Finding the True Frequent Itemsets*

Matteo Riondato [†] Fabio Vandin [‡]

Wednesday 9th April, 2014

Abstract

Frequent Itemsets (FIs) mining is a fundamental primitive in knowledge discovery. It requires to identify all itemsets appearing in at least a fraction θ of a transactional dataset $\mathcal{D}$. Often though, the ultimate goal of mining $\mathcal{D}$ is not an analysis of the dataset *per se*, but the understanding of the underlying process that generated it. Specifically, in many applications $\mathcal{D}$ is a collection of samples obtained from an unknown probability distribution π on transactions, and by extracting the FIs in $\mathcal{D}$ one attempts to infer itemsets that are frequently (i.e., with probability at least θ) generated by π, which we call the True Frequent Itemsets (TFIs). Due to the inherently stochastic nature of the generative process, the set of FIs is only a rough approximation of the set of TFIs, as it often contains a huge number of *false positives*, i.e., spurious itemsets that are not among the TFIs. In this work we design and analyze an algorithm to identify a threshold $\hat{\theta}$ such that the collection of itemsets with frequency at least $\hat{\theta}$ in $\mathcal{D}$ contains only TFIs with probability at least $1-\delta$, for some user-specified δ. Our method uses results from statistical learning theory involving the (empirical) VC-dimension of the problem at hand. This allows us to identify almost all the TFIs without including any false positive. We also experimentally compare our method with the direct mining of $\mathcal{D}$ at frequency θ and with techniques based on widely-used standard bounds (i.e., the Chernoff bounds) of the binomial distribution, and show that our algorithm outperforms these methods and achieves even better results than what is guaranteed by the theoretical analysis.

Keywords: Frequent itemsets, VC-dimension, False positives, Distribution-free methods, Frequency threshold identification, Pattern mining, Significant patterns.

1 Introduction

The extraction of association rules is one of the fundamental primitives in data mining and knowledge discovery from large databases [2]. In its most general definition, the problem can be reduced to identifying frequent sets of items, or *frequent itemsets*, appearing in at least a fraction θ of all transactions in a dataset, where θ is provided in input by the user. Frequent itemsets and association rules are not only of interest for classic data mining applications (e.g., market basket analysis), but are also useful for further data analysis and mining task, including clustering, classification, and indexing [3, 4].

In most applications, the set of frequent itemsets is not interesting *per se*. Instead, the mining results are used to infer properties of the *underlying process* that generated the dataset. Consider for example the following scenario: a team of researchers would like to identify frequent associations (i.e., itemsets) between preferences among Facebook users. To this end, they set up an online survey which is filled out by a *small fraction* of Facebook users (some users may even take the survey multiple times). Using this information, the researchers want to infer the associations (itemsets) that are frequent for the *entire* Facebook population. In fact, the whole Facebook population and the online survey define the underlying *process* that generated the dataset *observed* by the researchers. In this work we are interested in answering the following question: how

[*] Work supported in part by NSF grant IIS-1247581. This is an extended version of the work that appeared as [1].

[†] Department of Computer Science, Brown University, Providence, RI, USA. matteo@cs.brown.edu . Contact author.

[‡] Department of Computer Science, Brown University, Providence, RI, USA and Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark. vandinfa@imada.sdu.dk .

can we use the latter (the observed dataset) to identify itemsets that are frequent in the former (the whole population)? This is a very natural question, as is the underlying assumption that the observed dataset is *representative* of the generating process. For example, in market basket analysis, the observed purchases of customers are used to infer the future purchase habits of all customers while assuming that the purchase behavior that generated the dataset is representative of the one that will be followed in the future.

A natural and general model to describe these concepts is to assume that the transactions in the dataset $\mathcal{D}$ are *independent identically distributed* (i.i.d.) samples from an *unknown* probability distribution π defined on all possible transactions built on a set of items. Since π is fixed, each itemset A has a fixed (unknown) *probability* $t_\pi(A)$ to appear in a transaction sampled from π. We call $t_\pi(A)$ the *true frequency* of A (w.r.t. π). The true frequency corresponds to the fraction of transactions that would contain the itemset A among an hypothetical infinite set of transactions. The real goal of the mining process is then to identify itemsets that have true frequency t_π at least θ, i.e., the *True Frequent Itemsets* (TFIs). In the market basket analysis example, $\mathcal{D}$ contains the observed purchases of customers, the *unknown* distribution π describes the purchase behavior of the customers as a whole, and we want to analyze $\mathcal{D}$ to find the itemsets that have probability (i.e., true frequency) at least θ to be bought by a customer. Note that we made no assumption on π, except from the fact that the transactions in the dataset are i.i.d. samples from π. This is in contrast to other settings that assume that the generative distribution π is such that items appear in transactions generated by π totally or partially independently from each other.

Since $\mathcal{D}$ represents only a *finite* sample from π, the set F of frequent itemsets of $\mathcal{D}$ w.r.t. θ only provides an *approximation* of the True Frequent Itemsets: due to the stochastic nature of the generative process, the set F may contain a number of *false positives*, i.e., itemsets that appear among the frequent itemsets of $\mathcal{D}$ but whose *true* frequency is smaller than θ. At the same time, some itemsets with true frequency greater than θ may have a frequency in $\mathcal{D}$ that is *smaller* than θ (*false negatives*), and therefore not be in F. This implies that one can not aim at identifying *all and only* the itemsets having true frequency at least θ. Even worse, from the data analyst's point of view, there is *no guarantee or bound on the number of false positives* reported in F. Consider the following scenario as an example. Let A and B be two (disjoint) sets of pairs of items. The set A contains 1,000 disjoint pairs, while B contains 10,000 disjoint pairs. Let π be such that, for any pair $(a, a') \in A$, we have $t_\pi((a, a')) = 0.1$, and for any pair $(b, b') \in B$, we have $t_\pi((b, b')) = 0.09$. Let $\mathcal{D}$ be a dataset of 10,000 transactions sampled from π. We are interested in finding pairs of items that have true frequency at least $\theta = 0.095$. If we extract the pairs of items with frequency at least θ in $\mathcal{D}$, it is easy to see that in expectation 50 of the 1,000 pairs from A will have frequency in $\mathcal{D}$ *below* 0.095, and in expectation 400 pairs from B will have frequency in $\mathcal{D}$ *above* 0.095. Therefore, the set of pairs that have frequency at least θ in $\mathcal{D}$ does *not* contain some of the pairs that have true frequency at least θ (false negatives), but includes a huge number of pairs that have true frequency smaller than θ (false positives).

In general, one would like to avoid false positives and at the same time find as many TFIs as possible. These are somewhat contrasting goals, and care must be taken to achieve a good balance between them. A naïve but *overly conservative* method to avoid false positives involves the use of *Chernoff and union bounds* [5]. Let A be an itemset in $\mathcal{D}$. The quantity $|\mathcal{D}|f_{\mathcal{D}}(A)$ is a random variable with Binomial distribution $\mathcal{B}(|\mathcal{D}|, t_\pi(A))$. It is possible to use standard methods like the Chernoff and the union bounds to bound the deviation of the frequencies in the dataset of *all* itemsets from their expectations. These tools can be used to compute a value $\hat{\theta}$ such that the probability that a non-true frequent itemset B has frequency greater or equal to $\hat{\theta}$ is at most $1 - \delta$, for some $\delta \in (0, 1)$. This method has the following serious drawback: in order to achieve such guarantee, it is *necessary* to bound the deviation of the frequencies of *all itemsets possibly appearing in the dataset* [6]. This means that, if the transactions are built on a set of n items, the union bound must be taken over all $2^n - 1$ potential itemsets, even if some or most of them may appear with very low frequency or not at all in samples from π. As a consequence, the chosen value of $\hat{\theta}$ is extremely *conservative*, despite being sufficient to avoid the inclusion of false positives in mining results. The collection of itemsets with frequency at least $\hat{\theta}$ in $\mathcal{D}$, although consisting (probabilistically) only of TFIs, only contains a *very small* portion of them, due to the overly conservative choice of $\hat{\theta}$. (The results of our experimental evaluation in Sect. 6 clearly show the limitations of this method.) More refined algorithms are therefore needed to achieve the correct balance between the contrasting goals of avoiding false positives and finding as many TFIs as possible.

1.1 Our contributions.

The contributions of this work are the following:

- We formally define the problem of mining the *True Frequent Itemsets* w.r.t. a minimum threshold θ, and we develop and analyze an algorithm to *identify a value $\hat{\theta}$ such that, with probability at least $1-\delta$, all itemsets with frequency at least $\hat{\theta}$ in the dataset have true frequency at least θ*. Our method is completely *distribution-free*, i.e., it does not make *any* assumption about the unknown generative distribution π. By contrast, existing methods to assess the significance of frequent patterns after their extraction require a well specified, limited generative model to characterize the significance of a pattern. Our method also allows to include additional prior information about the distribution π, when available, to obtain even higher accuracy.
- We analyze our algorithm using results from *statistical learning theory* and *optimization*. We define a range set associated to a collection of itemsets and give an upper bound to its (empirical) VC-dimension and a procedure to compute this bound, showing an interesting connection with the Set-Union Knapsack Problem (SUKP) [7]. To the best of our knowledge, ours is the first work to apply these techniques to the field of TFIs, and in general the first application of the sample complexity bound based on *empirical* VC-dimension to the field of data mining.
- We implemented our algorithm and assessed its performances on simulated datasets with properties – number of items, itemsets frequency distribution, etc.– similar to real datasets. We computed the fraction of TFIs contained in the set of frequent itemsets in $\mathcal{D}$ w.r.t. $\hat{\theta}$, and the number of false positives, if any. The results show that the algorithm is even *more accurate* than the theory guarantees, since *no false positive* is reported in any of the many experiments we performed, and moreover allows the *extraction of almost all TFIs*. We also compared the set of itemsets computed by our method to those obtained with the "Chernoff and union bounds" method presented in the introduction, and found that our algorithm *vastly outperforms* it.

Outline. In Sect. 2 we review relevant previous contributions. Sections 3 and 4 contain preliminaries to formally define the problem and key concepts that we will use throughout the work. Our proposed algorithm is described and analyzed in Sect. 5. We present the methodology and results of our experimental evaluation in Sect. 6. Conclusions and future directions can be found in Sect. 7.

2 Previous work

Given a sufficiently low minimum frequency threshold, traditional itemsets mining algorithms can return a collection of frequent patterns so large to become almost uninformative to the human user. The quest for reducing the number of patterns given in output has been developing along two different different directions suggesting non-mutually-exclusive approaches. One of these lines of research starts from the observation that the information contained in a set of patterns can be compressed with or without loss to a much smaller collection. This lead to the definition of concepts like *closed*, *maximal*, *non-derivable* itemsets. This approach is orthogonal to the one we take and we refer the interested reader to the survey by Calders et al. [8].

The intuition at the basis of the second approach to reduce the number of output patterns consists in observing that a large portion of the patterns may be *spurious*, i.e., not actually *interesting* but only a consequence of the fact that the dataset is just a sample from the underlying process that generates the data, the understanding of which is the ultimate goal of data mining. This observation led to a prolification of interestingness measures. In this work we are interested in a very specific definition of interestingness that is based on statistical properties of the patterns. We refer the reader to the surveys on different readers by Han et al. [4, Sect. 3] and Geng and Hamilton [9]. We remark that, as noted by Liu et al. [10], that the use of the minimum support threshold θ, reflecting the level of domain significance, is complementary to the use of interestingness measures, and that "statistical significance measures and domain significance measures should be used together to filter uninteresting rules from different perspectives". The algorithm we present can be seen as a method to filter out patterns that are not interesting according to the measure represented by the true frequency.

A number of works explored the idea to use statistical properties of the patterns in order to assess their interestingness. While this is not the focus of our work, some of the techniques and models proposed are relevant to our framework. Most of these works are focused on association rules, but some results can be applied to itemsets. In these works, the notion of interestingness is related to the deviation between the observed frequency of a pattern in the dataset and its expected support in a random dataset generated according to a well-defined probability distribution that can incorporate prior belief and that can be updated during the mining process to ensure that the most "surprising" patterns are extracted. In many previous works, the probability distribution was defined by a simple independence model: an item belongs to a transaction independently from other items [6, 11–15]. In contrast, our work does not impose any restriction on the probability distribution generating the dataset, with the result that our method is as general as possible.

Kirsch et al. [6] developed a multi-hypothesis testing procedure to identify the best support threshold such that the number of itemsets with at least such support deviates significantly from its expectation in a random dataset of the same size and with the same frequency distribution for the individual items. In our work, the minimum threshold θ is an input parameter fixed by the user, and we identify a threshold $\hat{\theta} \geq \theta$ to guarantee that the collection of FIs w.r.t. $\hat{\theta}$ does not contain any false discovery.

Gionis et al. [14] present a method to create random datasets that can act as samples from a distribution satisfying an assumed generative model. The main idea is to swap items in a given dataset while keeping the length of the transactions and the sum over the columns constant. This method is only applicable if one can actually derive a procedure to perform the swapping in such a way that the generated datasets are indeed random samples from the assumed distribution. For the problem we are interested in, such procedure is not available and indeed it would be difficult to obtain a procedure that is valid for any distribution, given that we aim at developing a method that makes no assumption on the distribution. Considering the same generative model, Hanhijärvi [16] presents a direct adjustment method to bound the probability of false discoveries by taking into consideration the actual number of hypotheses to be tested.

Webb [17] proposes the use of established statistical techniques to control the probability of false discoveries. In one of these methods (called holdout), the available data are split into two parts: one is used for pattern discovery, while the second is used to verify the significance of the discovered patterns, testing one statistical hypothesis at a time. A new method (layered critical values) to choose the critical values when using a direct adjustment technique to control the probability of false discoveries is presented by Webb [18] and works by exploiting the itemset lattice. The method we present instead identify a threshold frequency such that all the itemsets with frequency above the threshold are TFIs. There is no need to test each itemset separately and no need to split the dataset.

Liu et al. [10] conduct an experimental evaluation of direct corrections, holdout data, and random permutations methods to control the false positives. They test the methods on a very specific problem (association rules for binary classification).

In contrast with the methods presented in the works above, ours does not employ an explicit direct correction depending on the number of patterns considered as it is done in traditional multiple hypothesis testing settings. It instead uses the entire available data to obtain more accurate results,without the need to re-sampling it to generate random datasets or to split the dataset in two parts, being therefore more efficient computationally.

2.1 The Vapnik-Chervonenkis dimension

The *Vapnik-Chervonenkis dimension* was first introduced in a seminal article [19] on the convergence of probability distributions, but it was only with the work of Haussler and Welzl [20] and Blumer et al. [21] that it was applied to the field of learning. Boucheron et al. [22] present a good survey of the field with many recent advances. Since then, VC-dimension has encountered enormous success and application in the fields of computational geometry [23, 24] and machine learning [25, 26]. Other applications include database management and graph algorithms. In the former, it was used in the context of constraint databases to compute good approximations of aggregate operators [27]. VC-dimension-related results were also recently applied in the field of database privacy by Blum et al. [28] to show a bound on the number of queries needed for an attacker to learn a private concept in a database. Gross-Amblard [29] showed that content with unbounded VC-dimension can not be watermarked for privacy purposes. Riondato et al. [30] computed an upper bound to the VC-dimension of classes of SQL queries and used it to develop a sampling-based algorithm for estimating the size of the output (selectivity) of queries run on a dataset. The work of Riondato and Upfal [31] on the VC-dimension of frequent itemsets and association rules inspired us, although we deal with a different problem, different goals, and use different tools. In the graph algorithms literature, VC-Dimension has been used to develop algorithms to efficiently detect network failures [32, 33], balanced separators [34], events in a sensor networks [35], compute approximate shortest paths [36], and estimate betweenness centrality [37].

3 Preliminaries

In this section we introduce the definitions, lemmas, and tools that we will use throughout the work, providing the details that are needed in later sections.

3.1 Itemsets mining

Given a ground set $\mathcal{I}$ of *items*, let π be a probability distribution on $2^{\mathcal{I}}$. A *transaction* $\tau \subseteq \mathcal{I}$ is a single sample drawn from π. The *length* $|\tau|$ of a transaction τ is the number of items in τ. A *dataset* $\mathcal{D}$ is a bag of n transactions $\mathcal{D} = \{\tau_1, \ldots, \tau_n \ : \ \tau_i \subseteq \mathcal{I}\}$, i.e., of n *independent identically distributed* (i.i.d.) samples from π. We call a subset of $\mathcal{I}$ an *itemset*. For any itemset A, let $T(A) = \{\tau \subseteq \mathcal{I} \ : \ A \subseteq \tau\}$ be the *support set* of A. The members of the set $T(A)$ are all the transactions built on $\mathcal{I}$ that contain the itemset A. We define the *true frequency* $t_\pi(A)$ of A with respect to π as the probability that a transaction sampled from π contains A:

$$t_\pi(A) = \sum_{\tau \in T(A)} \pi(\tau) \ .$$

Analogously, given a (observed) dataset $\mathcal{D}$, let $T_\mathcal{D}(A)$ denote the set of transactions in $\mathcal{D}$ containing A. The *frequency* of A in $\mathcal{D}$ is the fraction of transactions in $\mathcal{D}$ that contain A: $f_\mathcal{D}(A) = |T_\mathcal{D}(A)|/|\mathcal{D}|$. It is easy to see that $f_\mathcal{D}(A)$ is the *empirical average* (and an *unbiased estimator*) for $t_\pi(A)$: $\mathbf{E}[f_\mathcal{D}(A)] = t_\pi(A)$.

Traditionally, the interest has been on extracting the set of *Frequent Itemsets* (FIs) from $\mathcal{D}$ with respect to a minimum frequency threshold $\theta \in (0, 1]$ [2], that is, the set

$$\mathsf{FI}(\mathcal{D}, \mathcal{I}, \theta) = \{A \subseteq \mathcal{I} \ : \ f_\mathcal{D}(A) \geq \theta\} \ .$$

In most applications the final goal of data mining is to gain a better understanding of the *process generating the data*, i.e., of the distribution π, through the true frequencies t_π, which are *unknown* and only approximately reflected in the dataset $\mathcal{D}$. Therefore, we are interested in finding the itemsets with *true* frequency t_π at least θ for some $\theta \in (0, 1]$. We call these itemsets the *True Frequent Itemsets* (TFIs) and denote their set as

$$\mathsf{TFI}(\pi, \mathcal{I}, \theta) = \{A \subseteq \mathcal{I} \ : \ t_\pi(A) \geq \theta\} \ .$$

If one is only given a *finite* number of random samples (the dataset $\mathcal{D}$) from π as it is usually the case, one can not aim at finding the exact set $\mathsf{TFI}(\pi, \mathcal{I}, \theta)$: no assumption can be made on the set-inclusion relationship between $\mathsf{TFI}(\pi, \mathcal{I}, \theta)$ and $\mathsf{FI}(\mathcal{D}, \mathcal{I}, \theta)$, because an itemset $A \in \mathsf{TFI}(\pi, \mathcal{I}, \theta)$ may not appear in $\mathsf{FI}(\mathcal{D}, \mathcal{I}, \theta)$, and vice versa. One can instead try to *approximate* the set of TFIs. This is what we are interested in this work.

Goal. Given an user-specified parameter $\delta \in (0,1)$, we aim at providing a threshold $\hat{\theta} \geq \theta$ such that $\mathcal{C} = \mathsf{FI}(\mathcal{D}, \mathcal{I}, \hat{\theta})$ *well approximates* $\mathsf{TFI}(\pi, \mathcal{I}, \theta)$, in the sense that

1. With probability at least $1-\delta$, $\mathcal{C}$ does not contain any false positive:
$$\Pr(\exists A \in \mathcal{C} \ : \ t_\pi(A) < \theta) < \delta \ .$$
2. $\mathcal{C}$ contains as many TFIs as possible.

The method we present does not make *any* assumption about π. It uses information from $\mathcal{D}$, and guarantees a small probability of false positives while achieving a high success rate.

3.2 Vapnik-Chervonenkis dimension

Let D be a domain and $\mathcal{R}$ be a collection of subsets from D. We call $\mathcal{R}$ a *range set on* D. The Vapnik-Chernovenkis (VC) Dimension of $\mathcal{R}$ is a measure of its complexity or expressiveness [19]. A finite bound on the VC-dimension implies a bound on the number of random samples required for approximate the relative sizes of the subsets in $\mathcal{R}$. We outline here some basic definitions and results and refer the reader to the works of Mohri et al. [38, Chap. 3], Boucheron et al. [22, Sect. 3], and Vapnik [39] for more details on VC-dimension. See Sect. 2.1 for applications of VC-dimension in computer science.

Given $B \subseteq D$, the *projection of* $\mathcal{R}$ *on* B is the set $P_{\mathcal{R}}(B) = \{B \cap A \ : \ A \in \mathcal{R}\}$. We say that the set B is *shattered* by $\mathcal{R}$ if $P_{\mathcal{R}}(B) = 2^B$.

Definition 1. Given a set $B \subseteq D$, the *empirical Vapnik-Chervonenkis (VC) dimension of* $\mathcal{A}$ *on* B, denoted as $\mathsf{EVC}(\mathcal{R}, B)$ is the cardinality of the largest subset of B that is shattered by $\mathcal{R}$. The *VC-dimension of* $\mathcal{R}$ is defined as $\mathsf{VC}(\mathcal{R}) = \mathsf{EVC}(\mathcal{R}, D)$.

Note that an arbitrary large range set $\mathcal{R}$ defined on an arbitrary large domain D can have a bounded VC-dimension. A simple example is the family of intervals in $[0,1]$ (i.e. D is all the points in $[0,1]$ and $\mathcal{R}$ all the intervals $[a,b]$, such that $0 \leq a \leq b \leq 1$). Let $A = \{x, y, z\}$ be the set of three points $0 < x < y < z < 1$. No interval in $\mathcal{R}$ can define the subset $\{x, z\}$ so the VC-dimension of this range set is less than 3 [24, Lemma 10.3.1]. Another example is shown in Fig. 1.

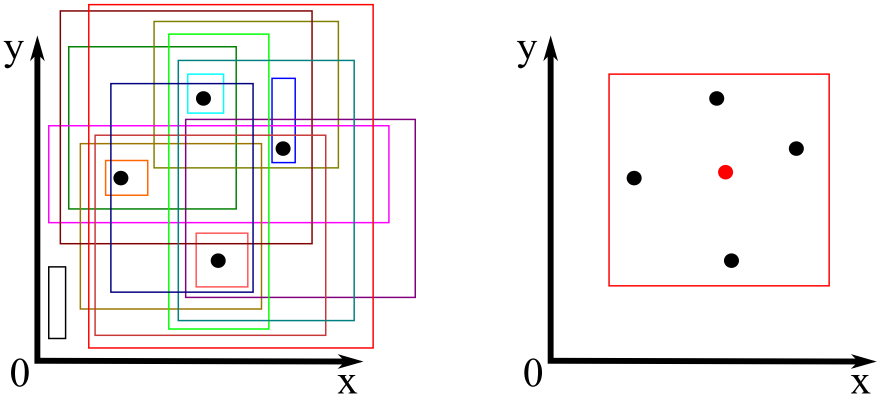

Figure 1: Example of range set and VC-dimension. The domain is the plane $\mathbb{R}^2$ and the range set is the set of all *axis-aligned rectangles*. The figure on the left shows graphically that it is possible to shatter a set of four points using 16 rectangles. On the right instead, one can see that it is impossible to shatter five points, as, for any choice of the five points, there will always be one (the red point in the figure) that is internal to the convex hull of the other four, so it would be impossible to find an axis-aligned rectangle containing the four points but not the internal one. Hence $\mathsf{VC}(\mathcal{R}) = 4$.

The main application of (empirical) VC-dimension in statistics and learning theory is in computing the number of samples needed to approximate the probabilities associated to the ranges through their empirical averages. Formally, let $X_1^k = (X_1, \ldots, X_k)$ be a collection of independent identically distributed random variables taking values in D, sampled according to some distribution ν on the elements of D. For a set $A \subseteq D$, let $\nu(A)$ be the probability that a sample from ν belongs to the set A, and let

$$\nu_{X_1^k}(A) = \frac{1}{k} \sum_{j=1}^{k} \mathbb{1}_A(X_j),$$

where $\mathbb{1}_A$ is the indicator function for A. The function $\nu_{X_1^k}(A)$ is the *empirical average* of $\nu(A)$ on X_1^k.

Definition 2. Let $\mathcal{R}$ be a range set on a domain D and ν be a probability distribution on D. For $\varepsilon \in (0,1)$, an *ε-approximation to* $(\mathcal{R}, \nu)$ is a bag S of elements of D such that

$$\sup_{A \in \mathcal{R}} |\nu(A) - \nu_S(A)| \leq \varepsilon \ .$$

An ε-approximation can be constructed by sampling points of the domain according to the distribution ν, provided an upper bound to the VC-dimension of $\mathcal{R}$ or to its empirical VC-dimension is known:

Theorem 1 (Thm. 2.12 [40])**.** *Let $\mathcal{R}$ be a range set on a domain D with $\mathsf{VC}(\mathcal{R}) \leq d$, and let ν be a distribution on D. Given $\delta \in (0,1)$ and a positive integer ℓ, let*

$$\varepsilon = \sqrt{\frac{c}{\ell}\left(d + \log\frac{1}{\delta}\right)} \tag{1}$$

where c is an universal positive constant. Then, a bag of ℓ elements of D sampled independently *according to ν is an ε-approximation to $(\mathcal{R}, \nu)$ with probability at least $1 - \delta$.*

The constant c is approximately 0.5 [41].

A similar result holds when an upper bound to the empirical VC-Dimension is available [22].

Theorem 2 (Sect. 3 [22])**.** *Let $\mathcal{R}$ be a range set on a domain D, and let ν be a distribution on D. Let $X_1^\ell = (X_1, \ldots, X_\ell)$ be a collection of elements from D sampled independently according to ν. Let d be an integer such that $\mathsf{EVC}(\mathcal{R}, X_1^\ell) \leq d$. Given $\delta \in (0,1)$, let*

$$\varepsilon = 2\sqrt{\frac{2d\log(\ell+1)}{\ell}} + \sqrt{\frac{2\log\frac{2}{\delta}}{\ell}}. \tag{2}$$

Then, X_1^ℓ is a ε-approximation for $(\mathcal{R}, \nu)$ with probability at least $1 - \delta$.

4 The range set of a collection of itemsets

In this section we define the concept of a range set associated to a collection of itemsets and show how to bound the VC-dimension and the empirical VC-dimension of this range set. We use these definitions and results to develop our algorithm in later sections.

Definition 3. Given a collection $\mathcal{C}$ of itemsets built on a ground set $\mathcal{I}$, the *range set $\mathcal{R}(\mathcal{C})$ associated to $\mathcal{C}$ is a range set on* $2^{\mathcal{I}}$ containing the support sets of the itemsets in $\mathcal{C}$:

$$\mathcal{R}(\mathcal{C}) = \{T(A) \ : \ A \in \mathcal{C}\} \ .$$

The following Theoerem presents an upper bound to the empirical VC-dimension of $\mathcal{R}(\mathcal{C})$ on a dataset $\mathcal{D}$.

Theorem 3. *Let $\mathcal{C}$ be a collection of itemsets and let $\mathcal{D}$ be a dataset. Let d be the maximum integer for which there are at least d transactions $\tau_1, \ldots, \tau_d \in \mathcal{D}$ such that the set $\{\tau_1, \ldots, \tau_d\}$ is an* antichain[1]*, and each τ_i, $1 \leq i \leq d$, contains at least 2^{d-1} itemsets from $\mathcal{C}$. Then $\mathsf{EVC}(\mathcal{R}(\mathcal{C}), \mathcal{D}) \leq d$.*

Proof. The antichain requirement guarantees that the set of transactions considered in the computation of d could indeed theoretically be shattered. Assume that a subset $\mathcal{F}$ of $\mathcal{D}$ contains two transactions τ' and τ'' such that $\tau' \subseteq \tau''$. Any itemset from $\mathcal{C}$ appearing in τ' would also appear in τ'', so there would not be any itemset $A \in \mathcal{C}$ such that $\tau'' \in T(A) \cap F$ but $\tau' \notin T(A) \cap \mathcal{F}$, which would imply that $\mathcal{F}$ can not be shattered. Hence sets that are not antichains should not be considered. This has the net effect of potentially resulting in a lower d, i.e., in a stricter upper bound to $\mathsf{EVC}(\mathcal{R}(\mathcal{C}), \mathcal{D})$.

Let now $\ell > d$ and consider a set $\mathcal{L}$ of ℓ transactions from $\mathcal{D}$ that is an antichain. Assume that $\mathcal{L}$ is shattered by $\mathcal{R}(\mathcal{C})$. Let τ be a transaction in $\mathcal{L}$. The transactions τ belongs to $2^{\ell-1}$ subsets of L. Let $\mathcal{K} \subseteq \mathcal{L}$ be one of these subsets. Since $\mathcal{L}$ is shattered, there exists an itemset $A \in \mathcal{C}$ such that $T(A) \cap \mathcal{L} = \mathcal{K}$. From this and the fact that $t \in \mathcal{K}$, we have that $\tau \in T(A)$ or equivalently that $A \subseteq \tau$. Given that all the subsets $\mathcal{K} \subseteq \mathcal{L}$ containing τ are different, then also all the $T(A)$'s such that $T(A) \cap \mathcal{L} = \mathcal{K}$ should be different, which in turn implies that all the itemsets A should be different and that they should all appear in τ. There are $2^{\ell-1}$ subsets $\mathcal{K}$ of $\mathcal{L}$ containing τ, therefore τ must contain at least $2^{\ell-1}$ itemsets from $\mathcal{C}$, and this holds for all ℓ transactions in $\mathcal{L}$. This is a contradiction because $\ell > d$ and d is the maximum integer for which there are at least d transactions containing at least 2^{d-1} itemsets from $\mathcal{C}$. Hence $\mathcal{L}$ cannot be shattered and the thesis follows. □ □

4.1 Computing the VC-Dimension

The naïve computation of d according to the definition in Thm. 3 requires to scan the transactions one by one, compute the number of itemsets from $\mathcal{C}$ appearing in each transaction, and make sure to consider only itemsets constituting antichains. Given the very large number of transactions in typical dataset and the fact that the number of itemsets in a transaction is exponential in its length, this method would be computationally too expensive. An upper bound to d (and therefore to $\mathsf{EVC}(\mathcal{R}(\mathcal{C}), \mathcal{D})$) can be computed by solving a *Set-Union Knapsack Problem* (SUKP) [7] associated to $\mathcal{C}$.

Definition 4 ([7])**.** Let $U = \{a_1, \ldots, a_\ell\}$ be a set of elements and let $\mathcal{S} = \{A_1, \ldots, A_k\}$ be a set of subsets of U, i.e. $A_i \subseteq U$ for $1 \leq i \leq k$. Each subset A_i, $1 \leq i \leq k$, has an associated non-negative *profit* $\rho(A_i) \in \mathbb{R}^+$, and each element a_j, $1 \leq j \leq \ell$ as an associated non-negative weight $w(a_j) \in \mathbb{R}^+$. Given a subset $\mathcal{S}' \subseteq \mathcal{S}$, we define the profit of $\mathcal{S}'$ as $P(\mathcal{S}') = \sum_{A_i \in \mathcal{S}'} \rho(A_i)$. Let $U_{\mathcal{S}'} = \cup_{A_i \in \mathcal{S}'} A_i$. We define the weight of $\mathcal{S}'$ as $W(\mathcal{S}') = \sum_{a_j \in U_{\mathcal{S}'}} w(a_j)$. Given a non-negative parameter c that we call *capacity*, the *Set-Union Knapsack Problem* (SUKP) requires to find the set $\mathcal{S}^* \subseteq \mathcal{S}$ which *maximizes* $P(\mathcal{S}')$ over all sets $\mathcal{S}'$ such that $W(\mathcal{S}') \leq c$.

In our case, U is the set of items that appear in the itemsets of $\mathcal{C}$, $\mathcal{S} = \mathcal{C}$, the profits and the weights are all unitary, and the capacity constraint is an integer ℓ. We call this optimization problem the *SUKP associated to $\mathcal{C}$ with capacity ℓ*. It is easy to see that the optimal profit of this SUKP is the maximum number of itemsets from $\mathcal{C}$ that a transaction of length ℓ can contain. In order to show how to use this fact to compute an upper bound to $\mathsf{EVC}(\mathcal{R}(\mathcal{C}), \mathcal{D})$, we need to define some additional terminology. Let $\ell_1, \ldots, \ell_w$ be the sequence of the *transaction lengths* of $\mathcal{D}$, i.e., for each value ℓ for which there is at least a transaction in $\mathcal{D}$ of length ℓ, there is one (and only one) index i, $1 \leq i \leq w$ such that $\ell_i = \ell$. Assume that the ℓ_i's are labelled in sorted decreasing order: $\ell_1 > \ell_2 > \cdots > \ell_w$. Let now L_i, $1 \leq i \leq w$ be the maximum number of transactions in $\mathcal{D}$ that have length at least ℓ_i and such that for no two τ', τ'' of them we have either $\tau' \subseteq \tau''$ or $\tau'' \subseteq \tau'$. Let now q_i be the optimal profit of the SUKP associated to $\mathcal{C}$ with capacity L_i, and let $b_i = \lfloor \log_2 q_i \rfloor + 1$. The sequences $(\ell_i)_1^w$ and a sequence $(L_i^*)^w$ of upper bounds to $(L_i)_1^w$ can be computed efficiently with a scan of the dataset. The following lemma uses these sequences to show how to obtain an upper bound to the empirical VC-dimension of $\mathcal{C}$ on $\mathcal{D}$.

Lemma 1. *Let j be the minimum integer for which $b_i \leq L_i$. Then $\mathsf{EVC}(\mathcal{C}, \mathcal{D}) \leq b_j$.*

[1] An antichain is a collection of sets such no one of them is a subset of another.

Proof. If $b_j \leq L_j$, then there are at least b_j transactions which can contain 2^{b_j-1} itemsets from $\mathcal{C}$ and this is the maximum b_i for which it happens, because the sequence $b_1, b_2, \ldots, b_w$ is sorted in decreasing order, given that the sequence $q_1, q_2, \ldots, q_w$ is. Then b_j satisfies the conditions of Lemma 3. Hence $\mathsf{EVC}(\mathcal{C}, \mathcal{D}) \leq b_j$. □ □

Corollary 1. *Let q be profit of the SUKP associated to $\mathcal{C}$ with capacity equal to $\ell = |\{a \in \mathcal{I} : \exists A \in \mathcal{C}$ s.t. $a \in A\}|$ (ℓ is the number of items such that there is at least one itemset in $\mathcal{C}$ containing them). Let $b = \lfloor \log_2 q \rfloor + 1$. Then $\mathsf{VC}(\mathcal{R}(\mathcal{C})) \leq b$.*

Complexity and runtime considerations. Solving the SUKP optimally is NP-hard in the general case, although there are known restrictions for which it can be solved in polynomial time using dynamic programming [7]. Since we have unit weights and unit profits, our SUKP is equivalent to the *densest k-subhypergraph* problem, which can not be approximated within a factor of $2^{O(\log n)^\delta}$ for any $\delta > 0$ unless $3STA \in DTIME(2^{n^{3/4+\varepsilon}})$ [42]. A greedy algorithm by Arulselvan [43] allows a constant factor approximation if each items only appear in a constant fraction of itemsets of $\mathcal{C}$. For our case, it is actually *not necessary to compute the optimal solution* to the SUKP: any upper bound solution for which we can prove that there is no power of two between that solution and the optimal solution would result in the *same upper bound* to the (empirical) VC-dimension, while substantially speeding up the computation. This property can be specified in currently available optimization problem solvers (e.g., CPLEX), which can then can compute the bound to the (empirical) VC-dimension very fast even for very large instances with thousands of items and hundred of thousands of itemsets in $\mathcal{C}$, making this approach practical.

Refinements. It is possible to make some refinements to our computation of the *empirical* VC-dimension of a collection $\mathcal{C}$ of itemsets on a dataset $\mathcal{D}$. First of all, one can remove from $\mathcal{C}$ all itemsets that never appear in $\mathcal{D}$, as the corresponding ranges can not help shattering any set of transactions in $\mathcal{D}$. Identifying which itemsets to remove requires a single linear scan of $\mathcal{D}$. Secondly, when computing the capacities L_i (i.e., their upper bounds L_i^*), we can ignore all the transactions that do not contain *any* of the itemsets in $\mathcal{C}$ (or the filtered version of $\mathcal{C}$), as there is no way of shatter them using the ranges corresponding to itemsets in $\mathcal{C}$. Both refinements aim at reducing the optimal value of the SUKP associated to $\mathcal{C}$, and therefore at computing a smaller bound to the empirical VC-dimension of $\mathcal{C}$ on $\mathcal{D}$. We remark that these refinements can not be used when computing the (non-empirical) VC-dimension.

The range set of all itemsets. The range set associated to $2^{\mathcal{I}}$ is particularly interesting for us. It is possible to compute bounds to $\mathsf{VC}(\mathcal{R}(2^{\mathcal{I}}))$ and $\mathsf{EVC}(\mathcal{R}(2^{\mathcal{I}}), \mathcal{D})$ without having to solve a SUKP.

Theorem 4 ([31]). *Let $\mathcal{D}$ be a dataset built on a ground set $\mathcal{I}$. The* d-index $\mathsf{d}(\mathcal{D})$ *of $\mathcal{D}$ is the maximum integer d such that $\mathcal{D}$ contains at least d transactions of length at least d that form an antichain. We have $\mathsf{EVC}(\mathcal{R}(2^{\mathcal{I}}), \mathcal{D}) \leq \mathsf{d}(\mathcal{D})$.*

Corollary 2. $\mathsf{VC}(\mathcal{R}(2^{\mathcal{I}})) \leq |\mathcal{I}| - 1$.

Riondato and Upfal [31] presented an efficient algorithm to compute an upper bound to the d-index of a dataset with a single linear scan of the dataset $\mathcal{D}$. The upper bound presented in Thm. 4 is tight: there are datasets for which $\mathsf{EVC}(\mathcal{R}(2^{\mathcal{I}}), \mathcal{D}) = \mathsf{d}(\mathcal{D})$ [31]. This implies that the upper bound presented in Corol. 2 is also tight.

5 Finding the True Frequent Itemsets

In this section we present an algorithm that receives in input a dataset $\mathcal{D}$, a minimum frequency threshold θ, and a confidence parameter $\delta \in (0,1)$ and identifies a threshold $\hat{\theta}$ such that, with probability at least δ, all itemsets with frequency at least $\hat{\theta}$ in $\mathcal{D}$ are True Frequent Itemsets with respect to θ. The threshold $\hat{\theta}$ can be used to find a collection $\mathcal{C} = \mathsf{FI}(\mathcal{D}, \mathcal{I}, \hat{\theta})$ of itemsets such that $\Pr(\exists A \in \mathcal{C}$ s.t. $t_\pi(A) < \theta) < \delta$.

The intuition behind the method is the following. Let $\mathcal{B}$ be the *negative border* of $\mathsf{TFI}(\pi,\mathcal{I},\theta)$, that is the set of itemsets not in $\mathsf{TFI}(\pi,\mathcal{I},\theta)$ but such that all their proper subsets are in $\mathsf{TFI}(\pi,\mathcal{I},\theta)$. If we can find an ε such that $\mathcal{D}$ is an ε-approximation to $(\mathcal{R}(\mathcal{B}),\pi)$ then we have that any itemset $A\in\mathcal{B}$ has a frequency $f_\mathcal{D}(A)$ in $\mathcal{D}$ less than $\hat{\theta}=\theta+\varepsilon$, given that it must be $t_\pi(A)<\theta$. By the antimonotonicity property of the frequency, the same holds for all itemsets that are supersets of those in $\mathcal{B}$. Hence, the only itemsets that can have frequency in $\mathcal{D}$ greater or equal to $\hat{\theta}=\theta+\varepsilon$ are those with true frequency at least θ. In the following paragraphs we show how to compute ε.

Let δ_1 and δ_2 be such that $(1-\delta_1)(1-\delta_2)\geq 1-\delta$. Let $\mathcal{R}(2^\mathcal{I})$ be the range space of all itemsets. We use Corol. 2 (resp. Thm. 4) to compute an upper bound d' to $\mathsf{VC}(\mathcal{R}(2^\mathcal{I}))$ (resp. d'' to $\mathsf{EVC}(\mathcal{R}(2^\mathcal{I}),\mathcal{D})$). Then we can use d' in Thm. 1 (resp. d'' in Thm. 2) to compute an ε_1' (resp. an ε_1'') such that $\mathcal{D}$ is, with probability at least $1-\delta_1$, an ε_1'-approximation (resp. ε_1''-approximation) to $(\mathcal{R}(2^\mathcal{I}),\pi)$.

Fact 1. *Let $\varepsilon_1=\min\{\varepsilon_1',\varepsilon_1''\}$. With probability at least $1-\delta_1$, $\mathcal{D}$ is an ε_1-approximation to $(\mathcal{R}(2^\mathcal{I}),\pi)$.*

We want to find an upper bound the (empirical) VC-dimension of $\mathcal{R}(\mathcal{B})$. To this end, we use the fact that the negative border of a collection of itemsets is a *maximal antichain* on $2^\mathcal{I}$, that is, a collection of sets from $2^\mathcal{I}$ such that for no two of them, one of them is included in the other and such that is not a proper subset of any other antichain. Let now $\mathcal{W}$ be the *negative border* of $\mathcal{C}_1=\mathsf{FI}(\mathcal{D},\mathcal{I},\theta-\varepsilon_1)$, $\mathcal{G}=\{A\subseteq\mathcal{I}\ :\ \theta-\varepsilon_1\leq f_\mathcal{D}(A)<\theta+\varepsilon_1\}$, and $\mathcal{F}=\mathcal{G}\cup\mathcal{W}$.

Lemma 2. *Let $\mathcal{Y}$ be the set of maximal antichains in $\mathcal{F}$. If $\mathcal{D}$ is an ε_1-approximation to $(\mathcal{R}(2^\mathcal{I}),\pi)$, then*

1. $\max_{\mathcal{A}\in\mathcal{Y}}\mathsf{EVC}(\mathcal{R}(\mathcal{A}),\mathcal{D})\geq\mathsf{EVC}(\mathcal{R}(\mathcal{B}),\mathcal{D})$*, and*
2. $\max_{\mathcal{A}\in\mathcal{Y}}\mathsf{VC}(\mathcal{R}(\mathcal{A}))\geq\mathsf{VC}(\mathcal{R}(\mathcal{B}))$*.*

Proof. Given that $\mathcal{D}$ is an ε_1-approximation to $(\mathcal{R}(2^\mathcal{I}),\pi)$, then $\mathsf{TFI}(\pi,\mathcal{I},\theta)\subseteq\mathcal{G}\cup\mathcal{C}_1$. From this and the definition of negative border and of $\mathcal{F}$, we have that $\mathcal{B}\subseteq\mathcal{F}$. Since $\mathcal{B}$ is a maximal antichain, then $\mathcal{B}\in\mathcal{Y}$. Hence the thesis. □ □

In order to compute upper bounds to $\mathsf{VC}(\mathcal{R}(\mathcal{B}))$ and $\mathsf{EVC}(\mathcal{R}(\mathcal{B}),\mathcal{D})$ we can solve slightly modified SUKPs associated to $\mathcal{F}$ with the additional constraint that the optimal solution, which is a collection of itemsets, *must be a maximal antichain*. Lemma 1 still holds even for the solutions of these modified SUKPs. Using these bounds in Thms. 1 and 2, we compute an ε_2 such that, with probability at least $1-\delta_2$, $\mathcal{D}$ is an ε_2-approximation to $(\mathcal{R}(\mathcal{B}),\pi)$. Let $\hat{\theta}=\theta+\varepsilon_2$. The following Theorem shows that $\hat{\theta}$ has the desired properties.

Theorem 5. *With probability at least $1-\delta$, $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$ contains no false positives:*

$$\Pr\left(\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})\subseteq\mathsf{TFI}(\pi,\mathcal{I},\theta)\right)\geq 1-\delta\enspace.$$

Proof. Consider the two events E_1="$\mathcal{D}$ is an ε_1-approximation for $(\mathcal{R}(2^\mathcal{I}),\pi)$" and E_2="$\mathcal{D}$ is an ε_2-approximation for $(\mathcal{R}(\mathcal{B}),\pi)$". From the above discussion and the definition of δ_1 and δ_2 it follows that the event $\mathsf{E}=\mathsf{E}_1\cap\mathsf{E}_1$ occurs with probability at least $1-\delta$. Suppose from now on that indeed E occurs.

Since E_1 occurs, then Lemma 2 holds, and the bounds we compute by solving the modified SUKP problems are indeed bounds to $\mathsf{VC}(\mathcal{R}(\mathcal{B}))$ and $\mathsf{EVC}(\mathcal{R}(\mathcal{B},\mathcal{D}))$. Since E_2 also occurs, then for any $A\in\mathcal{B}$ we have $|t_\pi(A)-f_\mathcal{D}(A)|\leq\varepsilon_2$, but given that $t_\pi(A)<\theta$ because the elements of $\mathcal{B}$ are not TFIs, then we have $f_\mathcal{D}(A)<\theta+\varepsilon_2$. Because of the antimonotonicity property of the frequency and the definition of $\mathcal{B}$, this holds for any itemset that is not in $\mathsf{TFI}(\pi,\mathcal{I},\theta)$. Hence, the only itemsets that can have a frequency in $\mathcal{D}$ at least $\hat{\theta}=\theta+\varepsilon_2$ are the TFIs, so $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})\subseteq\mathsf{TFI}(\pi,\mathcal{I},\theta)$, which concludes our proof. □ □

The pseudocode of our method is presented in Alg. 1.

Exploiting additional knowledge about π. Our algorithm is completely *distribution-free*, i.e., it does not require any assumption about the unknown distribution π. On the other hand, when information about π is available, our method can exploit it to achieve better performances in terms of running time, practicality, and accuracy. For example, in most applications π will not generate any transaction longer than some known upper bound $\ell \ll |\mathcal{I}|$. Consider for example an online marketplace like Amazon: it is extremely unlikely (if not humanly impossible) that a single customer buys one of each available product. Indeed, given the hundred of thousands of items on sale, it is safe to assume that all the transactions will contains at most ℓ items, for some $\ell \ll |\mathcal{I}|$. Other times, like in an online survey, it is the nature of the process that limits the number of items in a transaction, in this case the number of questions. A different kind of information about the generative process may consists in knowing that some combination of items may never occur, because "forbidden" in some wide sense. Other examples are possible. All these pieces of information can be used to compute better (i.e., stricter) upper bounds to the VC-dimension $\mathsf{VC}(\mathcal{R}(2^{\mathcal{I}}))$. For example, if we know that π will never generate transactions with more than ℓ items, we can safely say that $\mathsf{VC}(\mathcal{R}(2^{\mathcal{I}})) \leq \ell$, a much stricter bound than $|\mathcal{I}| - 1$ from Corol. 2. This may result in a smaller ε_1, a smaller ε, and a smaller $\hat{\theta}$, which allows to produce more TFIs in the output collection. In the experimental evaluation, we show the positive impact of including additional information may on the performances of our algorithm.

Dataset	Freq. θ	TFIs	Times FPs	Times FNs
`accidents`	0.2	889883	100%	100%
`BMS-POS`	0.005	4240	100%	100%
`chess`	0.6	254944	100%	100%
`connect`	0.85	142127	100%	100%
`kosarak`	0.015	189	45%	55%
`pumsb*`	0.45	1913	5%	80%
`retail`	0.0075	277	10%	20%

Table 1: Fractions of times that $\mathsf{FI}(\mathcal{D}, \mathcal{I}, \theta)$ contained false positives and missed TFIs (false negatives) over 20 datasets from the same ground truth.

6 Experimental evaluation

We conducted an extensive evaluation to assess the performances of the algorithm we propose. In particular, we used it to compute values $\hat{\theta}$ for a number of frequencies θ on different datasets, and compared the collection of FIs w.r.t. $\hat{\theta}$ with the collection of TFIs, measuring the number of false positives and the fraction of TFIs that were found.

Implementation. We implemented the algorithm in Python 3.3. To mine the FIs, we used the C implementation by Grahne and Zhu [44]. Our solver of choice for the SUKPs was IBM® ILOG® CPLEX® Optimization Studio 12.3. We run the experiments on a number of machines with x86-64 processors running GNU/Linux 3.2.0.

Datasets generation. We evaluated the algorithm using pseudo-artificial datasets generated by taking the datasets from the FIMI'04 repository[2] as the *ground truth* for the true frequencies t_π of the itemsets. We considered the following datasets: `accidents`, `BMS-POS`, `chess`, `connect`, `kosarak`, `pumsb*`, and `retail`. These datasets differ in size, number of items, and, more importantly for our case, distribution of the frequencies of the itemsets [45]. We created a dataset by *sampling 20 million transactions uniformly at*

[2] `http://fimi.ua.ac.be/data/`

			Reported TFIs (Average Fraction)			
			“Vanilla” (no info)		Additional Info	
Dataset	Freq. θ	TFIs	CU Method	This Work	CU Method	This Work
accidents	0.8	149	0.838	**0.981**	0.853	**0.981**
	0.7	529	0.925	**0.985**	0.935	**0.985**
	0.6	2074	0.967	**0.992**	0.973	**0.992**
	0.5	8057	0.946	**0.991**	0.955	**0.991**
	0.45	16123	0.948	**0.992**	0.955	**0.992**
	0.4	32528	0.949	0.991	0.957	**0.992**
	0.3	149545			0.957	**0.989**
	0.2	889883			0.957	**0.987**
BMS-POS	0.05	59	0.845	**0.938**	0.851	**0.938**
	0.03	134	0.879	**0.992**	0.895	**0.992**
	0.02	308	0.847	**0.956**	0.876	**0.956**
	0.01	1099	0.813	0.868	0.833	**0.872**
	0.0075	1896			0.826	**0.854**
	0.005	4240			0.762	**0.775**
chess	0.8	8227	0.964	**0.991**	0.964	**0.991**
	0.775	13264	0.957	**0.990**	0.957	**0.990**
	0.75	20993	0.957	**0.983**	0.957	**0.983**
	0.65	111239			0.972	**0.991**
	0.6	254944			0.970	**0.989**
connect	0.95	2201	0.802	**0.951**	0.802	**0.951**
	0.925	9015	0.881	**0.975**	0.881	**0.975**
	0.9	27127	0.893	**0.978**	0.893	**0.978**
	0.875	65959			0.899	**0.974**
	0.85	142127			0.918	**0.974**
kosarak	0.04	42	0.738	**0.939**	0.809	**0.939**
	0.035	50	0.720	**0.980**	0.780	**0.980**
	0.025	82			0.682	**0.963**
	0.02	121			0.650	**0.975**
	0.015	189			0.641	**0.933**
pumsb*	0.55	305	0.791	**0.926**	0.859	**0.926**
	0.5	679	0.929	**0.998**	0.957	**0.998**
	0.49	804	0.858	**0.984**	0.907	**0.984**
	0.475	1050			0.942	**0.996**
	0.45	1913			0.861	**0.976**
retail	0.03	32	0.625	**1.00**	0.906	**1.00**
	0.025	38	0.842	**0.973**	0.972	**0.973**
	0.0225	46	0.739	0.934	0.869	**0.935**
	0.02	55			0.882	**0.945**
	0.01	159			0.902	**0.931**
	0.0075	277			0.811	**0.843**

Table 2: Recall. Average fraction (over 20 runs) of reported TFIs in the output of an algorithm using Chernoff and Union bound and of the one presented in this work. For each algorithm we present two versions, one (Vanilla) which uses no information about the generative process, and one (Add. Info) in which we assume the knowlegde that the process will not generate any transaction longer than twice the size of the longest transaction in the original FIMI dataset. In bold, the best result (highest reported fraction).

random from a FIMI repository dataset. In this way the the true frequency of an itemset is its frequency in the original FIMI dataset. Given that our method to find the TFIs is distribution-free, this is a valid procedure to establish a ground truth. We used these enlarged datasets in our experiments, and use the original name of the datasets in the FIMI repository to annotate the results for the datasets we generated.

False positives and false negatives in $\mathsf{FI}(\mathcal{D},\mathcal{I},\theta)$. In the first set of experiments we evaluated the performances, in terms of inclusion of false positives and false negatives in the output, of mining the dataset at frequency θ. Table 1 reports the fraction of times (over 20 datasets from the same ground truth) that the set $\mathsf{FI}(\mathcal{D},\mathcal{I},\theta)$ contained false positives (FP) and was missing TFIs (false negatives (FN)). In most cases, especially when there are many TFIs, the inclusion of false positives when mining at frequency θ should be expected. This highlights a need for methods like the one presented in this work, as there is no guarantee that $\mathsf{FI}(\mathcal{D},\mathcal{I},\theta)$ only contains TFIs. On the other hand, the fact that some TFIs have frequency in the dataset *smaller* than θ (false negatives) points out how one can not aim to extract all and only the TFIs by using a fixed threshold approach (as the one we present).

Control of the false positives (Precision). In this set of experiments we evaluated how well the threshold $\hat{\theta}$ computed by our algorithm allows to avoid the inclusion of false negatives in $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$. To this end, we used a wide range of values for the minimum true frequency threshold θ (see Table 2) and fixed $\delta = 0.1$. We repeated each experiment on 20 different enlarged datasets generated from the same original FIMI dataset. In all the *hundreds* of runs of our algorithms, $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$ *never* contained *any false positive*, i.e., *always contained only TFIs*. In other words, the *precision* of the output was 1.0 in all our experiments. Not only our method can give a frequency threshold to extract only TFIs, but it is more *conservative*, in terms of including false positives, than what the theoretical analysis guarantees.

Inclusion of TFIs (Recall). In addition to avoid false positives in the results, one wants to include as many TFIs as possible in the output collection. To this end, we assessed what fraction of the total number of TFIs is reported in $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$. Since there were no false positives, this is corresponds to evaluating the *recall* of the output collection. We fixed $\delta = 0.1$, and considered different values for the minimum true frequency threshold θ (see Table 2). For each frequency threshold, we repeated the experiment on 20 different datasets sampled from the same original FIMI dataset, and found very small variance in the results. We compared the fraction of TFIs that our algorithm included in output with that included by the "Chernoff and Union bounds" (CU) method we presented in Introduction. We compared two variants of the algorithms: one ("vanilla") which makes no assumption on the generative distribution π, and another ("additional info") which assumes that the process will not generate any transaction longer than twice the longest transaction found in the original FIMI dataset. Both algorithms can be easily modified to include this information. In Table 2 we report the average fraction of TFIs contained in $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$. We can see that the amount of TFIs found by our algorithm is always very high: only a *minimal* fraction (often less than 3%) of TFIs do not appear in the output. This is explained by the fact that the value ε_2 computed in our method (see Sect. 5) is always smaller than 10^{-4}. Moreover our solution *uniformly outperforms* the CU method, often by a huge margin, since our algorithm does not have to take into account all possible itemsets when computing $\hat{\theta}$. Only partial results are reported for the "vanilla" variant because of the very high number of items in the considered datasets: the mining of the dataset is performed at frequency threshold $\theta - \varepsilon_1$ and if there are many items, then the value of ε_1 becomes very high because the bound to the VC-dimension of $\mathcal{R}(2^{\mathcal{I}})$ is $|\mathcal{I}| - 1$, and as a consequence we have $\theta - \varepsilon_1 \leq 0$. We stress, though, that assuming no knowledge about the distribution π is not realistic, and usually additional information, especially regarding the length of the transactions, is available and can and should be used. The use of additional information gives flexibility to our method and improves its practicality. Moreover, in some cases, it allows to find an even larger fraction of the TFIs.

7 Conclusions

The usefulness of frequent itemset mining is often hindered by spurious discoveries, or false positives, in the results. In this work we developed an algorithm to compute a frequency threshold $\hat{\theta}$ such that the collection of FIs at frequency $\hat{\theta}$ is a good approximation the collection of True Frequent Itemsets. The threshold is such that that the probability of reporting *any* false positive is bounded by a user-specified quantity. We used concepts from statistical learning theory and from optimization to develop and analyze the algorithm. The experimental evaluation shows that the method we propose can indeed be used to control the presence of false positives while, at the same time, extracting a very large fraction of the TFIs from huge datasets. There are a number of directions for further research. Among these, we find particularly interesting and challenging the extension of our method to other definitions of statistical significance for patterns and to other definitions of patterns such as sequential patterns [46]. Also interesting is the derivation of better lower bounds to the VC-dimension of the range set of a collection of itemsets. Moreover, while this work focuses on itemsets mining, we believe that it can be extended and generalized to other settings of multiple hypothesis testing, and give another alternative to existing approaches for controlling the probability of false discoveries.

References

[1] Matteo Riondato and Fabio Vandin. Finding the true frequent itemsets. In *Proc. SIAM Int. Data Mining Conf.*, 2014.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22:207–216, June 1993. ISSN 0163-5808. doi: 10.1145/170036.170072.

[3] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques.* Morgan kaufmann, 2006.

[4] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowl. Disc.* , 15:55–86, 2007. ISSN 1384-5810. doi: 10.1007/s10618-006-0059-1.

[5] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[6] Adam Kirsch, Michael Mitzenmacher, Andrea Pietracaprina, Geppino Pucci, Eli Upfal, and Fabio Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. *J. ACM*, 59(3):12:1–12:22, June 2012. ISSN 0004-5411. doi: 10.1145/2220357.2220359.

[7] Olivier Goldschmidt, David Nehme, and Gang Yu. Note: On the set-union knapsack problem. *Naval Research Logistics*, 41(6):833–842, 1994.

[8] Toon Calders, Christophe Rigotti, and Jean-François Boulicaut. A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases*, volume 3848 of *Lecture Notes in Computer Science*, pages 64–80. Springer Berlin Heidelberg, 2006. doi: 10.1007/11615576_4.

[9] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), September 2006. ISSN 0360-0300. doi: 10.1145/1132960.1132963.

[10] Guimei Liu, Haojun Zhang, and Limsoon Wong. Controlling false positives in association rule mining. *Proc. VLDB Endow.*, 5(2):145–156, October 2011. ISSN 2150-8097.

[11] Craig Silverstein, Sergey Brin, and Rajeev Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining Knowl. Disc.*, 2(1):39–68, January 1998. ISSN 1384-5810. doi: 10.1023/A:1009713703947.

[12] Nimrod Megiddo and Ramakrishnan Srikant. Discovering predictive association rules. In Rakesh Agrawal, Paul E. Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proc. 4th Int. Conf. Knowl. Disc. Data Mining*, KDD '98, pages 274–278, New York, NY, USA, 1998. AAAI Press.

[13] William DuMouchel and Daryl Pregibon. Empirical Bayes screening for multi-item associations. In *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, KDD '01, pages 67–76, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502526.

[14] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. Assessing data mining results via swap randomization. *ACM Trans. Knowl. Disc. from Data*, 1(3), December 2007. ISSN 1556-4681. doi: 10.1145/1297332.1297338.

[15] Wilhelmiina Hämäläinen. StatApriori: an efficient algorithm for searching statistically significant association rules. *Knowl. Inf. Sys.*, 23(3):373–399, 2010. doi: 10.1007/s10115-009-0229-8.

[16] Sami Hanhijärvi. Multiple hypothesis testing in pattern discovery. In *Discovery Science*, volume 6926 of *Lecture Notes in Computer Science*, pages 122–134. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-24477-3_12.

[17] Geoffrey I. Webb. Discovering significant patterns. *Mach. Learn.*, 68(1):1–33, 2007. doi: 10.1007/s10994-007-5006-x.

[18] Geoffrey I. Webb. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. *Mach. Learn.*, 71:307–323, 2008. doi: 10.1007/s10994-008-5046-x.

[19] Vladimir N. Vapnik and Alexey J. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025.

[20] David Haussler and Emo Welzl. Epsilon-nets and simplex range queries. In *Proc. 2nd Annu. Symp. Computational geometry*, SCG '86, pages 61–71, New York, NY, USA, 1986. ACM. ISBN 0-89791-194-6. doi: 10.1145/10515.10522.

[21] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36:929–965, October 1989. ISSN 0004-5411. doi: 10.1145/76359.76371.

[22] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification : A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[23] Bernard Chazelle. *The discrepancy method: randomness and complexity.* Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-00357-1.

[24] Jiří Matoušek. *Lectures on Discrete Geometry.* Springer-Verlag, Secaucus, NJ, USA, 2002. ISBN 0387953744.

[25] Martin Anthony and Peter L. Bartlett. *Neural Network Learning - Theoretical Foundations.* Cambridge University Press, New York, NY, USA, 1999. ISBN 978-0-521-57353-5.

[26] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Applications of Mathematics. Springer Berlin Heidelberg, 1996. ISBN 9780387946184.

[27] Michael Benedikt and Leonid Libkin. Aggregate operators in constraint query languages. *J. Comput. Syst. Sci.*, 64(3):628–654, 2002. ISSN 0022-0000. doi: 10.1006/jcss.2001.1810.

[28] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Proc. 40th Annu. ACM Symp. Theory of Computing*, STOC '08, pages 609–618, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-047-0. doi: 10.1145/1374376.1374464.

[29] David Gross-Amblard. Query-preserving watermarking of relational databases and xml documents. *ACM Trans. Database Syst.*, 36:3:1–3:24, March 2011. ISSN 0362-5915. doi: 10.1145/1929934.1929937.

[30] Matteo Riondato, Mert Akdere, Uğur Çetintemel, Stanley B. Zdonik, and Eli Upfal. The VC-dimension of SQL queries and selectivity estimation through sampling. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Mach. Learn. Knowl. Disc. Databases - European Conf., ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proc., Part II*, volume 6912 of *Lecture Notes in Computer Science*, pages 661–676, Berlin / Heidelberg, 2011. Springer. ISBN 978-3-642-23782-9.

[31] Matteo Riondato and Eli Upfal. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Mach. Learn. Knowl. Disc. Databases - European Conf., ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proc., Part I*, volume 7523 of *Lecture Notes in Computer Science*, pages 25–41, Berlin / Heidelberg, 2012. Springer. ISBN 978-3-642-33459-7.

[32] Jon M. Kleinberg. Detecting a network failure. *Internet Mathematics*, 1(1):37–55, 2003. doi: 10.1080/15427951.2004.10129077.

[33] Jon M. Kleinberg, Mark Sandler, and Aleksandrs Slivkins. Network failure detection and graph connectivity. *SIAM J. Comput.*, 38(4):1330–1346, 2008. doi: 10.1137/070697793.

[34] Uriel Feige and Mohammad Mahdian. Finding small balanced separators. In *Proc. 38th ann. ACM Symp. Theory of Computing*, STOC '06, pages 375–384, New York, NY, USA, 2006. ACM. ISBN 1-59593-134-1. doi: 10.1145/1132516.1132573.

[35] Sorabh Gandhi, Subhash Suri, and Emo Welzl. Catching elephants with mice: Sparse sampling for monitoring sensor networks. *ACM Trans. Sensor Netw.*, 6(1):1:1–1:27, January 2010. ISSN 1550-4859. doi: 10.1145/1653760.1653761.

[36] Ittai Abraham, Daniel Delling, Amos Fiat, Andrew V. Goldberg, and Renato F. Werneck. VC-dimension and shortest path algorithms. In *Automata, Languages and Programming*, volume 6755 of *Lecture Notes in Computer Science*, pages 690–699. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-22006-7_58.

[37] Matteo Riondato and Evgenios M. Kornaropoulos. Fast approximation of betweenness centrality through sampling. In Ben Carterette, Fernando Diaz, Carlos Castillo, and Donald Metzler, editors, *WSDM*, pages 413–422. ACM, 2014. ISBN 978-1-4503-2351-2.

[38] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. The MIT Press, 2012.

[39] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Statistics for engineering and information science. Springer-Verlag, New York, NY, USA, 1999. ISBN 9780387987804.

[40] Sariel Har-Peled and Micha Sharir. Relative (p, ε)-approximations in geometry. *Discrete & Computational Geometry*, 45(3):462–496, 2011. ISSN 0179-5376. doi: 10.1007/s00454-010-9248-1.

[41] Maarten Löffler and Jeff M. Phillips. Shape fitting on point sets with probability distributions. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009*, volume 5757 of *Lecture Notes in Computer Science*, pages 313–324. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-04128-0_29.

[42] M. T. Hajiaghayi, K. Jain, K. Konwar, L. C. Lau, I. I. Măndoiu, A. Russell, A. Shvartsman, and V. V. Vazirani. The minimum k-colored subgraph problem in haplotyping and dna primer selection. In *Proc. Int. Workshop on Bioniformatics Research and Applications (IWBRA)*, 2006.

[43] Ashwin Arulselvan. A note on the set union knapsack problem. *Discrete Applied Mathematics*, 169(0): 214 – 218, 2014. ISSN 0166-218X. doi: 10.1016/j.dam.2013.12.015.

[44] Gösta Grahne and Jianfei Zhu. Efficiently using prefix-trees in mining frequent itemsets. In Bart Goethals and Mohammed Javeed Zaki, editors, *FIMI*, volume 90 of *CEUR Workshop Proc.* CEUR-WS.org, 2003.

[45] Bart Goethals and Mohammed J. Zaki. Advances in frequent itemset mining implementations: report on FIMI'03. *SIGKDD Explor. Newsl.*, 6(1):109–117, June 2004. ISSN 1931-0145. doi: 10.1145/1007730.1007744.

[46] Cécile Low-Kam, Chedy Raïssi, Mehdi Kaytoue, Jian Pei, et al. Mining statistically significant sequential patterns. In *IEEE Int. Conf. Data Mining*, ICDM '13, 2013.

Algorithm 1: Compute freq. threshold $\hat{\theta}$ s. t. $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$ contains only TFIs with prob. at least $1-\delta$.

```
Input   : Dataset $\mathcal{D}$, freq. threshold $\theta \in (0,1)$, confidence $\delta \in (0,1)$
Output: Freq. threshold $\hat{\theta}$ s. t. $\mathsf{FI}(\mathcal{D},\mathcal{I},\hat{\theta})$ contains only TFIs with prob. at least $1-\delta$.
$\delta_1, \delta_2 \leftarrow 1-\sqrt{1-\delta}$   // $\delta_1$ and $\delta_2$ do not need to have the same value
$d'_1 \leftarrow$ upper bound to $\mathsf{VC}(\mathcal{R}(2^{\mathcal{I}}))$ (e.g., $|\mathcal{I}|-1$)
$\varepsilon'_1 \leftarrow \sqrt{\frac{c}{|\mathcal{D}|}\left(d'_1+\log\frac{1}{\delta_1}\right)}$
$d''_1 \leftarrow$ upper bound to $\mathsf{EVC}(\mathcal{R}(2^{\mathcal{I}}),\mathcal{D})$   // i.e., d-index of $\mathcal{D}$ [31]
$\varepsilon''_1 \leftarrow 2\sqrt{\frac{2d''_1\log(|\mathcal{D}|+1)}{|\mathcal{D}|}}+\sqrt{\frac{2\log\frac{2}{\delta}}{|\mathcal{D}|}}$
$\varepsilon_1 \leftarrow \min\{\varepsilon'_1, \varepsilon''_1\}$
$\mathcal{C}_1 = \mathsf{FI}(\mathcal{D},\mathcal{I},\theta-\varepsilon_1)$
$\mathcal{G} = \{A \subseteq \mathcal{I} \ : \ \theta-\varepsilon_1 \leq f_{\mathcal{D}}(A) < \theta+\varepsilon_1\}$
$\mathcal{W} \leftarrow$ negative border of $\mathcal{C}_1$
$\mathcal{F} = \mathcal{G} \cup \mathcal{W}$
$U \leftarrow \{a \in \mathcal{I} : \exists A \in \mathcal{F}$ s.t. $a \in A\}$
$b'_2 \leftarrow$ solveAntichainSUKP($U, \mathcal{F}, |U|$)
$d'_2 \leftarrow \lfloor \log_2 b'_2 \rfloor + 1$
$\varepsilon'_2 \leftarrow \sqrt{\frac{c}{|\mathcal{D}|}\left(d'_2+\log\frac{2}{\delta_2}\right)}$
$\ell_1, \ldots, \ell_w \leftarrow$ transaction lengths of $\mathcal{D}$   // see Sect. 4.1
$L_1, \ldots, L_w \leftarrow$ the maximum number of transactions in $\mathcal{D}$ that have length at least $\ell_i$, $1 \leq i \leq w$, and
  such that for no two $\tau'$, $\tau''$ of them we have either $\tau' \subseteq \tau''$ or $\tau'' \subseteq \tau'$
$i \leftarrow 0$
while True do
    $b''_2 \leftarrow$ solveAntichainSUKP($U, \mathcal{F}, \ell_i$)
    $d''_2 \leftarrow \lfloor \log_2 b''_2 \rfloor + 1$
    if $d''_2 < L_i$ then
        break
    else
        $i \leftarrow i+1$
    end
end
$\varepsilon''_2 \leftarrow 2\sqrt{\frac{2d''_2\log(|\mathcal{D}|+1)}{|\mathcal{D}|}}+\sqrt{\frac{2\log\frac{2}{\delta}}{|\mathcal{D}|}}$
$\varepsilon_2 \leftarrow \min\{\varepsilon'_2, \varepsilon''_2\}$
return $\theta+\varepsilon_2$
```