu_1^n) = 0$. □

If one is interested in patterns with a true frequency above a value $\theta \in (0, 1)$, i.e., $t_{\pi_i}(p) \geq \theta, \forall i \in \{1, \ldots, n\}$, 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 $\tilde{\theta}^E_n$. Instead, if one is interested in patterns p with $t_{\pi_n}(p) \geq \theta$, the minimum frequency threshold $\tilde{\theta}^E_n$ for dataset $\mathcal{D}_n$ is

$$\tilde{\theta}^E_n = \max\{(n-1) \cdot \varepsilon + \mu_1 + \mu_n + \sum_{i=2}^{n-1} 2 \cdot \mu_i,\ \theta + \mu_n\}.$$

4.2 FPF approximation of 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 $cond^D_P(\mathcal{F}_p, \mu_1^n)$ as

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

Iterating such a 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 $cond^D_P(\mathcal{F}_p, \mu_1^n)$. In the case of a minimum frequency threshold $\theta \in (0, 1)$, reasoning analogous to the EP can be applied.

Theorem 5 $cond^D(\mathcal{T}_p) = 0 \implies cond^D_P(\mathcal{F}_p, \mu_1^n) = 0.$ □

The proof is analogous to the proof of Theorem 4.

4.3 FPF approximation of 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 $cond^S_P(\mathcal{F}_p, \mu_1^n)$ as

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

Given $i \neq j \in \{1, \ldots, n\}$, 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 a condition is true, we can prove that $t_{\pi_i}(p) \geq \theta$. (See Fig. 1.) The only condition that affects the minimum frequency thresholds $\tilde{\theta}_i^S$ is $f_{\mathcal{D}_i}(p) \geq \theta + \mu_i, \forall\, i \in \{1, \ldots, n\}$. So, we have $\tilde{\theta}_i^S = \theta + \mu_i,\ \forall\, i \in \{1, \ldots, n\}$, and the set $FP(\mathcal{D}_k, \tilde{\theta}_k^S)$, with $k = \arg\max_{i \in \{1,\ldots,n\}} \tilde{\theta}_i^S$, provides the starting candidates. Finally, we analyze the remaining datasets $\mathcal{D}_i \in \mathcal{D}_1^n \setminus \mathcal{D}_k$ and check whether the candidates verify $cond^S_P(\mathcal{F}_p, \mu_1^n)$.

Theorem 6 $cond^S(\mathcal{T}_p) = 0 \implies cond^S_P(\mathcal{F}_p, \mu_1^n) = 0.$

Proof Let us consider that $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i, \forall\, i \in \{1, \ldots, n\}$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i, f_{\mathcal{D}_i}(p) + \mu_i], \forall\, i \in \{1, \ldots, n\}$. 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 $i \in \{1, \ldots, n\}$ s.t. $t_{\pi_i}(p') < \theta$ and/or there is at least a couple of distributions π_k, π_j, with $k \neq j \in \{1, \ldots, n\}$, 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 $i \in \{1, \ldots, n\}$, 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 $cond_P^S(\mathcal{F}_{p'}, \mu_1^n) = 0$. Now, let us consider the case in which there is a couple of distributions π_k, π_j, with $k \neq j \in \{1, \ldots, n\}$ 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 $cond_P^S(\mathcal{F}_{p'}, \mu_1^n) = 0$. □

4.4 Guarantees on false negatives

In this section, we explain how GRoSSo can be modified to obtain an approximation without false negatives with high probability. Analogously to what done in Sect. 3, we first define a *false negatives free (FNF) approximation* $\mathcal{A}_N$ of $SRP(\Pi_1^n)$ as

$$\mathcal{A}_N = \{(p, \mathcal{F}_p), \forall (p, \mathcal{T}_p) \in SRP(\Pi_1^n)\}.$$

The approximation $\mathcal{A}_N$ does not contain *false negatives*, that is, it contains all patterns $p \in SRP(\Pi_1^n)$.

Our algorithm GRoSSo can be used to obtain FNF approximations. The procedure is the same described above (see Algorithm 1) but we need to define a new condition $cond_N(\mathcal{F}_p, \mu_1^n)$, a FNF condition that takes into account the uncertainty in the data in order to avoid false negatives, and for a pattern p it must be $cond(\mathcal{T}_p) = 1 \implies cond_N(\mathcal{F}_p, \mu_1^n) = 1$ if $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$ holds $\forall i \in \{1, \ldots, n\}$. Figure 2 shows such a condition for the EP scenario as an example. Then, we compute the minimum frequency threshold $\hat{\theta}_i$ for each dataset $\mathcal{D}_i$, with $i \in \{1, \ldots, n\}$, considering $cond_N(\mathcal{F}_p, \mu_i^n)$, and we mine the set of the starting candidates from the dataset with the highest minimum frequency threshold.

Theorem 7 *The set $\mathcal{A}_N$ returned by* GRoSSo *using $cond_N(\mathcal{F}_p, \mu_1^n)$ is a FNF approximation of $SRP(\Pi_1^n)$ with probability $\geq 1 - \delta$.*

Proof From the definition of $cond_N(\mathcal{F}_p, \mu_1^n)$, we know that $cond(\mathcal{T}_p) = 1 \implies cond_N(\mathcal{F}_p, \mu_1^n) = 1$ if $\sup_{p \in \mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i$ holds $\forall\, i \in \{1, \ldots, n\}$. In such a scenario, all the patterns $p \in SRP(\Pi_1^n)$ appear in $\mathcal{A}_N$, and thus $\mathcal{A}_N$ is a FNF approximation of $SRP(\Pi_1^n)$. The remaining of the proof is analogous of the proof of Theorem 3. □

4.4.1 FNF approximation of the EP

Now, we apply the strategy defined above to find a FNF approximation of the EP. With a similar reasoning, it is possible to mine a FNF approximation of the descending and stable patterns. Starting from $cond^E(\mathcal{T}_p)$ (Equation 1), we define $cond_N^E(\mathcal{F}_p, \mu_1^n)$ as

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

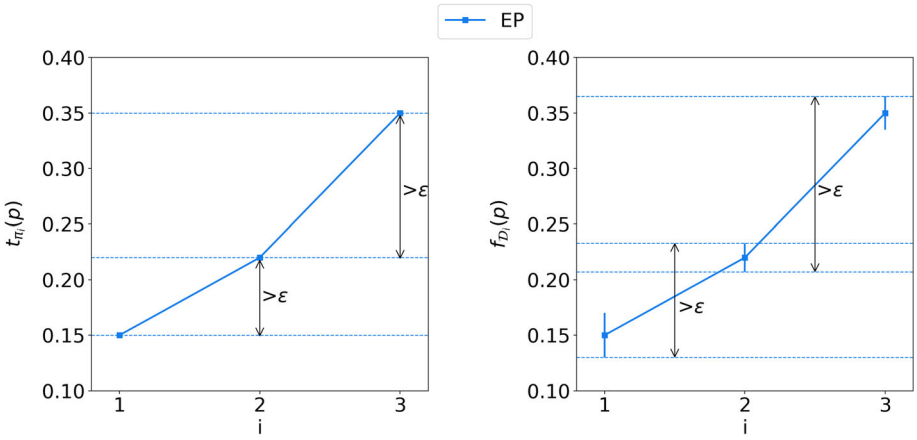

Fig. 2 FNF condition for the EP. The two panels show the $cond(\mathcal{T}_p)$ (left) and the corresponding $cond_N(\mathcal{F}_p, \mu_1^n)$ (right), which takes into account the uncertainty of the data and avoids false negatives, for the EP task, with $n = 3$ datasets

For a given $i \in \{1, \ldots, n-1\}$, such a 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 maximum over all possible values that they can assume. Only if such a condition is false, we can prove that $t_{\pi_{i+1}}(p) \leq t_{\pi_i}(p) + \varepsilon$. (See Fig. 2.) Then, starting from such a 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 a reasoning for all the n datasets and considering $f_{\mathcal{D}_1}(p) \geq 0$, we obtain the minimum frequency threshold

$$\hat{\theta}_i^E = (i-1) \cdot \varepsilon - \mu_1 - \mu_i - \sum_{j=2}^{i-1} 2 \cdot \mu_j,$$

for each dataset $\mathcal{D}_i$, $i \in \{2, \ldots, n\}$, and $\hat{\theta}_1^E = 0$. Thus, the set $FP(\mathcal{D}_k, \hat{\theta}_k^E)$, with $k = \arg\max_{i \in \{1,\ldots,n\}} \hat{\theta}_i^E$, provides the starting candidates. Then, we analyze the remaining datasets $\mathcal{D}_i \in \mathcal{D}_1^n \setminus \mathcal{D}_k$ and check whether the candidates verify $cond_N^E(\mathcal{F}_p, \mu_1^n)$. Let us note that, depending on the values of ε and μ_1^n, the highest minimum frequency threshold $\hat{\theta}_k^E$ can be equal or very close to 0, resulting in a huge amount of starting candidates, sometimes infeasible to mine. Thus, to obtain FNF approximations, a reasonable solution is to only consider patterns with a minimum true frequency. If one is interested in patterns with a true frequency above a value $\theta \in (0, 1)$, i.e., $t_{\pi_i}(p) \geq \theta, \forall\, i \in \{1, \ldots, n\}$, the following strategy can be used to reduce the set of starting candidates. Since we require that $f_{\mathcal{D}_i}(p) \geq \theta - \mu_i$ for all $i \in \{1, \ldots, n\}$ to discard possible false negatives, we obtain the minimum frequency threshold

$$\hat{\theta}_i^E = \max\{(i-1) \cdot \varepsilon + \theta - \mu_i - \sum_{j=1}^{i-1} 2 \cdot \mu_j, \theta - \mu_i\},$$

for each dataset $\mathcal{D}_i, i \in \{1, \ldots, n\}$. Instead, if one is interested in patterns p with $t_{\pi_n}(p) \geq \theta$, the highest minimum frequency threshold is the maximum between $\theta - \mu_n$, for dataset $\mathcal{D}_n$, and the minimum frequency threshold described above without frequency constraints.

Theorem 8 $cond^E(\mathcal{T}_p) = 1 \implies cond^E_N(\mathcal{F}_p, \mu_1^n) = 1.$

Proof Let us consider that $\sup_{p\in\mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i, \forall\, i \in \{1, \ldots, n\}$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i, f_{\mathcal{D}_i}(p) + \mu_i], \forall\, i \in \{1, \ldots, n\}$. Let p' be a pattern s.t. $cond^E(\mathcal{T}_{p'}) = 1$. From Equation 1, we have $t_{\pi_{j+1}}(p') > t_{\pi_j}(p') + \varepsilon$ for all couples of consecutive distributions π_j, π_{j+1}, with $j \in \{1, \ldots, n-1\}$. 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 $f_{\mathcal{D}_{j+1}}(p') + \mu_{j+1} - (f_{\mathcal{D}_j}(p') - \mu_j) > \varepsilon$ is verified for such p' for all $j \in \{1, \ldots, n-1\}$, and thus $cond^E_N(\mathcal{F}_{p'}, \mu_1^n) = 1$. □

4.5 Additional guarantees of GROSSO

In this section, we provide additional guarantees of GROSSO for both types of approximations. In particular, it is possible to derive guarantees on the false negatives that can appear in a FPF approximation returned by GROSSO, and, vice versa, guarantees on the false positives that can appear in a FNF approximation. Such guarantees differ considering different types of patterns (i.e., emerging, descending, and stable), and thus, they must be separately derived for each type of pattern. Here, we prove additional guarantees for the emerging patterns but, with a similar reasoning, it is possible to obtain analogous guarantees for the descending and stable patterns.

Theorem 9 *For any pattern p with $t_{\pi_{i+1}}(p) > t_{\pi_i}(p) + \varepsilon + 2\cdot\mu_i + 2\cdot\mu_{i+1}\ \forall i \in \{1, \ldots, n-1\}$,* $cond^E_P(\mathcal{F}_p, \mu_1^n) = 1$.

Proof Let us consider that $\sup_{p\in\mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i, \forall\, i \in \{1, \ldots, n\}$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i, f_{\mathcal{D}_i}(p) + \mu_i], \forall\, i \in \{1, \ldots, n\}$. Let p' be a pattern s.t. $t_{\pi_{i+1}}(p') > t_{\pi_i}(p') + \varepsilon + 2\cdot\mu_i + 2\cdot\mu_{i+1}\ \forall i \in \{1, \ldots, n-1\}$. Since we know that $t_{\pi_{i+1}}(p') \in [f_{\mathcal{D}_{i+1}}(p') - \mu_{i+1}, f_{\mathcal{D}_{i+1}}(p') + \mu_{i+1}]$ and that $t_{\pi_i}(p') \in [f_{\mathcal{D}_i}(p') - \mu_i, f_{\mathcal{D}_i}(p') + \mu_i]\ \forall i \in \{1, \ldots, n-1\}$, then we have that $f_{\mathcal{D}_{i+1}}(p') - \mu_{i+1} > f_{\mathcal{D}_i}(p') + \mu_i + \varepsilon\ \forall i \in \{1, \ldots, n-1\}$, and thus $cond^E_P(\mathcal{F}_{p'}, \mu_1^n) = 1$. □

Theorem 10 *For any pattern p with $t_{\pi_{i+1}}(p) \leq t_{\pi_i}(p) + \varepsilon - 2\cdot\mu_i - 2\cdot\mu_{i+1}\ \forall i \in \{1, \ldots, n-1\}$,* $cond^E_N(\mathcal{F}_p, \mu_1^n) = 0$.

Proof Let us consider that $\sup_{p\in\mathbb{U}} |t_{\pi_i}(p) - f_{\mathcal{D}_i}(p)| \leq \mu_i, \forall\, i \in \{1, \ldots, n\}$. Thus, we have that for all patterns $p \in \mathbb{U}$, it results $t_{\pi_i}(p) \in [f_{\mathcal{D}_i}(p) - \mu_i, f_{\mathcal{D}_i}(p) + \mu_i], \forall\, i \in \{1, \ldots, n\}$. Let p' be a pattern s.t. $t_{\pi_{i+1}}(p') \leq t_{\pi_i}(p') + \varepsilon - 2\cdot\mu_i - 2\cdot\mu_{i+1}\ \forall i \in \{1, \ldots, n-1\}$. Since we know that $t_{\pi_{i+1}}(p') \in [f_{\mathcal{D}_{i+1}}(p') - \mu_{i+1}, f_{\mathcal{D}_{i+1}}(p') + \mu_{i+1}]$ and that $t_{\pi_i}(p') \in [f_{\mathcal{D}_i}(p') - \mu_i, f_{\mathcal{D}_i}(p') + \mu_i]\ \forall i \in \{1, \ldots, n-1\}$, then we have that $f_{\mathcal{D}_{i+1}}(p') + \mu_{i+1} \leq f_{\mathcal{D}_i}(p') - \mu_i + \varepsilon$ and thus $cond^E_N(\mathcal{F}_{p'}, \mu_1^n) = 0$. □

Theorem 9 and Theorem 10 provide additional guarantees for the emerging patterns returned by GROSSO. In particular, Theorem 9 provides additional guarantees for a FPF approximation returned by GROSSO, stating that a pattern p with $t_{\pi_{i+1}}(p) > t_{\pi_i}(p) + \varepsilon + 2 \cdot \mu_i + 2 \cdot \mu_{i+1}\ \forall i \in \{1, \ldots, n-1\}$ is certainly included in a FPF approximation. Instead, Theorem 10 provides additional guarantees for a FNF approximation returned by GROSSO, stating that a pattern p with $t_{\pi_{i+1}}(p) \leq t_{\pi_i}(p) + \varepsilon - 2 \cdot \mu_i - 2 \cdot \mu_{i+1}\ \forall i \in \{1, \ldots, n-1\}$ cannot appear in a FNF approximation.

5 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 Sect. 2.1. Then, we introduce a novel algorithm to compute an upper bound on the capacity of a sequence and we use such an algorithm to compute an upper bound on the empirical VC-dimension of sequential patterns. Finally, we discuss 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.

5.1 Sequential pattern mining

Let $\mathcal{I} = \{i_1, i_2, \ldots, i_p\}$ 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*) $s = \langle S_1, S_2, \ldots, S_k \rangle$ is a finite ordered sequence of itemsets, with $S_i \subseteq \mathcal{I}$, $S_i \neq \emptyset$ for all $i \in \{1, \ldots, k\}$. We say that such a 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 $y = \langle Y_1, Y_2, \ldots, Y_a \rangle$ is a *subsequence* of an other sequential pattern $w = \langle W_1, W_2, \ldots, W_b \rangle$, denoted by $y \sqsubseteq w$, if and only if there exists a sequence of naturals $1 \leq i_1 < i_2 < \cdots < i_a \leq b$ s.t. $Y_1 \subseteq W_{i_1}, Y_2 \subseteq W_{i_2}, \ldots, Y_a \subseteq W_{i_a}$. Let us 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, that is, $c(s) = |\{a : a \sqsubseteq s\}|$.

Example 1 Let us consider the following sequential dataset $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ as an example:

$$\begin{aligned}
\tau_1 &= \langle \{2, 6, 7\}, \{2\} \rangle \\
\tau_2 &= \langle \{1\}, \{2\}, \{6, 7\}, \{2\} \rangle \\
\tau_3 &= \langle \{1, 4\}, \{3\}, \{2\}, \{1, 2, 5, 6\} \rangle \\
\tau_4 &= \langle \{7\}, \{2\}, \{6, 7\}, \{2\} \rangle.
\end{aligned}$$

The dataset above has 4 transactions. The first one, $\tau_1 = \langle \{2, 6, 7\}, \{2\} \rangle$ has length $|\tau_1| = 2$, item-length $||\tau_1|| = 4$ and capacity $c(\tau_1) = 14$. The frequency $f_{\mathcal{D}}(\langle \{7\}, \{2\} \rangle)$ of the sequence $\langle \{7\}, \{2\} \rangle$ in $\mathcal{D}$, is 3/4, since it is contained in all transactions but τ_3. Let us note that the sequence $\langle \{7\}, \{2\} \rangle$ occurs three times as a subsequence of τ_4, but τ_4 contributes only once to the frequency of $\langle \{7\}, \{2\} \rangle$.

5.2 VC-dimension of sequential patterns

Given a dataset $\mathcal{D}$ for the sequential pattern mining task, that is a finite bag of transactions sampled from $\mathbb{S}$ in according to π, we aim to compute the empirical VC-dimension $EVC(RS, \mathcal{D})$ of the range space (see Definition 3) 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 $EVC(RS, \mathcal{D})$ and using Theorem 2, it is possible to compute a $\mu \in (0, 1)$ s.t. $\sup_{s \in \mathbb{S}} |t_\pi(s) - f_{\mathcal{D}}(s)| \leq \mu$.

The exact computation of the empirical VC-dimension $EVC(RS, \mathcal{D})$ of sequential patterns on the dataset $\mathcal{D}$, required by Theorem 2, is computationally expensive. The *s-index*

introduced by Servan-Schreiber et al. [21] provides an efficiently computable upper bound on $EVC(RS, \mathcal{D})$.

Definition 4 [21] Let $\mathcal{D}$ be a sequential dataset. The s-index of $\mathcal{D}$ is the maximum integer d such that $\mathcal{D}$ contains at least d different sequential transactions with capacity at least $2^d - 1$, such that no one of them is a subsequence of another, i.e., the d transactions form an anti-chain.

5.3 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 [31], 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 [21]. 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, [21] 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 $2^a \cdot (2^{|T|} - 1) \cdot 2^b$. Let us 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 given in Algorithm 2. Let $s = \langle S_1, S_2, \ldots, S_{|S|} \rangle$ 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 $\hat{s} = \langle S_1, S_2, \ldots, S_{|\hat{s}|} \rangle$ be the sequence in the new order, s.t. $|S_i| \leq |S_{i+1}|, \forall\, i \in \{1, \ldots, |\hat{s}| - 1\}$. Let $N = [n_1, n_2, \ldots, n_{|\hat{s}|}]$ 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 $2^{||\hat{s}||} - 1$ (line 2). 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 (line 6). For such S_j, we compute the number of over-counted subsequences with the formula above (line 7). In Algorithm 2 (line 7), the *min* and *max* functions are used to check which itemset comes first in the original ordered sequence. 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 (line 9). 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 (lines 11-13).

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 $\hat{c}(\hat{s})$ on $c(s)$.

$t \leftarrow ||\hat{s}||$;
$\hat{c}(\hat{s}) \leftarrow 2^t - 1$;
for $i \leftarrow 1$ **to** $|\hat{s}| - 1$ **do**
  $val \leftarrow 0$;
  for $j \leftarrow i + 1$ **to** $|\hat{s}|$ **do**
    if $\exists\, T = S_i \cap S_j \ : \ T \neq \emptyset$ **then**
      $val \leftarrow \max\{val, 2^{\min(n_i, n_j)} \cdot \left(2^{|T|} - 1\right) \cdot 2^{t - \max(n_i + |S_i|, n_j + |S_j|)}\}$;
  if $val \neq 0$ **then**
    $\hat{c}(\hat{s}) \leftarrow \hat{c}(\hat{s}) - val$;
    $t \leftarrow t - |S_i|$;
    for $m \leftarrow i + 1$ **to** $|\hat{s}|$ **do**
      if $n_m > n_i$ **then**
        $n_m \leftarrow n_m - |S_i|$;
return $\hat{c}(\hat{s})$;

Example 2 Let us consider the 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 [21] 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., the sequence $\langle \{2, 5\}, \{1, 8\} \rangle$).

5.4 Bound on the maximum deviation

Using Algorithm 2 one can compute upper bounds on the capacities of the transactions of $\mathcal{D}$, which can be used to obtain an upper bound on the s-index. Such bound can be used in Theorem 2 as upper bound of the empirical VC-dimension of sequential patterns, in order to compute a bound on the maximum deviation of the true frequencies of sequential patterns. With such a bound on the maximum deviation, we can use GROSSO to find FPF and FNF approximations of the statistically robust sequential patterns.

6 Application: mining statistically robust itemsets

In this section, we introduce the task of itemset mining, as another concrete realization of the general framework of pattern mining we introduced in Sect. 2.1. Then, we apply the VC-dimension to itemsets and we discuss a VC-dimension-based strategy to bound the maximum deviation of the true frequencies of itemsets, which can be used in the SRP mining scenario.

6.1 Itemset mining

Let $\mathcal{I} = \{i_1, i_2, \ldots, i_p\}$ be a finite set of items. Let us remember that an *itemset* X is a non-empty subset of $\mathcal{I}$, i.e., $X \subseteq \mathcal{I}$, $X \neq \emptyset$. We denote by $\mathbb{I}$ the set of all possible itemsets composed by items from $\mathcal{I}$. The *length* $|X|$ of X is the number of items in X and an itemset X is contained in another itemset Y if and only if $X \subseteq Y$.

Example 3 Let us consider the following dataset $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ as an example:

$$\tau_1 = \{2, 6, 7\}$$
$$\tau_2 = \{1, 2, 6, 7\}$$
$$\tau_3 = \{1, 2, 3, 4, 5, 6\}$$
$$\tau_4 = \{2, 6, 7\}.$$

The dataset above has 4 transactions. The first one, $\tau_1 = \{2, 6, 7\}$, has length $|\tau_1| = 3$. The frequency $f_{\mathcal{D}}(\{6, 7\})$ of the itemset $\{6, 7\}$ in $\mathcal{D}$ is 3/4, since it is contained in all transactions but τ_3.

6.2 VC-dimension of itemsets and bound on the maximum deviation

As described in Sect. 5.2 for the sequential patterns, given a dataset $\mathcal{D}$ for the itemset mining task, that is a finite bag of transactions sampled from $\mathbb{I}$ in according to π, we aim to compute the empirical VC-dimension $EVC(RS, \mathcal{D})$ of the range space (see Definition 3) associated with $\mathbb{I}$ w.r.t. π on the dataset $\mathcal{D}$ in order to find a probabilistic bound $\mu \in (0, 1)$ on the maximum deviation $\sup_{X \in \mathbb{I}} |t_\pi(X) - f_{\mathcal{D}}(X)|$. In particular, given $EVC(RS, \mathcal{D})$ and using Theorem 2, it is possible to compute a $\mu \in (0, 1)$ s.t. $\sup_{X \in \mathbb{I}} |t_\pi(X) - f_{\mathcal{D}}(X)| \leq \mu$. The *d-index* introduced by Riondato and Upfal [23] provides an efficiently computable upper bound on $EVC(RS, \mathcal{D})$.

Definition 5 [23] Let $\mathcal{D}$ be a dataset for the itemset mining task. The d-index of $\mathcal{D}$ is the maximum integer d such that $\mathcal{D}$ contains at least d different transactions of length at least d, such that no one of them is a subset of another, i.e., the d transactions form an anti-chain.

The d-index can be used in Theorem 2 as upper bound of the empirical VC-dimension of itemsets in order to compute a bound on the maximum deviation of the true frequencies of itemsets. With such a bound on the maximum deviation, we can use GRoSSo to find FPF and FNF approximations of the statistically robust itemsets.

7 Experimental evaluation

In this section, we report the results of our experimental evaluation on multiple pseudo-artificial datasets to assess the performance of GRoSSo for approximating the statistically robust sequential patterns and itemsets. Then, we execute GRoSSo on multiple real datasets to approximate the statistically robust sequential patterns and we analyze the sequential patterns mined. To bound the maximum deviations, as required by GRoSSo, we use Theorem 2. The VC-dimension of sequential patterns is bounded using the s-index obtained by using our algorithm (Algorithm 2) to compute the upper bound on the capacity of each sequential transaction, while the VC-dimension of itemsets is bounded using the d-index (Definition 5).

The goals of the evaluation are the following:

- 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 [21] (see Sect. 5.3).
- Assess the performance of GRosSo on pseudo-artificial datasets to mine statistically robust sequential patterns and itemsets, checking whether, with probability $1 - \delta$, the set of patterns returned by GRosSo does not contain false positives or false negatives.
- Assess the performance of GRosSo to mine statistically robust sequential patterns on real datasets.

Since this is the first work that considers the problem of mining SRPs, there are not methods to compare with.

7.1 Implementation, environment, and real datasets

We implemented GRosSo for mining statistically robust sequential patterns and itemsets, and our algorithm to compute an upper bound on the capacity of a sequence in Java. To mine the frequent sequential patterns and frequent itemsets, we used, respectively, the PrefixSpan [20] and the FP-Growth [18] implementations both provided by the SPMF library [32]. 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 at https://github.com/VandinLab/gRosSo. In all experiments, we fixed $\delta = 0.1$.

Here, we provide the details on the generation of the real datasets for the sequential pattern mining task. The details on the generation of pseudo-artificial datasets for the sequential pattern and itemset mining tasks are, respectively, in Sects. 7.3.1 and 7.3.2. To obtain sequences of real sequential 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 a 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 perturbations 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 real datasets are reported in Table 1.

[1] https://www.kaggle.com/netflix-inc/netflix-prize-data.

[2] https://en.wikipedia.org/wiki/Netflix_Prize.

Table 1 Real datasets characteristics and comparison of the upper bounds on the capacity

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

The table reports: Dataset $\mathcal{D}$: name of the real dataset; $|\mathcal{D}|$: number of transactions; $|\mathcal{I}|$: total number of items; Avg $||\tau||$: average transaction item-length; $\Delta_{no}(\%)$ and $\Delta_{po}(\%)$: average relative differences between our upper bound on the capacity and the previously proposed ones. The datasets are grouped in sequences

7.2 Upper bound on the capacity

In this section, we report the results of Algorithm 2, 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 [21]. (See Sect. 5.3.) Table 1 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 [21].

7.3 Results with pseudo-artificial datasets

In this section, we report the results of our evaluation on pseudo-artificial datasets. First, we describe the experimental evaluation using pseudo-artificial datasets for the sequential pattern mining task (Sect. 7.3.1), and then for the itemset mining task (Sect. 7.3.2).

7.3.1 Sequential patterns

Here, we report the results of our experimental evaluation using pseudo-artificial datasets for the sequential pattern mining task. 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 a pattern has in that dataset). Then, we executed GROSSO on the pseudo-artificial datasets and, by knowing the true frequencies of the patterns, we assessed its performance in terms of false positives, false negatives, 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 scenarios, 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 Sect. 3.

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 a sequence, we mined the set of statistically robust sequential patterns without considering 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 or in false negatives.

We then ran GROSSO on the sequence of random datasets to mine a FPF or a FNF approximation of the statistically robust sequential patterns, and checked whether the returned approximation contained, respectively, false positives or false negatives. 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 2 reports the average results, over 5 different random sequences, denoted by $\mathcal{S}_1^n$, for mining FPF approximations of the EP and DP with $\varepsilon \in \{0, 0.01, 0.05\}$, Table 3 reports the average results for mining FPF approximations of the SP with $\alpha \in \{0.05, 0.1\}$, while Table 4 reports the average results for mining FNF approximations of the EP. We repeated the entire procedure with 5 sequences of random datasets, denoted by $\mathcal{S}_1^{n\times 2}$, where each random dataset had size twice the original one, and then with five sequences of random datasets, denoted by $\mathcal{S}_1^{n\times 3}$, with size three times the original one. For all the experiments, we used $\theta \in \{0.2, 0.3\}$. Here, we report only a representative subset of the results, with $\varepsilon = 0.01$ and $\alpha = 0.1$. Other results are analogous and discussed below.

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 or false negatives with high probability. In addition, such a 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 or false

Table 2 Results on pseudo-artificial datasets for EP and DP for the sequential pattern mining task with guarantees on the false positives

Datasets $\mathcal{D}_1^n$	ε	θ	EP				DP			
			$\lvert GT\rvert$	T.FP$_f$	T.FP$_g$	$\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$	$\lvert GT\rvert$	T.FP$_f$	T.FP$_g$	$\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$
$\mathcal{S}_1^n$	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
$\mathcal{S}_1^{n\times 2}$	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
$\mathcal{S}_1^{n\times 3}$	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

The table reports: $\mathcal{D}_1^n$: name of the sequences of datasets; ε: emerging threshold; θ: minimum frequency threshold; for both the EP and DP: $\lvert GT\rvert$: number of SRPs in the ground truth; T.FP$_f$: percentage of times that the SRPs mined using the observed frequencies contain false positives; T.FP$_g$: percentage of times that the SRPs mined using GRoSSo contain false positives; $\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$: average ratio between reported patterns by GRoSSo and patterns in the ground truth over 5 random sequences

Table 3 Results on pseudo-artificial datasets for SP for the sequential pattern mining task with guarantees on the false positives

Datasets $\mathcal{D}_1^n$	α	θ	$\lvert GT\rvert$	T.FP$_f$	T.FP$_g$	$\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$
$\mathcal{S}_1^n$	0.1	0.3	42	60%	0%	0.02
		0.2	430	100%	0%	0.06
$\mathcal{S}_1^{n\times 2}$	0.1	0.3	42	40%	0%	0.29
		0.2	430	100%	0%	0.30
$\mathcal{S}_1^{n\times 3}$	0.1	0.3	42	40%	0%	0.49
		0.2	430	100%	0%	0.46

The table reports: α: stability threshold. See Table 2 for the meaning of the other values

Table 4 Results on pseudo-artificial datasets for EP for the sequential pattern mining task with guarantees on the false negatives

Datasets $\mathcal{D}_1^n$	ε	θ	$\lvert GT\rvert$	T.FN$_f$	T.FN$_g$	$\lvert GT\rvert/\lvert\mathcal{A}_N\rvert$
$\mathcal{S}_1^n$	0.01	0.3	18	60%	0%	0.45
		0.2	104	80%	0%	0.09
$\mathcal{S}_1^{n\times 2}$	0.01	0.3	18	0%	0%	0.68
		0.2	104	20%	0%	0.35
$\mathcal{S}_1^{n\times 3}$	0.01	0.3	18	0%	0%	0.75
		0.2	104	40%	0%	0.50

The table reports: T.FN$_f$: percentage of times that the SRPs mined using the observed frequencies contain false negatives; T.FN$_g$: percentage of times that the SRPs mined using GRoSSo contain false negatives; $\lvert GT\rvert/\lvert\mathcal{A}_N\rvert$: average ratio of patterns in the ground truth and reported patterns by GRoSSo over 5 random sequences. See Table 2 for the meaning of the other values

negatives 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 or without false negatives. Let us note that in some cases, the percentage of reported SRPs is small, in particular for the SP. However, such a percentage increases with larger datasets, since techniques from statistical learning theory, such as the VC-dimension, perform better when larger collections of data are available. In the EP scenario, for both types of approximations, FPF and FNF, we also checked whether the approximations returned by GRosSo had the additional guarantees described in Sect. 4.5, and, in all the runs, we found that such additional guarantees were always respected.

For the EP and DP with guarantees on the false positives, the results obtained with $\varepsilon = 0$ are very close to the ones reported by Table 2, 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, while for the EP with guarantees on the false negatives, the results obtained with $\varepsilon = 0$ and $\varepsilon = 0.05$ are very close to the ones reported by Table 4, with a percentage of reported patterns between 0.06 and 0.78.

For the EP and DP scenarios with guarantees on the false positives, we also performed an additional 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 a sequence using $\theta = 0$ and $\varepsilon = 0$. Let us 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 an 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 or false negatives. Thus, techniques like the one introduced in this work are necessary to find large sets of SRPs without false positives or false negatives. In addition, GRosSo is an effective tool to find rigorous approximations of the statistically robust sequential patterns.

7.3.2 Itemsets

Here, we report the results of our experimental evaluation using pseudo-artificial datasets for the itemset mining task. Starting from the 2005(T1-T3) sequence of datasets for the sequential pattern mining task, we first generated the corresponding sequence of datasets, 2005(T1-T3)IT, for the itemset mining task. For each dataset in the sequence 2005(T1-T3), we generated a new dataset taking the union of the items in each transaction of the dataset, e.g., a sequential transaction $\tau = \langle\{1\}, \{2\}, \{6, 7\}, \{2\}\rangle$ becomes a transaction $\tau' = \{1, 2, 6, 7\}$. Then, we considered the 2005(T1-T3)IT sequence as ground truth for the itemsets, and we performed the same experimental evaluation described in Sect. 7.3.1 for the sequential patterns. We denote by $\mathcal{Q}_1^n$, $\mathcal{Q}_1^{n\times 2}$, and $\mathcal{Q}_1^{n\times 3}$ the analogous to $\mathcal{S}_1^n$, $\mathcal{S}_1^{n\times 2}$, and $\mathcal{S}_1^{n\times 3}$, but for itemsets. Table 5 reports the average results for mining FPF approximations of the EP and DP, Table 6 reports the average results for mining FPF approximations of the SP, while Table 7 reports the average results for mining FNF approximations of the EP. The results show that, for almost all parameters, the sets of patterns mined in the pseudo-artificial datasets only

Table 5 Results on pseudo-artificial datasets for EP and DP for the itemset mining task with guarantees on the false positives

Datasets $\mathcal{D}_1^n$	ε	θ	EP				DP			
			$\lvert GT\rvert$	T.FP$_f$	T.FP$_g$	$\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$	$\lvert GT\rvert$	T.FP$_f$	T.FP$_g$	$\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$
$\mathcal{Q}_1^n$	0.01	0.3	26	20%	0%	0.17	6	0%	0%	0.33
		0.2	48	60%	0%	0.14	60	80%	0%	0.04
$\mathcal{Q}_1^{n\times 2}$	0.01	0.3	26	0%	0%	0.36	6	100%	0%	0.67
		0.2	48	100%	0%	0.20	60	100%	0%	0.07
$\mathcal{Q}_1^{n\times 3}$	0.01	0.3	26	100%	0%	0.42	6	100%	0%	0.67
		0.2	48	100%	0%	0.23	60	100%	0%	0.07

See Table 2 for the meaning of the values

Table 6 Results on pseudo-artificial datasets for SP for the itemset mining task with guarantees on the false positives

Datasets $\mathcal{D}_1^n$	α	θ	$\lvert GT\rvert$	T.FP$_f$	T.FP$_g$	$\lvert\mathcal{A}_P\rvert/\lvert GT\rvert$
$\mathcal{Q}_1^n$	0.1	0.3	419	80%	0%	0.65
		0.2	10541	60%	0%	0.70
$\mathcal{Q}_1^{n\times 2}$	0.1	0.3	419	0%	0%	0.81
		0.2	10541	0%	0%	0.76
$\mathcal{Q}_1^{n\times 3}$	0.1	0.3	419	0%	0%	0.82
		0.2	10541	0%	0%	0.78

See Table 3 for the meaning of the values

considering the observed frequency of the patterns contain false positives or false negatives with high probability. Instead, the patterns returned by GRoSSo do not contain false positives or false negatives in all the runs, as observed for the sequential patterns. In addition, all the approximations of the EP reported by GRoSSo respected the additional guarantees introduced in Sect. 4.5. All these results emphasize that considering the observed frequency is not enough to find large sets of SRPs without false positives or false negatives, and that GRoSSo is an effective tool also to find rigorous approximations of the statistically robust itemsets. Comparing these results with the ones obtained for the sequential patterns, it is interesting to notice that in the EP and DP scenarios, almost always the returned statistically robust itemsets are less than the corresponding statistically robust sequential patterns, in particular for the DP, and that also the percentages of reported patterns are lower for the itemsets. Instead, in the SP scenario, more statistically robust itemsets are returned, always with higher percentages of reported patterns.

7.4 Results with real datasets

Here, we report the results of GRoSSo for mining statistically robust sequential patterns from the Netflix real datasets. First, we report and discuss the results with guarantees on the false positives for all the three types of SRPs. 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 8 shows the results for the EP and DP with guarantees on the false positives. In the EP scenario, for the sequences of datasets composed by four datasets

Table 7 Results on pseudo-artificial datasets for EP for the itemset mining task with guarantees on the false negatives

Datasets $\mathcal{D}_1^n$	ε	θ	$\lvert GT\rvert$	T.FN$_f$	T.FN$_g$	$\lvert GT\rvert/\lvert\mathcal{A}_N\rvert$
$\mathcal{Q}_1^n$	0.01	0.3	26	80%	0%	0.20
		0.2	48	60%	0%	0.02
$\mathcal{Q}_1^{n\times 2}$	0.01	0.3	26	0%	0%	0.24
		0.2	48	0%	0%	0.04
$\mathcal{Q}_1^{n\times 3}$	0.01	0.3	26	0%	0%	0.30
		0.2	48	0%	0%	0.05

See Table 4 for the meaning of the values

Table 8 Results on real datasets for EP and DP for the sequential pattern mining task with guarantees on the false positives

Datasets $\mathcal{D}_1^n$	ε	EP		DP	
		$\lvert\mathcal{A}_P\rvert$	Avg$\lvert\lvert s\rvert\rvert$	$\lvert\mathcal{A}_P\rvert$	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

The table reports: $\mathcal{D}_1^n$: name of the sequences of datasets; ε: emerging threshold; for both the EP and DP: $\lvert\mathcal{A}_P\rvert$: number of returned SRPs; Avg$\lvert\lvert s\rvert\rvert$: average item-length of the returned SRPs

(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 an 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 a 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,

Table 9 Results on real datasets for SP for the sequential pattern mining task with guarantees on the false positives

Datasets $\mathcal{D}_1^n$	α	θ	$\lvert\mathcal{A}_P\rvert$	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

The table reports: α : stability threshold; θ : minimum frequency threshold. See Table 8 for the meaning of the other values

Table 10 Results on real datasets for EP for the sequential pattern mining task with guarantees on the false positives and on the false negatives

Datasets $\mathcal{D}_1^n$	θ	ε	FPF		FNF	
			$\lvert\mathcal{A}_P\rvert$	Avg$\lvert\lvert s\rvert\rvert$	$\lvert\mathcal{A}_N\rvert$	Avg$\lvert\lvert s\rvert\rvert$
2004(Q1–Q4)	0.3	0	21	2.1	119	2.3
		0.01	15	2.1	114	2.3
		0.05	1	1.0	80	2.6
2005(Q1–Q4)	0.3	0	2	1.5	13	2.0
		0.01	1	1.0	11	2.1
		0.05	0	/	4	1.8
2004(T1–T3)	0.3	0	74	2.7	309	2.6
		0.01	73	2.7	298	2.6
		0.05	63	2.7	243	2.8
2005(T1–T3)	0.3	0	9	2.0	43	2.0
		0.01	8	2.3	41	2.0
		0.05	3	2.2	25	2.0
2003–2005(Y)	0.5	0	46	2.2	255	2.1
		0.01	44	2.1	242	2.1
		0.05	41	2.1	227	2.1

The table reports: $\mathcal{D}_1^n$: name of the sequences of datasets; θ : minimum frequency threshold; ε: emerging threshold; for both the FPF and FNF approximations: $\lvert\mathcal{A}\rvert$: number of returned SRPs; Avg$\lvert\lvert s\rvert\rvert$: average item-length of the returned SRPs

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 9 shows the results for the SP with guarantees on the false positives. 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). Such sequential patterns probably represent some classical movies that people always watch with the same frequency through time.

To conclude, we report the results for the EP with guarantees on the false negatives. As discussed in Sect. 4.4, we decided to consider only patterns p with $t_{\pi_n}(p) \geq \theta$ to reduce the amount of starting candidates. In order to compare the size of FPF and FNF approximations in the same scenario, we also executed the same experiments for the EP with guarantees on the false positives using a minimum frequency threshold. Table 10 reports the parameters and the results of such experiments. These results show that GRoSSo can detect EP when one is interested in finding FPF or FNF approximations, and it is possible to notice that almost always the FNF approximations contain a number of EP that is from 4 to 6 times larger than the corresponding number in the FPF approximations.

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.

8 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 a problem and described GRoSSo, an algorithm to identify approximations of the SRPs with probabilistic guarantees on false discoveries or on false negatives, and we applied it to identify statistically robust *sequential* patterns and statistically robust *itemsets*. 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.

Acknowledgements Part of this work was supported by the MIUR, the Italian Ministry of Education, University and Research, under PRIN Project no. 20174LF3T8 AHeAD (Efficient Algorithms for HArnessing Networked Data) and the initiative "Departments of Excellence" (Law 232/2016), by the University of Padova under project SEED 2020 RATED-X, and by the project "POR FESR 2014-2020: Piattaforma Integrata di Cruscotti Intelligenti per la Gestione Avanzata della Qualità e del Marketing Innovativo".

Funding Open access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Tonon A, Vandin F (2020) gRosSo: mining statistically robust patterns from a sequence of datasets. In: Proceedings of the 20th IEEE international conference on data mining, IEEE, ICDM'20, pp 551–560
2. Han J, Cheng H, Xin D, Yan X (2007) Frequent pattern mining: current status and future directions. Data Min Knowl Disc 15(1):55–86
3. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. SIGMOD Rec 22:207–216
4. Agrawal R, Srikant R (1995) Mining sequential patterns. In: Proceedings of the 11th international conference on data engineering, IEEE, ICDE'95, pp 3–14
5. Klösgen W (1992) Problems for knowledge discovery in databases and their treatment in the statistics interpreter explora. Int J Intell Syst 7(7):649–673
6. Ahmed NK, Neville J, Rossi RA, Duffield N (2015) Efficient graphlet counting for large networks. In: Proceedings of the 2015 IEEE international conference on data mining, IEEE, ICDM'15, pp 1–10
7. Hämäläinen W, Webb GI (2019) A tutorial on statistically sound pattern discovery. Data Min Knowl Disc 33(2):325–377
8. Pellegrina L, Riondato M, Vandin F (2019) Hypothesis testing and statistically-sound pattern mining. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp 3215–3216
9. Komiyama J, Ishihata M, Arimura H, Nishibayashi T, Minato SI (2017) Statistical emerging pattern mining with multiple testing correction. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 897–906
10. Llinares-López F, Sugiyama M, Papaxanthos L, Borgwardt K (2015) Fast and memory-efficient significant pattern mining via permutation testing. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining, pp 725–734
11. Pellegrina L, Riondato M, Vandin F (2019) SPuManTE: Significant pattern mining with unconditional testing. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1528–1538
12. Pellegrina L, Vandin F (2020) Efficient mining of the most significant patterns with permutation testing. Data Min Knowl Disc 34:1201–1234
13. Gwadera R, Crestani F (2010) Ranking sequential patterns with respect to significance. In: Zaki MJ, Yu JX, Ravindran B, Pudi V (eds) Advances in knowledge discovery and data mining, PAKDD 2010, pp 286–299
14. Low-Kam C, Raïssi C, Kaytoue M, Pei J (2013) Mining statistically significant sequential patterns. In: Proceedings of the 13th IEEE international conference on data mining, IEEE, ICDM'13, pp 488–497
15. Tonon A, Vandin F (2019) Permutation strategies for mining significant sequential patterns. In: Proceedings of the 19th IEEE international conference on data mining, IEEE, ICDM'19, pp 1330–1335
16. Dong G, Li J (1999) Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining, pp 43–52
17. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large data bases, VLDB'94, pp 487–499
18. Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min Knowl Disc 8(1):53–87
19. Srikant R, Agrawal R (1996) Mining sequential patterns: generalizations and performance improvements. In: Proceedings of the 5th international conference on extending database technology, EDBT'96, pp 1–17
20. Pei J, Han J, Mortazavi-Asl B, Wang J, Pinto H, Chen Q, Dayal U, Hsu MC (2004) Mining sequential patterns by pattern-growth: the prefixspan approach. IEEE Trans Knowl Data Eng 16(11):1424–1440
21. Servan-Schreiber S, Riondato M, Zgraggen E (2020) ProSecCo: progressive sequence mining with convergence guarantees. Knowl Inf Syst 62:1313–1340
22. Santoro D, Tonon A, Vandin F (2020) Mining sequential patterns with VC-dimension and rademacher complexity. Algorithms 13(5):123
23. Riondato M, Upfal E (2014) Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. ACM Trans Knowl Discov Data (TKDD) 8(4):1–32

24. Riondato M, Vandin F (2014) Finding the true frequent itemsets. In: Zaki MJ, Obradovic Z, Tan P, Banerjee A, Kamath C, Parthasarathy S (eds) Proceedings of the 2014 SIAM international conference on data mining, SIAM, pp 497–505
25. Zhu F, Yan X, Han J, Philip SY, Cheng H (2007) Mining colossal frequent patterns by core pattern fusion. In: 2007 IEEE 23rd international conference on data engineering, pp 706-715
26. Egho E, Gay D, Boullé M, Voisine N, Clérot F (2017) A user parameter-free approach for mining robust sequential classification rules. Knowl Inf Syst 52(1):53–81
27. Vapnik VN, Chervonenkis AY (2015) On the uniform convergence of relative frequencies of events to their probabilities. In: Vovk V, Papadopoulos H, Gammerman A (eds) Measures of complexity. Springer, Cham, pp 11–30
28. Boucheron S, Bousquet O, Lugosi G (2005) Theory of classification: a survey of some recent advances. ESAIM Probab Stat 9:323–375
29. Mitzenmacher M, Upfal E (2017) Probability and computing: randomization and probabilistic techniques in algorithms and data analysis. Cambridge University Press, Cambridge
30. Li Y, Long PM, Srinivasan A (2001) Improved bounds on the sample complexity of learning. J Comput Syst Sci 62(3):516–527
31. Egho E, Raïssi C, Calders T, Jay N, Napoli A (2015) On measuring similarity for sequences of itemsets. Data Min Knowl Discov 29(3):732–764
32. Fournier-Viger P, Lin JCW, Gomariz A, Gueniche T, Soltani A, Deng Z, Lam HT (2016) The SPMF open-source data mining library version 2. In: Proceedings of 19th European conference on machine learning and principles and practice of knowledge discovery and data mining (Part III), ECML PKDD'16

Andrea Tonon received the Laurea Triennale degree (2015) in Information Engineering from the University of Padova, Italy, and the Laurea Magistrale degree (summa cum laude, 2018) in Computer Engineering from the University of Padova, Italy. He is currently a Ph.D. student at the Department of Information Engineering of the University of Padova, Italy. His research interests focus on algorithms for data mining. In particular, the development of novel efficient methods for knowledge and pattern discovery from large collections of sequential data, often exploiting distributed/parallel approaches and statistically sound algorithms.

Fabio Vandin received the Laurea Triennale degree (summa cum laude, 2004) and the Laurea Specialistica degree (summa cum laude, 2006) in Computer Engineering from the University of Padova, Italy. He received the Ph.D. (2010) in Information Engineering from the University of Padova, Italy. He has been a postdoctoral researcher at Brown University and an Assistant Professor at the University of Southern Denmark. Since 2020 he is Professor at the Department of Information Engineering of the University of Padova. His research interests are in algorithms for data mining and machine learning and their application for the analysis of large biological datasets, in particular cancer genomic datasets. He has authored over 70 papers in international peer-reviewed venues, and he has used his methods for analyses published in Nature, Nature Genetics, Cell, NEJM. He has been a co-PI for two projects funded by the NSF (USA) and a participant in several European projects.