Finding the True Frequent Itemsets

Matteo Riondato and Fabio Vandin
Department of Computer Science, Brown University, Providence, RI, USA
{matteo,vandinfa}@cs.brown.edu

Thursday 12th December, 2013

Abstract

Frequent Itemsets (FIs) mining is a fundamental primitive in data mining that requires to identify all itemsets appearing in a fraction at least \( \theta \) of a transactional dataset \( D \). Often though, the ultimate goal of mining \( D \) is not an analysis of the dataset per se, but the understanding of the underlying process that generated \( D \). Specifically, in many applications \( D \) is a collection of samples obtained from an unknown probability distribution \( \pi \) on transactions, and by extracting the FIs in the dataset \( D \) one attempts to infer itemsets that are frequently generated by \( \pi \), which we call the True Frequent Itemsets (TFIs). Due to the inherently random nature of the generative process, the set of FIs is only a rough approximation to the set of TFIs, as it often contains a huge number of spurious itemsets, i.e., itemsets that are not among the TFIs. In this work we present two methods to identify a collection of itemsets that contains only TFIs with probability at least \( 1 - \delta \) (i.e., the methods have Family-Wise Error Rate bounded by \( \delta \)), for some user-specified \( \delta \), without imposing any restriction on \( \pi \). Our methods are distribution-free and make use of results from statistical learning theory involving the (empirical) VC-dimension of the problem at hand. This allows us to identify a larger fraction of the TFIs (i.e., to achieve higher statistical power) than what could be done using traditional multiple hypothesis testing corrections. In the experimental evaluation we compare our methods to established techniques (Bonferroni correction, holdout) and show that they return a very large subset of the TFIs, achieving a very high statistical power, while controlling the Family-Wise Error Rate.

1 Introduction

The extraction of association rules is one of the fundamental primitives in data mining and knowledge discovery from large databases [1]. In its most general definition, the problem can be reduced to identifying frequent sets of items, or frequent itemsets, appearing in a fraction at least \( \theta \) of all transactions in a dataset, where \( \theta \) 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 other data analysis and mining task, including clustering, classification, and indexing [15][16].

In most applications, one is not interested in mining a dataset to extract the frequent itemsets per se, but the mining process is used to infer properties of the underlying process that generated the dataset. For example, in market basket analysis the dataset is analyzed with the intent to understand the purchase behavior of customers, assuming that the purchase behavior of customers that generated the current dataset is the same that will be followed in the future. As an example, if one analyzes the transactional dataset of purchases to identify itemsets frequently sold on a specific day of the week (e.g., Monday), she will use these patterns to infer the itemsets that will be frequently sold on the same day of the following weeks. Analogously, in online recommendation systems the itemsets describing the frequent purchases of current customers are used to infer the frequent purchases of the future customers.

A natural and general model for these settings is to assume that the transactions in the dataset \( D \) are independent identically distributed (i.i.d) samples from a (unknown) probability distribution \( \pi \) on all the possible transactions on the items of \( D \), and each itemset has a fixed probability to appear in a random transaction from \( \pi \). The goal of the mining process is then to identify itemsets that have probability at least \( \theta \) to appear in a random transaction drawn from \( \pi \), given that for an itemset such probability corresponds to the fraction of transactions, among an infinite number of transactions, that contain the itemset. Since \( D \) represents only a finite sample from \( \pi \), the frequent itemsets of \( D \)

*This work is supported by NSF Grant BIGDATA 1247581.
only provide an approximation of such itemsets, and due to the random nature of the generative process a number of spurious, or false, discoveries, i.e., itemsets that appear among the frequent itemsets of \( D \) but are not generated with probability at least \( \theta \), may be reported. Given that \( \pi \) is not known, on one hand one can not aim at identifying all and only the itemsets having probability at least \( \theta \). On the other hand, using the frequent itemsets of \( D \) as a proxy for such itemsets does not provide any guarantee on the number of false discoveries that are reported. The problem of identifying the itemsets that appear with probability at least \( \theta \) with guarantees on the quality of the returned set has received scant attention in the literature.

1.1 Our contributions

In this paper we address the problem of identifying the True Frequent Itemsets (TFIs), that is the itemsets that appear with probability at least \( \theta \), while providing rigorous probabilistic guarantees on the number of false discoveries, and without making any assumption on the generative model of the transactions, i.e., on \( \pi \). This makes the methods we introduce completely distribution free. We develop our methods within the statistical hypothesis testing framework: each itemset has an associated null hypothesis claiming that the itemset has probability less than \( \theta \). This hypothesis is tested using information obtained from the dataset, and accepted or rejected accordingly. If the hypothesis is rejected, the itemset is included in the output collection. We focus on returning a set of TFIs with bounded Family-Wise Error Rate (FWER), that is we bound the probability that one or more false discovery (an itemset that have probability less than \( \theta \)) is reported among the TFIs. In particular, the methods have an FWER guaranteed to be within the user-specified limits.

We use results from statistical learning theory to develop and analyze our methods. We define a range set associated to a collection of itemsets and give an upper bound to its (empirical) VC-dimension, showing an interesting connection with a variant of the knapsack optimization problem. This generalizes results from Riondato and Upfal [23]. 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 methods and evaluate their performances in terms of actually controlling the FWER and achieving high statistical power. We notice that the methods are even better at controlling the FWER than what the theory guarantees, and they offer a very high power: only a small fraction of the TFIs is not included in the output collection. We also compare our methods with currently available techniques and find them competitive or superior.

We stress that we do not impose any restriction on the generative model of the transactions that are observed in the transactional dataset. In fact, we only assume that the transactions are independent samples from the distribution \( \pi \), without any constraint on the properties of \( \pi \). This is in contrast with the assumptions that are made by methods that perform statistical testing after the frequent patterns have been identified. In fact, these methods require a well specified, limited generative model to characterize the significance of a pattern.

Outline. The article is organized as follows. In Sect. 1.2 we review relevant previous contributions. Sections 2 and 3 contain preliminaries to formally define the problem and key concepts that we will use throughout the work. Our methods are described and analyzed in Sect. 4. We present the methodology and results of our experimental evaluation in Sect. 5. Conclusions and future work are presented in Sect. 6.

1.2 Previous work

While the problem of identifying the TFIs has received scant attention in the literature, a number of approaches have been proposed to filter the FIs of spurious patterns, i.e., patterns that are not actually interesting, according to some interestingness measure. We refer the reader to [15, Sect. 3] and [8] for surveys on different measures. We remark that, as noted by Liu et al. [20], the use of the minimum support threshold \( \theta \), 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.” 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 actual support of a pattern in the dataset and its expected support in a random dataset generated according to a statistical model 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 statistical model is a simple independence model: an item belongs to a transaction independently from other items \[3, 7, 10, 14, 19, 22, 25\]. In contrast, our work does not assume any statistical model for data generation, or better, does not impose any restriction on the model, with the result that our method is as general as possible, being distribution-free.

Kirsch et al. \[19\] 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 is an input parameter fixed by the user, and we return a collection of itemsets such that they all have a support at least as high as the threshold with respect to the distribution that generates the sample data.

Bolton et al. \[3\] suggest that, in pattern extraction settings, it may be more relevant to bound the False Discovery Rate rather than the Family-Wise Error Rate, due to the high number of statistical tests involved. In our experimental evaluation we noticed that the high number of tests is a problem when using traditional multiple-hypothesis correction techniques. The methods we present do not incur in this issue because they consider all the itemsets together, without the need to test each of them singularly.

Gionis et al. \[10\] 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 a procedure is not available. Considering the same generative model, Hanhijärvi \[17\] presents a direct adjustment method to bound the FWER while taking into consideration the actual number of hypotheses to be tested.

Webb \[28\] 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 FWER is presented by Webb \[29\] and works by exploiting the itemset lattice. The first method we present is inspired by the holdout technique, while the second can be used also when the dataset can not be split.

Liu et al. \[20\] 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 do not employ an explicit direct correction for multiple hypothesis testing, and use all data to obtain more accurate results, without the need to resampling it to create random datasets.

Teytaud and Lallich \[26\] suggest the use of VC-dimension to bound the risk of accepting spurious rules extracted from the database. Although referring to them as “association rules”, the rules they focus on involve ranges over domains and conjuctions of Boolean formulas to express subsets of interest. This is different than the transactional market basket analysis setting in our work.

2 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.

2.1 Itemsets mining

Given a ground set \( \mathcal{I} \) of items, let \( \pi \) be a probability distribution on \( 2^\mathcal{I} \). A transaction is a single sample drawn from \( \pi \). The length \( |\tau| \) of a transaction \( \tau \) is the number of items in \( \tau \). 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 \( \pi \). We call a subset of \( \mathcal{I} \) an itemset. For any itemset \( A \), let \( \Gamma(A) = \{ \tau \subseteq \mathcal{I} : A \subseteq \tau \} \) be the support set of \( A \). We define the true frequency \( t_\pi(A) \) of \( A \) with respect to \( \pi \) as the probability that a transaction sampled from \( \pi \) contains \( A \):

\[
t_\pi(A) = \sum_{\tau \in \Gamma(A)} \pi(\tau) .
\]

Analogously, given a dataset \( \mathcal{D} \), let \( T_D(A) \) 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 \( A \): \( f_D(A) = |T_D(A)|/|\mathcal{D}| \). It is easy to see that \( f_D(A) \) is the empirical average and an unbiased estimator for \( t_\pi(A) \): \( E[f_D(A)] = t_\pi(A) \).

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

\[ \text{FI}(\mathcal{D}, \mathcal{I}, \theta) = \{ A \subseteq \mathcal{I} : f_D(A) \geq \theta \} . \]

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 true frequency \( t_\pi \), which is unknown and only approximately reflected in the dataset \( \mathcal{D} \). Therefore, we are interested in finding the itemsets with true frequency \( t_\pi \) at least \( \theta \) for some \( \theta \in (0, 1] \). We call these itemsets the True Frequent Itemsets (TFIs) and denote the set they form as

\[ \text{TFI}(\pi, \mathcal{I}, \theta) = \{ A \subseteq \mathcal{I} : t_\pi(A) \geq \theta \} . \]

If one is only given a finite number of random samples (the dataset \( \mathcal{D} \)) from \( \pi \) as it is usually the case, one can not aim at finding the exact set \( \text{TFI}(\pi, \mathcal{I}, \theta) \): no assumption can be made on the set-inclusion relationship between \( \text{TFI}(\pi, \mathcal{I}, \theta) \) and \( \text{FI}(\mathcal{D}, \mathcal{I}, \theta) \), because an itemset \( A \in \text{TFI}(\pi, \mathcal{I}, \theta) \) may not appear in \( \text{FI}(\mathcal{D}, \mathcal{I}, \theta) \), and vice versa. One can instead try to approximate the set of TFIs. Specifically, given an user-specified parameter \( \delta \in (0, 1) \), we aim at approximating \( \text{TFI}(\pi, \mathcal{I}, \theta) \) with a collection \( \mathcal{C} \) of itemsets that, with probability at least \( 1 - \delta \), does not contain any spurious itemset:

\[ \Pr(\exists A \in \mathcal{C} : t_\pi(A) < \theta) < \delta . \]

At the same time, we want to maximize the size of \( \mathcal{C} \), i.e., finding as many TFIs as possible. In this work we present methods to identify a large number of the TFIs. These methods do not assume any limitation on \( \pi \) but use information from \( \mathcal{D} \), and guarantee a small probability of false positives while achieving a high success rate.

### 2.2 Vapnik-Chervonenkis dimension

The Vapnik-Chernovenkis (VC) dimension of a class of subsets defined on a set of points is a measure of the complexity or expressiveness of such class \[ (27) \]. A finite bound on the VC-dimension of a structure implies a bound on the number of random samples required to approximate the expectation of each indicator function associated to a set with its empirical average. We outline here some basic definitions and results and refer the reader to the works of Alon and Spencer \[ (2) \ Sect. 14.4 \] and Boucheron et al. \[ (4) \ Sect. 3 \] for an introduction of VC-dimension and a survey of recent developments.

Let \( \mathcal{D} \) be a domain and \( \mathcal{R} \) be a collection of subsets from \( \mathcal{D} \). We call \( \mathcal{R} \) a range set on \( \mathcal{D} \). Given \( B \subseteq \mathcal{D} \), 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 \( B \subseteq \mathcal{D} \), the empirical Vapnik-Chervonenkis (VC) dimension of \( \mathcal{R} \) on \( B \), denoted as \( \text{EVC}(\mathcal{R}, B) \) is the cardinality of the largest subset of \( B \) that is shattered by \( \mathcal{R} \). The VC-dimension of \( \mathcal{R} \) is defined as \( \text{VC}(\mathcal{R}) = \text{EVC}(\mathcal{R}, \mathcal{D}) \).

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 \( X^1 = (X_1, \ldots, X_k) \) be a collection of independent identically distributed random variables taking values in \( \mathcal{D} \),
sampled according to some distribution $\nu$ on the elements of $D$. For a set $A \subseteq D$, let $\nu(A)$ be the probability that a sample from $\nu$ belongs to the set $A$, and let

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

where $I_A$ is the indicator function for the set $A$. The function $\nu_{X_1}(A)$ is the empirical average of $\nu(A)$ on $X_1$.

**Definition 2.** Let $\mathcal{R}$ be a range set on $D$ and $\nu$ be a probability distribution on $D$. For $\varepsilon \in (0, 1)$, an $\varepsilon$-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 $\varepsilon$-approximation can be constructed by sampling points of the domain according to the distribution $\nu$, provided an upper bound to the VC-dimension of $\mathcal{R}$ or to its empirical VC-dimension is known:

**Theorem 1** (Thm. 2.12 [13]). Let $\mathcal{R}$ be a range set on $D$ with $\text{VC}(\mathcal{R}) \leq d$, and let $\nu$ be a distribution on $D$. Given $\delta \in (0, 1)$ and a positive integer $\ell$, let

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

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

Löffler and Phillips [21] showed experimentally that the constant $c$ is approximately 0.5. We used this value in our experiments.

**Theorem 2** (Sect. 3 [4]). Let $\mathcal{R}$ be a range set on $D$, and let $\nu$ 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 $\nu$. Let $d$ be an integer such that $\text{EVC}(\mathcal{R}, X_1^\ell) \leq d$. Given $\delta \in (0, 1)$, let

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

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

### 2.3 Statistical hypothesis testing

We develop our methods to identify the True Frequent Itemsets within the framework of statistical hypothesis testing. In statistical hypothesis testing, one uses some data $Y$ to evaluate a null hypothesis $H$, whose rejection corresponds to claiming the identification of a significant phenomenon. A test statistic $t_H(Y)$ associated to the hypothesis is computed from the data. If $t_H(Y)$ belongs to a predefined acceptance region (a subset of the domain of $t_H$), then $H$ is accepted, otherwise it is rejected. The acceptance region is defined a priori in such a way that the probability of a false positive, i.e., the probability of accepting a true null hypothesis (corresponding to a non significant phenomenon), is at most some critical value $\alpha$. Accepting a true null hypothesis is also called a “Type-1 error”. Often, defining an acceptance region as function of $\alpha$ is done implicitly, and instead of verifying whether the statistics belongs to it, one evaluates the associated $p$-value, viz. the probability that the statistic $t_H$ is at least as extreme as the value $t_H(Y)$ observed on the given data, conditioning on the null hypothesis being true, and reject $H$ if the $p$-value is not larger than $\alpha$. Another important factor for a statistical test is its power, that is the probability that the test correctly rejects the null hypothesis when the null hypothesis is false (also defined as $1 - \Pr$(Type-2 error), where a “Type-2 error” consists in non rejecting a false null hypothesis).

In our scenario, the naïve method to employ statistical hypothesis testing to find the TFIs is to define a null hypothesis $H_A$ for every itemset $A \subseteq I$, with $H_A = \text{“}t_\alpha(A) < \theta\text{“}$, and to compute the $p$-value for such null hypothesis using $f_D(A)$ as statistic. In particular, the $p$-value is given by the probability that the frequency of $A$ in a random
dataset (with the same number of transactions of $\mathcal{D}$) sampled from $\pi$ is $\geq f_{\mathcal{D}}(A)$ conditioning on the event $"t_{\pi}(A) < \theta"$. This is easy to compute given that the number of transactions in a dataset that contain $A$ has a Binomial distribution whose parameters are the size of the dataset and the true frequency of $A$ (Binomial test). If the $p$-value is “small enough”, the null hypothesis is rejected, that is $A$ is flagged as a TFI. The issue, in our case, is that we are considering a number of itemsets, that is we are facing a multiple hypothesis testing problem \[20\]. In this case, one is interested in controlling the probability of false discoveries among all the hypotheses tested, i.e., in controlling the Family-Wise Error Rate.

**Definition 3.** The Family-Wise Error Rate (FWER) of a statistical test is the probability of reporting at least one false discovery.

In order to achieve the desired FWER, one must define a sequence of critical values for the individual hypotheses, that is, implicitly define a sequence of acceptance region for the test statistics. The Bonferroni correction \[6\] is a widely-employed method to define such a sequence. In its simplest form, the Bonferroni correction suggests to compare the $p$-value of each hypothesis to the critical value $\alpha/m$, where $\alpha$ is the desired FWER, and $m$ is the number of hypotheses to be tested. The Bonferroni correction is not a good choice to identify TFIs, since the statistical power of methods that use the Bonferroni correction decreases with the number of hypotheses tested. In practical cases there would be hundreds or thousands of TFIs, making the use of the Bonferroni correction impractical if one wants to achieve high statistical power.

### 3 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 such range sets. We use these tools in the methods presented in later sections.

**Definition 4.** 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 space on $2^\mathcal{I}$ containing the support sets of the itemsets in $\mathcal{C}$: $\mathcal{R}(\mathcal{C}) = \{T(A) : A \in \mathcal{C}\}$.

**Lemma 1.** 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, and each $\tau_i$, $1 \leq i \leq d$ contains at least $2^{d-1}$ itemsets from $\mathcal{C}$. Then $\text{EVC}(\mathcal{R}(\mathcal{C}), \mathcal{D}) \leq d$.

**Proof.** The first 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 $\tau'$ and $\tau''$ such that $\tau' \subseteq \tau''$. Any itemset from $\mathcal{C}$ appearing in $\tau'$ would also appear in $\tau''$, so there would not be any itemset $A \in \mathcal{C}$ such that $\tau'' \in T(A) \cap \mathcal{F}$ 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 $\text{EVC}(\mathcal{R}(\mathcal{C}), \mathcal{D})$.

Let now $\ell > d$ and consider a set $\mathcal{L}$ of $\ell$ transactions from $\mathcal{D}$ that is an antichain. Assume that $\mathcal{L}$ is shattered by $\mathcal{R}(\mathcal{C})$. Let $\tau$ be a transaction in $\mathcal{L}$. The transactions $\tau$ belongs to $2^{\ell-1}$ subsets of $\mathcal{L}$. 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 $\tau$ 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 $\tau$. There are $2^{\ell-1}$ subsets $\mathcal{K}$ of $\mathcal{L}$ containing $\tau$, therefore $\tau$ must contain at least $2^{\ell-1}$ itemsets from $\mathcal{C}$, and this holds for all $\ell$ 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. $\square$

The exact computation of $d$ as defined in Lemma \[1\] could be extremely expensive since it 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 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 \(d\) (hence to \(\text{EVC}(\mathcal{R}(\mathcal{C}), D)\)) can be computed more efficiently by solving a Set-Union Knapsack Problem (SUKP) \[12\] associated to \(\mathcal{C}\).

**Definition 5** \([12]\). 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 associate non-negative weight \(w(a_j) \in \mathbb{R}^+\). Given a subset \(S' \subseteq \mathcal{S}\), we define the profit of \(S'\) as \(P(S') = \sum_{A_i \in S'} \rho(A_i)\). We define the weight of \(S'\) as \(W(S') = \sum_{a_j \in U_{\mathcal{S}}} w(a_j)\). Given a non-negative parameter \(c\) that we call capacity, the Set-Union Knapsack Problem (SUKP) requires to find the set \(S^* \subseteq \mathcal{S}\) which maximizes \(P(S')\) over all sets \(S'\) for which \(W(S') \leq c\).

In our case, \(U\) is the set of items that appear in the itemsets of \(\mathcal{C}\), \(S = \mathcal{C}\), the profits and the weights are all unitary, and the capacity constraint is an integer \(\ell\). We call this optimization problem the SUKP associated to \(\mathcal{C}\) with capacity \(\ell\).

It should be evident that the optimal profit of this SUKP is the maximum number of itemsets from \(\mathcal{C}\) that a transaction of length \(\ell\) can contain. In order to show how to use this fact to compute an upper bound to \(\text{EVC}(\mathcal{R}(\mathcal{C}), D)\), we need to define some additional terminology. Let \(\ell_1, \cdots, \ell_w\) be the sequence of the transaction lengths of \(D\), i.e., for each value \(\ell\) for which there is at least a transaction in \(D\) of length \(\ell\), there is one (and only one) index \(i\), \(1 \leq i \leq w\) such that \(\ell_i = \ell\). Assume that the \(\ell_i\)'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 number of transactions in \(D\) that have length at least \(\ell_i\) and such that for no two \(\tau', \tau''\) of them we have either \(\tau' \subseteq \tau''\) or \(\tau'' \subseteq \tau'\). The sequences \((\ell_i)^w\) and a sequence \((L_i)^w\) of upper bounds to \((L_i)^w\) can be computed efficiently with a scan of the dataset. Let now \(q_i\) be the optimal profit of the SUKP associated to \(\mathcal{C}\) with capacity \(\ell_i\), and let \(b_i = \lceil \log_2 q_i \rceil + 1\). The following lemma uses these sequences to show how to obtain an upper bound to the empirical VC-dimension of \(\mathcal{C}\) on \(D\).

**Lemma 2.** Let \(j\) be the minimum integer for which \(b_j \leq L_j\). Then \(\text{EVC}(\mathcal{C}, D) \leq b_j\).

**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 \(b_j\) 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 1. Hence \(\text{EVC}(\mathcal{C}, D) \leq b_j\). \(\square\)

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

The SUKP is NP-hard in the general case, but there are known restrictions for which it can be solved in polynomial time using dynamic programming \[12\]. For our case available optimization problem solvers can compute the optimal solution reasonably fast even for very large instances with thousands of items and hundred of thousands of itemsets in \(\mathcal{C}\). Moreover, 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.

The range set associated to \(2^I\) is particularly interesting for us. Better bounds to \(\text{VC}(\mathcal{R}(2^I))\) and \(\text{EVC}(\mathcal{R}(2^I), D)\) are available.

**Theorem 3** \([23]\). Let \(D\) be a dataset built on a ground set \(I\). The d-index \(d(D)\) of \(D\) is the maximum integer \(d\) such that \(D\) contains at least \(d\) transactions of length at least \(d\) that form an antichain. Then \(\text{EVC}(\mathcal{R}(2^I), D) \leq d(D)\).

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

Riondato and Upfal \[23\] presented an efficient algorithm to compute an upper bound to the d-index of a dataset with a single linear scan of the dataset \(D\). The upper bound presented in Thm. 3 is tight: there are datasets for which \(\text{EVC}(\mathcal{R}(2^I), D) = d(D)\) \[23\]. This implies that the upper bound presented in Corol. 2 is also tight.

### 4 Finding the true frequent itemsets

In this section we present two methods to identify True Frequent Itemsets with respect to a minimum true frequency threshold \(\theta \in (0, 1]\), while guaranteeing that the Family-Wise Error Rate (FWER) is less than \(\delta\), for some user-specified parameter \(\delta \in (0, 1]\). In other words, we present two algorithms to find a collection \(\mathcal{C}\) of itemsets such that
\[ \Pr(\exists A \in \mathcal{C} \text{ s.t. } t_{\pi}(A) < \theta) < \delta \] .

Our methods achieve this goal using the same tools in two different ways and are applicable to different and complementary situations. Specifically the first method can be used when it is possible to randomly split the available dataset in two parts, and is inspired by the holdout technique \cite{28}. While splitting the dataset may be advantageous in certain cases, this may not always be possible and in general a better characterization of the TFIs may be obtained using the whole dataset. We develop a second method for this situation.

The intuition behind the two methods is similar. Each method starts by building a set \( \mathcal{G} \) of “candidates TFIs”. To each itemset \( A \in \mathcal{G} \) we associate a null hypothesis \( H_A = "t_{\pi}(A) < \theta" \), and the tests use the frequency \( f(A) \) of \( A \) in the dataset (or in a portion of the dataset) as the test statistic. If the frequency falls into the acceptance region \([0, \theta + \varepsilon]\), where \( \varepsilon \) is a function of \( \delta \) and \( \mathcal{G} \) computed by our methods, then \( H_A \) is accepted, otherwise \( H_A \) is rejected and \( A \) is flagged as True Frequent and included in the output collection \( \mathcal{C} \). Any itemset not in \( \mathcal{G} \) is not considered and will not be reported in output. It should be clear that the definition of the acceptance region is critical for the method to have the desired FWER at most \( \delta \); one needs to compute a \( \varepsilon \) such that

\[ \Pr \left( \exists A \in \mathcal{C} \text{ s.t. } t_{\pi}(A) < \theta \right) = \Pr \left( \exists A \in \mathcal{G} \text{ s.t. } t_{\pi}(A) < \theta \text{ and } f(A) \geq \theta + \varepsilon \right) \leq \delta \ . \]

The two methods we present differ in the definition of \( \mathcal{G} \) and in the computation of \( \varepsilon \), but both use the tools we developed in Sect. \[5\].

### 4.1 Method 1: a split-dataset approach

We now present and analyze our first method for identifying TFIs, which draws inspiration from the holdout technique presented by Webb \cite{28}. This test requires that the dataset \( \mathcal{D} \) could be randomly split into two parts, not necessarily of the same size: an exploratory part \( \mathcal{D}_e \) and an evaluation part \( \mathcal{D}_v \). The method works in two phases: we first use the exploratory part \( \mathcal{D}_e \) to identify a small set \( \mathcal{G} \) of candidate TFIs which will be used to compute the acceptance region, then we test these same itemsets using their frequencies in the evaluation part \( \mathcal{D}_v \) as test statistics.

Let \( \delta_e \) and \( \delta_v \) be such that \( (1 - \delta_e)(1 - \delta_v) \geq (1 - \delta) \). Let \( \mathcal{R}(2^I) \) be the range space of all itemsets. We use Corol. \[2\] (resp. Thm. \[3\]) to compute an upper bound \( d' \) to \( \text{VC}(\mathcal{R}(2^I)) \) (resp. \( d'' \) to \( \text{EVC}(\mathcal{R}(2^I), \mathcal{D}_e) \)). Then, given that \( \mathcal{D}_e \) is still a collection of i.i.d. samples from the generative distribution \( \pi \), we can use \( d' \) in Thm. \[1\] (resp. \( d'' \) in Thm. \[2\] ) to compute an \( \varepsilon_{\mathcal{G}}' \) (resp. an \( \varepsilon_{\mathcal{G}}'' \) ) such that \( \mathcal{D}_e \) is, with probability at least \( 1 - \delta_1 \), an \( \varepsilon_{\mathcal{G}}' \)-approximation (resp. \( \varepsilon_{\mathcal{G}}'' \)-approximation) to \( (\mathcal{R}(2^I), \pi) \). Then, if we let \( \varepsilon_v = \min \{ \varepsilon_{\mathcal{G}}', \varepsilon_{\mathcal{G}}'' \} \), \( \mathcal{D}_v \) is, with probability at least \( 1 - \delta_1 \), an \( \varepsilon_v \)-approximation to \( (\mathcal{R}(2^I), \pi) \).

In the first phase of the set we compute the collections of itemsets \( \mathcal{C}_e = \{ X \subseteq I : f_{\mathcal{D}_e}(X) \geq \theta + \varepsilon_e \} \) and \( \mathcal{G} = \{ X \subseteq I : \theta \leq f_{\mathcal{D}_e}(X) < \theta + \varepsilon_v \} \). To obtain these sets, we extract the set \( \text{Fl}(\mathcal{D}_e, I, \theta) \) and partition it appropriately into \( \mathcal{C}_e \) and \( \mathcal{G} \). In the second phase, we compute a value \( \varepsilon_v \) such that, with probability at least \( 1 - \delta_2 \), the evaluation dataset \( \mathcal{D}_v \) is an \( \varepsilon_v \)-approximation to \( (\mathcal{R}(\mathcal{G}), \pi) \). In order to obtain \( \varepsilon_v \) through Thms. \[1\] and \[2\], we need to compute upper bounds to \( \text{VC}(\mathcal{R}(\mathcal{G})) \) and \( \text{EVC}(\mathcal{R}(\mathcal{G}), \mathcal{D}_v) \). We solve SUKPs associated to \( \mathcal{G} \) and obtain such bounds, as stated in Corol. \[1\] and Lemma \[2\] respectively. As in the first phase, we use these bounds and Thm. \[1\] (resp. Thm. \[2\] ) to compute a \( \varepsilon_v' \) (resp. \( \varepsilon_v'' \ )) such that \( \mathcal{D}_v \) is, with probability at least \( 1 - \delta_2 \), an \( \varepsilon_v' \)-approximation (resp. an \( \varepsilon_v'' \)-approximation) to \( (\mathcal{R}(\mathcal{G}), \pi) \). Once we have obtained \( \varepsilon_v = \min \{ \varepsilon_v', \varepsilon_v'' \} \), we compute the set

\[ \mathcal{C}_v = \{ X \subseteq I : X \in \mathcal{G} \text{ and } f_{\mathcal{D}_v}(X) \geq \theta + \varepsilon_v' \} = \mathcal{G} \cap \text{Fl}(\mathcal{D}_v, I, \theta + \varepsilon_v) \ . \]

The method returns the collection of itemsets \( \mathcal{C}_e \cup \mathcal{C}_v \). The proof that \( \Pr(\exists A \in \mathcal{C}_e \cup \mathcal{C}_v \text{ s.t. } t_{\pi}(A) < \theta) \leq \delta \) comes from the definition of \( \varepsilon_e \) and \( \varepsilon_v \) through Thms. \[1\] and \[2\].

**Lemma 3.** The FWER of Method 1 is at most \( \delta \).

**Proof.** Consider the two events \( E_e = "\mathcal{D}_e \text{ is an } \varepsilon_e \text{-approximation for } (\mathcal{R}(2^I), \pi)" \) and \( E_v = "\mathcal{D}_v \text{ is an } \varepsilon_v \text{-approximation for } (\mathcal{R}(\mathcal{G}), \pi)" \). From the above discussion it follows that the event \( E = E_e \cap E_v \) occurs with probability at least \( 1 - \delta \).

Suppose from now on that indeed \( E \) occurs.

Given that \( E \) occurs, then all the itemsets with frequency in \( \mathcal{D}_e \) at least \( \theta + \varepsilon_e \) must have a real frequency at least \( \theta \). This equals to say that all itemsets in \( \mathcal{C}_e \) are True Frequent Itemsets (\( \mathcal{C}_e \subseteq TFI(\theta) \)).
Given that $E_v$ occurs, then we know that all itemsets in $G$ have frequency in $D_v$ that is at most $\varepsilon_v$ far from their true frequency:

$$\max_{A \in G} |t_v(A) - f_{D_v}(A)| \leq \varepsilon_v.$$  

In particular this means that an itemset $A \in G$ can have $f_{D_v}(A) \geq \theta + \varepsilon_v$ only if $t_v(A) \geq \theta$, that is only if $A \in TFI(\pi, I, \theta)$. Hence, $C_v \subseteq TFI(\pi, I, \theta)$.

We can then conclude that if the event $E$ occurs, we have $C_v \cup C_\pi \subseteq TFI(\pi, I, \theta)$. Since $E$ occurs with probability at least $1 - \delta$, this equals to say that

$$\Pr(\exists A \in C_v \cup C_\pi : t(A) < \theta) \leq \delta.$$  

### 4.2 Method 2: a full-dataset approach

We now present a method for identifying TFI’s that can be used in all cases, even when it is not possible or desirable to split the available dataset $D$ in two parts. Our experimental evaluation also showed that this method can often identify a larger fraction of the TFIs than Method 1, at the expense of requiring more computational resources. The intuition behind the definition of the acceptance region $[0, \theta + \varepsilon]$ is the following. Let $B$ be the negative border of $TFI(\pi, I, \theta)$, that is the set of itemsets not in $TFI(\pi, I, \theta)$ but such that all their proper subsets are in $TFI(\pi, I, \theta)$. If we can find an $\varepsilon$ such that $D$ is an $\varepsilon$-approximation to $(R(B), \pi)$ with probability at least $1 - \delta$, then, if this is the case, we have that any itemset $A \in B$ has a frequency $f_D(A)$ in $D$ less than $\theta + \varepsilon$, given that it must be $t(A) < \theta$. By the antimonotonicity property of the frequency, the same holds for all itemsets that are supersets of those in $B$. Hence, the only itemsets that can have frequency in $D$ greater or equal to $\theta + \varepsilon$ are those with true frequency at least $\theta$. In the following paragraphs we show how to compute $\varepsilon$.

Let $\delta_1$ and $\delta_2$ be such that $(1 - \delta_1)(1 - \delta_2) \geq 1 - \delta$. Following the same steps as in the first phase of Method 1, we can find an $\varepsilon_1$ such that $D$ is an $\varepsilon_1$-approximation for $(R(2^I), \pi)$ and frequency $t_A$ approximately to $[0, \theta + \varepsilon]$ with probability at least $1 - \delta_1$. Let $W$ be the negative border of $R(I, \theta - \varepsilon_1)$, $G = \{A \subseteq I : \theta - \varepsilon_1 \leq f_D(A) < \theta + \varepsilon_1 \}$, and $F = G \cup W$. We want to find an upper bound on the (empirical) VC-dimension of $B$. To this end, we use the fact that the negative border of a collection of itemsets is a maximal antichain on $2^I$.

**Lemma 4.** Let $\mathcal{Y}$ be the set of maximal antichains in $F$. If $D$ is an $\varepsilon_1$-approximation to $(R(2^I), \pi)$, then

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

**Proof.** Assume that $D$ is an $\varepsilon_1$-approximation to $(R(2^I), \pi)$. From the definition of $\varepsilon_1$ this happens with probability at least $1 - \delta$. Then $\text{TFI}(\pi, I, \theta) \subseteq G \cup C_\pi$. From this and the definition of the negative border and of $F$, we have that $B \subseteq F$. Since $B$ is a maximal antichain, then $B \in \mathcal{Y}$, and the thesis follows immediately.

In order to compute upper bounds to $\text{VC}(R(B))$ and $\text{EVC}(R(B), D)$ we can solve slightly modified SUKPs associated to $F$ with the additional constraint that the optimal solution, which is a collection of itemsets, must be a maximal antichain. Using these bounds in Thms. [1] and [2], we compute an $\varepsilon_2$ such that, with probability at least $1 - \delta_2$, $D$ is an $\varepsilon_2$-approximation to $(R(B), \pi)$. The method returns the collection of itemsets $\text{Fl}(D, I, \theta + \varepsilon_2)$, and we prove that the probability that this collection contains any itemset not in TFI’s is at most $\delta$.

**Lemma 5.** The FWER of Method 2 is at most $\delta$.

**Proof.** Consider the two events $E_1$="$D$ is an $\varepsilon_1$-approximation for $(R(2^I), \pi)$" and $E_2$="$D$ is an $\varepsilon_2$-approximation for $(R(B), \pi)$". From the above discussion and the definition of $\delta_1$ and $\delta_2$ it follows that the event $E = E_1 \cap E_2$ occurs with probability at least $1 - \delta$. Suppose from now on that indeed $E$ occurs.

Since $E_1$ occurs, then Lemma [4] holds, and the bounds we compute by solving the modified SUKP problems are indeed bounds to $\text{VC}(R(B))$ and $\text{EVC}(R(B), D))$. Then the computation of $\varepsilon_2$ is valid. Since $E_2$ also occurs, then for any $A \in B$ we have $|t(A) - f_D(A)| \leq \varepsilon_2$, but given that $t(A) < \theta$, because the elements of $B$ are not TFIs, then we have $f_D(\theta) < \theta + \varepsilon_2$. Because of the antimonotonicity property of the frequency and the definition of $B$, this hold for any itemset that is not True Frequent. Hence, the only itemsets that can have a frequency in $D$ at least $\theta + \varepsilon_2$ are the TFIs, so $\text{Fl}(D, I, \theta + \varepsilon_2) \subseteq TFI(\pi, I, \theta)$.
5 Experimental evaluation

We conducted an extensive evaluation to assess the practical applicability and the statistical power of the methods we propose. In the following sections we describe the methodology and present the results.

Implementation. We implemented our methods in Python 3.3. To mine the FIs, we used the C implementation by Grahne and Zhu [13]. 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 2.6.32.

5.1 Datasets generation

We evaluated our methods using datasets from the FIMI’04 repository: accidents [9], BMS-POS, chess, connect, kosarak, pumsb*, and retail [5]. These datasets differ in size, number of items, and more importantly distribution of the frequencies of the itemsets [11]. For currently available datasets the ground truth is not known, that is, we do not know the distribution $\pi$, hence neither the true frequencies $t_{i\pi}$ of the itemsets, which we need to evaluate the performances of our methods. Therefore to establish a ground truth, we created new enlarged versions of the datasets from the FIMI repository by sampling transactions uniformly at random from them until the desired size (in our case 20 million transactions) has been reached. We considered the true frequencies of the itemsets to be the frequencies in the original datasets. Given that our method to control the FWER is distribution-independent this is a valid way to establish a ground truth. We used these enlarged datasets in our experiment.

5.2 Control of the FWER

We evaluated the capability of our methods to control the FWER by first creating a number of enlarged datasets, and then run our methods to extract a collection of TFIs from thee dataset. We used a wide range of values (see Table 1) for the minimum true frequency threshold $\theta$ and fixed $\delta = 0$. We repeated each experiment on 20 different enlarged datasets generated from the same distribution, for each original distribution. In all the hundreds of runs of our algorithms, the returned collection of itemsets never contained any false positives, i.e., always contained only TFIs. In other words, the precision of the output was 1.0 in all our experiments. This means that not only our methods can control the FWER effectively, but they are more conservative than what is guaranteed by the theoretical analysis.

5.3 Statistical power and comparison with other methods

The power of a statistical test is the probability that the test will reject the null hypothesis when the null hypothesis is false. In the case of TFIs, this corresponds to the probability of including a TFIs in the output collection. It is often difficult to analytically quantify the statistical power of a test, especially in multiple hypothesis testing settings with correlated hypotheses. This is indeed our case, and we therefore conducted an empirical evaluation of the statistical power of our methods by assessing what fraction of the total number of TFIs is reported in output. This is corresponds to evaluating the recall of the output collection. We fixed $\delta = 0.1$, and considered $\theta$ in a wide range (see Table 1) and repeated each experiment 20 times on different datasets, finding insignificant variance in the results. We also wanted to compare the power of our methods with that of established techniques that can control the probability of Type-1 errors. To this end we adapted the holdout technique proposed by Webb [28] to the TFIs problem and compared its power with that of Method 1. In this technique, the dataset is randomly split into two portions, an exploratory part and an evaluation part. First the frequent itemsets w.r.t. $\theta$ are extracted from the exploratory part. Then, each of the null hypotheses of these itemsets is tested using a Binomial test on the frequency of the itemsets in the evaluation dataset, with critical value $\delta/k$, where $k$ is the number of patterns found in the first step. To evaluate the power of Method 2, which does not need to split the dataset, we compared its results with the power of a method that uses the Bonferroni correction over all possible itemsets to test, with a Binomial test, the FIs w.r.t. $\theta$ obtained from the entire dataset.

The statistical power evaluation and the comparison with the holdout technique are presented in Table 1, where we reported the average over 20 runs for each experiment. Only partial results are reported for the full dataset approaches

[^1]: http://fimi.ua.ac.be/data/
| Dataset   | $\theta$ | No. of TFIs | Split dataset Holdout | Split dataset Method 1 | Full dataset Bonferroni | Full dataset Method 2 |
|-----------|----------|-------------|------------------------|------------------------|--------------------------|------------------------|
| accidents | 0.8      | 149         | 94.631                 | 97.987                 | 83.893                   | 95.973                 |
|           | 0.7      | 529         | 97.543                 | 98.488                 | 92.439                   | 98.488                 |
|           | 0.6      | 2074        | 98.505                 | 98.987                 | 96.625                   | 99.132                 |
|           | 0.5      | 8057        | 98.349                 | 99.007                 | 94.551                   | 99.081                 |
|           | 0.45     | 16123       | 98.177                 | 98.915                 | 94.691                   | 99.076                 |
|           | 0.4      | 32528       | 98.032                 | 98.761                 | 94.774                   | 98.970                 |
|           | 0.35     | 68222       | 98.140                 | 98.666                 |                         |                        |
|           | 0.3      | 149545      | 98.033                 | 98.529                 |                         |                        |
|           | 0.25     | 346525      | 98.165                 | 98.382                 |                         |                        |
|           | 0.2      | 889883      | 97.995                 | 98.057                 |                         |                        |
| BMS-POS   | 0.05     | 59          | 98.305                 | 93.220                 | 84.746                   | 93.220                 |
|           | 0.03     | 134         | 99.254                 | 99.254                 | 88.060                   | 99.254                 |
|           | 0.02     | 308         | 98.377                 | 95.130                 | 84.740                   | 95.455                 |
|           | 0.01     | 1099        | 98.914                 | 98.293                 | 81.620                   | 88.171                 |
|           | 0.0075   | 1896        | 96.097                 | 92.331                 |                         |                        |
| chess     | 0.8      | 8227        | 97.265                 | 98.918                 | 96.475                   | 99.198                 |
|           | 0.75     | 20993       | 96.375                 | 98.042                 | 95.756                   | 98.418                 |
|           | 0.7      | 48731       | 97.613                 | 99.001                 |                         |                        |
|           | 0.6      | 254944      | 97.243                 | 98.464                 |                         |                        |
|           | 0.5      | 1272932     | 97.352                 | 98.293                 |                         |                        |
| connect   | 0.9      | 27127       | 93.217                 | 97.213                 | 89.369                   | 97.921                 |
|           | 0.875    | 65959       | 94.481                 | 96.040                 |                         |                        |
|           | 0.85     | 142127      | 94.242                 | 96.882                 |                         |                        |
|           | 0.8      | 539375      | 95.047                 | 97.464                 |                         |                        |
| kosarak   | 0.04     | 42          | 97.620                 | 97.620                 | 73.810                   | 95.238                 |
|           | 0.035    | 50          | 100                    | 98.000                 | 72.000                   | 98.000                 |
|           | 0.03     | 65          | 96.923                 | 98.000                 | 93.846                   |                        |
|           | 0.025    | 82          | 98.780                 | 97.561                 |                         |                        |
|           | 0.02     | 121         | 100                    | 97.521                 |                         |                        |
|           | 0.015    | 189         | 98.942                 | 93.122                 |                         |                        |
| pumsb'    | 0.55     | 305         | 92.131                 | 94.426                 | 80.984                   | 95.401                 |
|           | 0.5      | 679         | 99.853                 | 99.853                 | 92.931                   | 99.853                 |
|           | 0.49     | 804         | 97.637                 | 98.507                 | 86.692                   | 98.756                 |
|           | 0.45     | 1913        | 96.968                 | 97.909                 |                         |                        |
|           | 0.4      | 27354       | 98.183                 | 99.101                 |                         |                        |
|           | 0.35     | 116787      | 96.435                 | 96.972                 |                         |                        |
|           | 0.3      | 432698      | 96.567                 | 97.326                 |                         |                        |
| retail    | 0.03     | 32          | 100                    | 100                    | 62.500                   | 100                    |
|           | 0.025    | 38          | 100                    | 97.368                 | 84.211                   | 97.368                 |
|           | 0.02     | 55          | 100                    | 94.545                 | 72.340                   | 95.745                 |
|           | 0.015    | 84          | 100                    | 95.238                 |                         |                        |
|           | 0.01     | 159         | 99.371                 | 93.082                 |                         |                        |

Table 1: Statistical power of various methods to extract the True Frequent Itemsets while controlling the FWER ($\delta = 0.1$). In bold the best result for each setting (split or full dataset).
because of issues with the amount of computational resources required by this method when there is a very high number of itemsets in the negative border. The statistical power of our methods is very high in general, for all the tested datasets and values for $\theta$. The comparison of Method 1 with the holdout technique shows that the relative performances of these methods are dataset specific: the statistical power seems to depend heavily on the true frequency distribution of the itemsets. From the table it is possible to see that Method 1 performs better than the holdout technique when the number of TFIs is high, while the opposite is true when the minimum threshold frequency $\theta$ is very low. We conjecture that this is due to the fact that the value for $\theta$ is not taken in consideration when computing the $\varepsilon$ to define the acceptance region $[0, \theta + \varepsilon]$ in Method 1, as the computation of $\varepsilon$ in Thms. 1 and 2 does not depend on $\theta$. In future work, we plan to investigate whether it is possible to include $\theta$ as part of the computation of $\varepsilon$. On the other hand the direct correction applied in the holdout technique becomes less and less powerful as the number of TFIs increases because it does not take into account the many correlations between the itemsets. Given that in common applications the number of frequent itemsets of interest is large, we believe that in real scenarios Method 1 can be more effective in extracting the True Frequent Itemsets while rigorously controlling the probability of false discoveries. For the full dataset case, we can see that Method 2 performs much better than the Bonferroni correction method in all cases, due to not having to take into account, when computing the acceptance region, all possible hypotheses. One can also compare Method 1 and Method 2 to each other, given that Method 2 can be used in cases where Method 1 can. Method 2 manages to achieve higher statistical power than Method 1 in almost all cases, but we must stress that this is at the expense of a longer running time and an higher amount of required memory.

5.4 Runtime evaluation

We found our methods to be quite efficient in practice in terms of time needed to compute the output collection. The running time is dominated by the time needed to compute the negative border (only for Method 2) and to solve the SUKPs. Various optimizations and shortcuts can be implemented to substantially speed up our methods (see also the discussion in Sect. 3).

6 Conclusions

In this work we developed two methods to approximate the collection of True Frequent Itemsets while guaranteeing that the probability of any false positive (FWER) is bounded by a user-specified threshold, and without imposing any restriction on the generative model. We used tools from statistical learning theory to develop and analyze our methods. The two methods cover complementary use scenarios. Our experimental evaluation shows that the methods we propose achieve very high statistical power and are competitive or superior to existing techniques to solve the same problem. There are a number of directions for further research, including improving the power of methods to identify TFIs by making the computation of the acceptance region dependent on the minimum true frequency threshold, and studying lower bounds to the VC-dimension of the range set of a collection of itemsets. Moreover, while this work focuses on itemsets mining, further research is needed for the extraction of more structured (e.g., sequences, graphs) True Frequent patterns.

7 References

[1] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD Rec. 22, 207–216 (1993)
[2] Alon, N., Spencer, J.H.: The Probabilistic Method. Wiley, third edn. (2008)
[3] Bolton, R.J., Hand, D.J., Adams, N.M.: Determining hit rate in pattern search. In: Pattern Detection and Discovery. LNCS, vol. 2447, pp. 36–48 (2002)
[4] Boucheron, S., Bousquet, O., Lugosi, G.: Theory of classification : A survey of some recent advances. ESAIM: Probab. and Stati. 9, 323–375 (2005)
[5] Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Using association rules for product assortment decisions: A case study. KDD’99 (1999)

[6] Dudoit, S., Shaffer, J., Boldrick, J.: Multiple hypothesis testing in microarray experiments. Statis. Science 18(1), 71–103 (2003)

[7] DuMouchel, W., Pregibon, D.: Empirical Bayes screening for multi-item associations. KDD’01 (2001)

[8] Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surveys 38(3) (2006)

[9] Geurts, K., Wets, G., Brijs, T., Vanhoof, K.: Profiling of high-frequency accident locations by use of association rules. Transp. Res. Rec.: J. of the Transp. Res. Board 1840, 123–130 (2003)

[10] Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. ACM Trans. on Knowl. Disc. from Data 1(3) (2007)

[11] Goethals, B., Zaki, M.J.: Advances in frequent itemset mining implementations: report on FIMI’03. SIGKDD Explor. Newsl. 6(1), 109–117 (2004)

[12] Goldsmith, O., Nehme, D., Yu, G.: Note: On the set-union knapsack problem. Naval Research Logistics 41(6), 833–842 (1994)

[13] Grahne, G., Zhu, J.: Efficiently using prefix-trees in mining frequent itemsets. FIMI’03 (2003)

[14] Hämäläinen, W.: StatApriori: an efficient algorithm for searching statistically significant association rules. Knowl. and Inf. Sys. 23(3), 373–399 (2010)

[15] Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. and Knowl. Disc. 15, 55–86 (2007),

[16] Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan Kaufmann (2006)

[17] Hanhijärvi, S.: Multiple hypothesis testing in pattern discovery. DS’11 (2011)

[18] Har-Peled, S., Sharir, M.: Relative $(p, \varepsilon)$-approximations in geometry. Discr. & Comput. Geom. 45(3), 462–496 (2011),

[19] Kirsch, A., Mitzenmacher, M., Pietracaprina, A., Pucci, G., Upfal, E., Vandin, F.: An efficient rigorous approach for identifying statistically significant frequent itemsets. J. of the ACM 59(3), 12:1–12:22 (2012),

[20] Liu, G., Zhang, H., Wong, L.: Controlling false positives in association rule mining. Proc. of the VLDB Endow. 5(2), 145–156 (2011),

[21] Löffler, M., Phillips, J.M.: Shape fitting on point sets with probability distributions. ESA’09 (2009)

[22] Megiddo, N., Srikant, R.: Discovering predictive association rules. KDD’98 (2008)

[23] Riondato, M., Upfal, E.: Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. ECML PKDD’12 (2012)

[24] Riondato, M., Vandin, F.: Finding the True Frequent Itemsets. CoRR abs/1301.1218 (2013). [http://arxiv.org/abs/1301.1218]

[25] Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. Data Min. and Knowl. Disc. 2(1), 39–68 (1998)

[26] Teytaud, O., Lallich, S.: Contribution of statistical learning to validation of association rules (2001)
[27] Vapnik, V.N., Chervonenkis, A.J.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications 16(2), 264–280 (1971)

[28] Webb, G.I.: Discovering significant patterns. Mach. Learn. 68(1), 1–33 (2007)

[29] Webb, G.I.: Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. Mach. Learn. 71, 307–323 (2008)