An Asymptotically Tighter Bound on Sampling for Frequent Itemsets Mining

Shiyu Ji, Kun Wan
{shiyu,kun}@cs.ucsb.edu

Abstract

In this paper we present a new error bound on sampling algorithms for frequent itemsets mining. We show that the new bound is asymptotically tighter than the state-of-art bounds, i.e., given the chosen samples, for small enough error probability, the new error bound is roughly half of the existing bounds. Based on the new bound, we give a new approximation algorithm, which is much simpler compared to the existing approximation algorithms, but can also guarantee the worst approximation error with precomputed sample size. We also give an algorithm which can approximate the top-k frequent itemsets with high accuracy and efficiency.

1 Introduction

Frequent Itemsets (FI) mining has been popular in research recently [1, 8, 18]. The goal of FI mining is to find out the items that most frequently appear in the observed transactions, e.g., the researchers who are the most prolific in writing papers with others, the patterns that appear frequently in long pieces of genetic code, etc.

In the era of big data, to compute the exact frequencies can be very time consuming. Thus in many cases approximate values are also acceptable [1, 12, 6, 8, 10, 18]. For FI mining in large scale transactional datasets, we often take samplings on the transactions, and compute the frequencies of the itemsets among the sampled transactions as approximate results of their true frequencies among all the transactions. Usually the sampling size is much less than the scale of all the transactions, and the approximations can achieve acceptable precision. Also in reality we often only want to know the most frequent itemsets without the need of their actual frequencies, and there are already many works [1, 12, 6, 8, 10] on this area. Thus FI approximation can be useful in practice.

The state-of-art progressive sampling based FI approximation algorithms [18] need an upper bound of the approximation error for the worst case, i.e., the maximum error the algorithm can generate among all the items. The algorithms keep taking new samples until the upper bound is less than the acceptable threshold. Hence how to bound the maximum error as tightly as possible is an interesting problem. The current bounds use some results of Rademacher average in statistical learning theory [24, 25, 3, 2]. However, we find that based on the ideas given by [3, 23], we can develop a new upper bound *without* Rademacher average. We also find that this new bound is asymptotically tighter than the existing bounds, i.e., given the chosen samples, as the allowed error probability approaches zero, the new bound is roughly only half of the existing ones. This implies that by using the new bound, a progressive sampling based FI approximation algorithm can reach the guaranteed accuracy with much fewer samples. We also notice that there is no parameter in the new bound that needs to be progressively computed. Hence the sample size that will guarantee the worst error can be precomputed.

Based on the similar idea, we also consider the top-k FI mining problem, which seeks for the k most frequent itemsets in the observed ones. We need to decide when the sampling should stop. The number of the sampled transactions is enough if the worst-case error upper bound is less than the frequency gap between the k-th and the $(k+1)$-th most frequent itemsets. Hence we propose a progressive approximation algorithm to address the top-k FI mining problem.

Our Contributions. We give a worst-case error upper bound that is asymptotically tighter than the state-of-art bounds, and propose an approximation algorithm which can guarantee the worst-case error upper bound with precomputed sample size. We also give a progressive sampling algorithm to find the top-k most frequent itemsets. Combining with existing methods, our algorithms can approximate the frequent itemsets accurately and efficiently.

The rest of this paper is organized as follows. Section 2 reviews the related research works. Section 3 introduces the notations and preliminaries throughout this paper. Section 4 gives the worst-case error upper bound without Rademarcher average and compares it with the existing ones. Section 5 proposes our approximation algorithms based on our upper bounds. Section 6 gives our evaluation results, which compare our algorithms with the state-of-art.

2 Related Works

Frequent Itemset Mining has been very popular in the communities of information retrieval and data mining [10]. Unsurprisingly, many algorithms that can compute the exact frequencies have been proposed, e.g., A-Priori algorithm [1], Park-Chen-Yu's algorithm [12], Multistage algorithms [6]. However it is very challenging to deal with large scaled data sets with limited main memory. Thus the classical exact algorithms may not fit well in practice. As a result, how to approximate the frequent itemsets by sampling has become interesting, since usually the sample size is much less than the entire data scale. Toivonen [23] was among the first to study sampling on FI approximation, and suggested the first worst-case error bound on frequencies. However his algorithm did not directly use the bound and still needed to parse all the dataset. Thus for scenarios like streams where the size of dataset is unbounded, we cannot use Toivonen's algorithm directly.

Sampling-based frequent itemset approximation has been studied extensively by the researchers. The first works on this problem used heuristic methods to progressively approximate the frequencies [4, 5, 13]. There were no guarantee on the worst-case error upper bound. To fix this, Riondato and Upfal were the first to propose FI approximation algorithms that could guarantee the worst error bounds by using results of Vapnik-Chervonenkis (VC) dimension [16, 17] and Rademacher average [18]. Note that in statistical learning theory VC dimension and Rademacher average are usually used to address the worst-case error upper bound for *infinite case*, i.e., the number of possible functions in the learning model is infinite [3]. However in the case of FI mining, since there are only *finite* itemsets, it is possible to develop bounds without VC dimension or Rademacher average [3]. In this paper we apply this idea on FI mining problem. Riondato et al. also considered using parallelism in FI mining [15], which is an orthogonal topic to sampling-based FI approximation.

In practice we are often only interested in the most frequent itemsets. Thus top-k FI mining is a popular research topic with many research works [14, 22, 18, 20]. Another interesting question is to find all itemsets with frequencies larger than a threshold. Savasere, Omiecinski, and Navathe [21] give an two-pass algorithm (called SON algorithm) that can find the exact solutions. We will use SON algorithm to significantly reduce the number of itemsets to be observed, and then apply our algorithms to approximate the frequencies and select the top k ones. Also Toivonens Algorithm [23] is an alternative way to find the most frequent itemsets given a threshold.

3 Preliminaries

3.1 Frequency of Itemset

In this paper we use the notations and definitions from Riondato and Upfal's pioneering work [18]. Let $\mathcal{I}$ be the set of items. A transaction τ is a subset of $\mathcal{I}$ (i.e., $\tau \subseteq \mathcal{I}$). An itemset A is a set of items that appear together in a transaction τ, i.e., $A \subseteq \tau$. Clearly any itemset is also a subset of $\mathcal{I}$. Let transactional dataset $\mathcal{D}$ be the set of all the transactions. In this paper we always assume $\mathcal{D}$ is a finite set. Denote by $T_{\mathcal{D}}(A)$ the set of all the transactions in $\mathcal{D}$ that contain the itemset A. $T_{\mathcal{D}}(A)$ is also known as the support set of A in

$\mathcal{D}$. If $\mathcal{D}$ is a finite set, we can define the frequency of itemset A in $\mathcal{D}$ as the fraction of transactions in $\mathcal{D}$ that contain A.

$$f_{\mathcal{D}}(A) = |T_{\mathcal{D}}(A)|/|\mathcal{D}|.$$

Clearly $0 \leq f_{\mathcal{D}}(A) \leq 1$ for any $A \subseteq \mathcal{I}$.

The goal of our sampling algorithm is to approximate $f_{\mathcal{D}}(A)$ given an itemset A as accurately as possible.

3.2 Approximation Algorithms

An (ϵ, δ)-approximation algorithm of the frequencies $f_{\mathcal{D}}(\cdot)$ takes as input all the items $\mathcal{I}$ and outputs a sampled average $f_{\mathcal{S}}(A)$ for each $A \subseteq \mathcal{I}$ such that with probability at least $1 - \delta$,

$$\max_{A \subseteq \mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| \leq \epsilon.$$

We often use progressive sampling [18, 19], i.e., to keep taking more samples until a stopping condition is reached. A stopping condition usually takes the form $\Delta(n, \delta) \leq \epsilon$, where n is the number of samples that have been taken, and Δ is an upper bound of the worst approximation error given by statistical learning theory. Note that Δ is usually a function of n and δ.

There is a variant called top-k approximation, which returns the k most frequent itemsets among the observed ones based on the approximated frequencies. This is quite popular in practice since we are often only interested in the most common itemsets.

3.3 Risk Bounds

We briefly review some risk bounds in statistical learning theory [2] with the background of frequent itemsets mining.

For each itemset $A \subseteq \mathcal{I}$, define the indicator function $\phi_A : 2^{\mathcal{I}} \to \{0, 1\}$ as follows.

$$\phi_A(\tau) = \begin{cases} 1 & \text{if } A \subseteq \tau \\ 0 & \text{otherwise} \end{cases}, \quad \tau \subseteq \mathcal{I}.$$

Clearly, the frequency $f_{\mathcal{D}}(A)$ equals to the *true* average of $\phi_A(\tau)$ where τ goes over all the transactions in $\mathcal{D}$.

$$f_{\mathcal{D}}(A) = \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \phi_A(\tau).$$

Similarly let $\mathcal{S}$ be the set of the sampled transactions. Then the *sampled* average of $\phi_A(\tau)$ can be defined as

$$f_{\mathcal{S}}(A) = \frac{1}{|\mathcal{S}|} \sum_{\tau \in \mathcal{S}} \phi_A(\tau).$$

Clearly $f_{\mathcal{S}}(A)$ is the frequency of A appearing in the sampled transactions $\mathcal{S}$.

Assume $|\mathcal{S}| = n$. For each transaction $\tau_i \in \mathcal{S}$, let σ_i be a Rademacher random variable taking value from $\{-1, 1\}$ with uniform probability distribution. The σ_i's are independent. Assuming $\mathcal{I}$ is finite, we define the sample conditional Rademacher average as follows.

$$\mathcal{R}_{\mathcal{S}} = \mathbb{E}_{\sigma} \left[\max_{A \subseteq \mathcal{I}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi_A(\tau_i) \right],$$

where $\mathbb{E}_{\sigma}$ denotes the expectation taken over all the random variables σ_i's, conditionally on the sample $\mathcal{S}$.

The following theorem tells us that Rademacher average can be used to upper bound the approximation error, even for the worst case.

Theorem 1. (Theorem 3.2, [2]) For any $\delta > 0$, with probability at least $1 - \delta$,

$$\max_{A \subseteq \mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| \leq 2\mathcal{R}_{\mathcal{S}} + \sqrt{\frac{2\log(2/\delta)}{n}}.$$

If we want to use the upper bound given in Theorem 1 in an approximation algorithm, we still need to upper bound the $\mathcal{R}_{\mathcal{S}}$. A classical result is given by Massart [11].

Theorem 2. (Lemma 5.2, [11]) Let $\ell = \max_{A \subseteq \mathcal{I}} [\sum_{i=1}^{n} \phi_A(\tau_i)^2]^{1/2}$ where each $\tau_i \in \mathcal{S}$. Then

$$\mathcal{R}_{\mathcal{S}} \leq \frac{\ell}{n}\sqrt{2\log N},$$

where $N = 2^{|\mathcal{I}|}$ and $n = |\mathcal{S}|$.

Hence we have the following stopping condition for an (ϵ, δ)-approximation sampling algorithm.

$$\Delta_1 := \frac{2\ell}{n}\sqrt{2\log N} + \sqrt{\frac{2\log(2/\delta)}{n}} \leq \epsilon.$$

However for many applications the above bound is not tight enough [18, 19]. In the next section we will first review the state-of-art bound on the worst approximation error, and then give an asymptotically tighter bound.

4 Refining the Upper Bound

The reason why the bound given in the previous section is often not tight enough in practice is that the ℓ defined in Theorem 2 can be quite large. Suppose there is an itemset A that almost always appears in every transaction in $\mathcal{D}$. Then no matter which sample the algorithm chooses, ℓ is roughly $\sqrt{n}$. For $\delta = 0.01$, $N = 2^{1000}$, even 100,000 samples are taken, the upper bound is still larger than 0.15. For many applications such an upper bound cannot be acceptable and thus we need to take more samples. Clearly if the upper bound is tighter, a lot of samples can be saved.

4.1 A Brief Review on the Existing Results

Riondato and Upfal [18] attempted to give a tighter bound of the Rademacher average $\mathcal{R}_{\mathcal{S}}$.

Theorem 3. (Theorem 3, [18], revised) Let $w : \mathbb{R}^+ \to \mathbb{R}^+$ be the function defined as

$$w(s) = \frac{1}{s}\log \sum_{A \subseteq \mathcal{I}} \exp\left(\frac{s^2 \sum_{i=1}^{n} \phi_A(\tau_i)^2}{2n^2}\right).$$

Then $\mathcal{R}_{\mathcal{S}} \leq \min_{s>0} w(s)$.

Remark. Note that in Theorem 3, the summation in $w(s)$ takes *exactly* $2^{|\mathcal{I}|}$ terms. However in the original version in [18], the authors claimed that the summation could take much less than $2^{|\mathcal{I}|}$ terms. We argue that there is a gap between these two versions. Based on the proof given in [18], one can reach the inequality as follows.

$$\exp(s\mathcal{R}_{\mathcal{S}}) \leq \sum_{A \subseteq \mathcal{I}} \exp\left(\frac{s^2 \sum_{i=1}^{n} \phi_A(\tau_i)^2}{2n^2}\right). \tag{1}$$

Note that on the right hand side, each term in the summation is no less than 1. Hence when taking the logarithm on both sides and dividing by s, each of the $2^{|\mathcal{I}|}$ terms cannot be eliminated. Thus the range of the summation cannot be compressed.

Formally, suppose there is a set $\mathcal{V} \subseteq 2^{\mathcal{I}}$, where $2^{\mathcal{I}}$ denotes the power set of $\mathcal{I}$, such that

$$\alpha(s) := \sum_{A \in 2^{\mathcal{I}}} \exp\left(\frac{s^2 \sum_{i=1}^{n} \phi_A(\tau_i)^2}{2n^2}\right) \leq \sum_{A \in \mathcal{V}} \exp\left(\frac{s^2 \sum_{i=1}^{n} \phi_A(\tau_i)^2}{2n^2}\right) := \beta(s).$$

We take the limits as s approaches 0.

$$2^{|\mathcal{I}|} = \lim_{s \to 0} \alpha(s) \leq \lim_{s \to 0} \beta(s) = |\mathcal{V}|.$$

Hence $\mathcal{V} = 2^{\mathcal{I}}$. This implies any summation over only a part of $2^{\mathcal{I}}$ must be less than the summation over all of $2^{\mathcal{I}}$. Thus one cannot use Inequality (1) to reach Theorem 3 in [18].

4.2 Tighter Bound Without Rademacher Average

In statistical learning theory, the upper bound given by Theorem 1 is for the general case, i.e., the set of itemsets can be infinite or finite. However, for frequent itemsets mining, the number of itemsets is always finite (at most $2^{|\mathcal{I}|}$). Given this assumption, can we establish any upper bound without using the Rademacher average? Following the similar lines given by Boucheron, Bousquet and Lugosi [3] and Toivonen [23], we can give a positive answer.

For any $\epsilon > 0$,

$$\begin{aligned}
&\Pr[\max_{A \subseteq \mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| > \epsilon] \\
=&\Pr[\exists A \subseteq \mathcal{I}, f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A) > \epsilon \vee f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A) < -\epsilon] \\
\leq&\Pr[\exists A \subseteq \mathcal{I}, f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A) > \epsilon] + \Pr[\exists A \subseteq \mathcal{I}, f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A) < -\epsilon] \quad \text{(union bound)} \\
\leq&\sum_{A \subseteq \mathcal{I}} \Pr[f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A) > \epsilon] + \sum_{A \subseteq \mathcal{I}} \Pr[f_{\mathcal{S}}(A) - f_{\mathcal{D}}(A) > \epsilon] \quad \text{(union bound)}.
\end{aligned}$$

Recall Hoeffding's inequalities [9]. Let $X_1, \cdots, X_n$ be independent random variables bounded by the intervals $[a_i, b_i]$. Define the sampled average of them as

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Then for any $t > 0$,

$$\Pr[\overline{X} - \mathbb{E}[\overline{X}] > t] \leq \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}\right),$$

and

$$\Pr[\mathbb{E}[\overline{X}] - \overline{X} > t] \leq \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}\right).$$

Note that if we set $X_i = \phi_A(\tau_i)$, then X_i's are independent since τ_i's are independent, and thus $f_{\mathcal{D}}(A) = \mathbb{E}[\overline{X}]$ and $f_{\mathcal{S}}(A) = \overline{X}$. Based on Hoeffding's inequalities,

$$\Pr[f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A) > \epsilon] \leq \exp\left(-2n\epsilon^2\right),$$

and

$$\Pr[f_{\mathcal{S}}(A) - f_{\mathcal{D}}(A) > \epsilon] \leq \exp\left(-2n\epsilon^2\right).$$

Putting the above results together, we have

$$\begin{aligned}
&\Pr[\max_{A \subseteq \mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| > \epsilon] \\
\leq&2 \sum_{A \subseteq \mathcal{I}} \exp\left(-2n\epsilon^2\right) \\
=&2N \exp\left(-2n\epsilon^2\right),
\end{aligned}$$

where $N = 2^{|\mathcal{I}|}$. Equivalently for any $\delta > 0$, with probability at least $1 - \delta$,

$$\max_{A \subseteq \mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| \leq \sqrt{\frac{\log(2N) + \log(1/\delta)}{2n}} =: \Delta_2.$$

Note that the above bound Δ_2 is very similar to the result in Section 3.4, [3]. Now the bound Δ_2 can generate a new stopping condition for an approximation algorithm.

Recall the classical upper bound given in Section 3.3.

$$\Delta_1 := \frac{2\ell}{n}\sqrt{2\log N} + \sqrt{\frac{2\log(2/\delta)}{n}}.$$

Clearly $\lim_{\delta \to 0} \Delta_1/\Delta_2 = 2$, i.e., when δ is very small, the bound Δ_1 is roughly twice of Δ_2 given the sample size n. This assures us that the bound Δ_2 is highly competitive.

Theorem 3 can give another upper bound on the worst approximation error. However, since the number of terms in the summation grows exponentially on $|\mathcal{I}|$, to find the minimum is computationally infeasible. Furthermore, even if the minimum $w(s^*)$ is found, let Δ_1' be the upper bound of this variant defined as

$$\Delta_1' := w(s^*) + \sqrt{\frac{2\log(2/\delta)}{n}}.$$

By fixing the sample $\mathcal{S}$, we still have $\lim_{\delta \to 0} \Delta_1'/\Delta_2 = 2$. For small δ, the bound without Rademacher average still outperforms the existing ones.

5 Our Frequent Itemset Approximation Algorithm

5.1 Approximating with Precomputed Sample Size

We observe the bound given in the previous section:

$$\Delta_2 := \sqrt{\frac{\log(2N) + \log(1/\delta)}{2n}},$$

where $N = 2^{|\mathcal{I}|}$. The upper bound Δ_2 can be treated as a function of allowed error probability δ, sample size n and $N = 2^{|\mathcal{I}|}$, all of which are already given. A good news is that there is no parameter that needs to be progressively computed (e.g., ℓ in Δ_1). Thus to guarantee an worst approximation error at most ϵ, we only need to make sure $\Delta_2 \leq \epsilon$. By solving it we have

$$n \geq \frac{1}{2\epsilon^2}(\log(2N) + \log(1/\delta)).$$

Note that this sampling bound agrees with Toivonen's result (Corollary 2 in [23]).

Hence an (ϵ, δ)-approximation algorithm takes a very simple form. We first consider a brute-force algorithm to approximate frequencies for *all* the itemsets. Note that since the number of subsets (itemsets) in $\mathcal{I}$ is exponential (i.e., $2^{|\mathcal{I}|}$), the brute-force algorithm is not efficient.

Frequent Itemsets Approximation Algorithm (brute-force for all itemsets)

Input: items $\mathcal{I}$, transactional dataset $\mathcal{D} \subseteq 2^{\mathcal{I}}$, $\epsilon > 0$, $\delta > 0$.

Output: approximated frequencies $\hat{f}_{\mathcal{D}}(A)$ for each $A \subseteq \mathcal{I}$ s.t. with probability at least $1 - \delta$, $|\hat{f}_{\mathcal{D}}(A)| \leq \epsilon$ for any $A \subseteq \mathcal{I}$.

1. $n \leftarrow \lceil \frac{1}{2\epsilon^2}(\log(2^{|\mathcal{I}|+1}) + \log(1/\delta)) \rceil$.
2. $\mathcal{S} \leftarrow \emptyset$.
3. If $n \geq |\mathcal{D}|$, $\mathcal{S} \leftarrow \mathcal{D}$; otherwise, choose n itemsets in $\mathcal{D}$ at uniformly random and add them to $\mathcal{S}$.
4. Label the transactions in $\mathcal{S}$: $\mathcal{S} = \{\tau_1, \cdots, \tau_n\}$.
5. For each $A \subseteq \mathcal{I}$, compute $\hat{f}_{\mathcal{D}}(A) \leftarrow \frac{1}{n} \sum_{i=1}^{n} \phi_A(\tau_i)$.
6. Return all the $\hat{f}_{\mathcal{D}}(A)$'s for $A \subseteq \mathcal{I}$.

Since the brute-force algorithm above is computationally infeasible when $|\mathcal{I}|$ is large, in practice, we often only consider the frequencies of a few itemsets, e.g., most popular pairs of complementary goods, influential coauthoring in a community, etc. For this case, we do not have to consider the itemsets, which do not appear frequently enough. Denote by **Ob** the set of the itemsets to be observed. Then the worst approximation error is defined as the maximum error on every itemset in **Ob**. By the same reasoning in the derivation of Δ_2, we have the adjusted new bound:

$$\Delta_2' := \sqrt{\frac{\log(2|\mathbf{Ob}|) + \log(1/\delta)}{2n}}.$$

Since **Ob** is a subset of $2^{\mathcal{I}}$, this bound Δ_2' is tighter than Δ_2. Note that Toivonen (Corollary 2, [23]) also found a similar result as our bound here. The approximation algorithm will also be revised as follows.

Frequent Itemsets Approximation Algorithm with Observed Itemsets

Input: all the items $\mathcal{I}$, the observed itemsets **Ob** $\subseteq 2^{\mathcal{I}}$, transactional dataset $\mathcal{D} \subseteq 2^{\mathcal{I}}$, $\epsilon > 0$, $\delta > 0$.

Output: approximated frequencies $\hat{f}_{\mathcal{D}}(A)$ for each $A \in$ **Ob** s.t. with probability at least $1 - \delta$, $|\hat{f}_{\mathcal{D}}(A)| \leq \epsilon$ for any $A \in$ **Ob**.

1. $n \leftarrow \lceil \frac{1}{2\epsilon^2}(\log(2|\mathbf{Ob}|) + \log(1/\delta)) \rceil$.
2. $\mathcal{S} \leftarrow \emptyset$.
3. If $n \geq |\mathcal{D}|$, $\mathcal{S} \leftarrow \mathcal{D}$; otherwise, choose n itemsets in $\mathcal{D}$ at uniformly random and add them to $\mathcal{S}$.
4. Label the transactions in $\mathcal{S}$: $\mathcal{S} = \{\tau_1, \cdots, \tau_n\}$.
5. For each $A \in$ **Ob**, compute $\hat{f}_{\mathcal{D}}(A) \leftarrow \frac{1}{n} \sum_{i=1}^{n} \phi_A(\tau_i)$.
6. Return all the $\hat{f}_{\mathcal{D}}(A)$'s for $A \in$ **Ob**.

Note that we do not have to estimate for any itemset which is out of the observed ones **Ob**. Also we need the size of **Ob** to be as small as possible. Depending on the practical requirements, the choice of **Ob** can vary a lot. We will give a SON-based idea in the next section. However many other methods can be tried, e.g., most potentially frequent itemsets can be suggested by the users' experience or historic records.

5.2 Approximating Top-k Frequent Itemsets

In practice we often need to find out the top-k frequent itemsets among the given candidates **Ob**. We can slightly revise the algorithm given in the previous section to approximate the k most frequent itemsets. A new problem here is how to give the stopping condition. Note that if we only need the top-k frequent itemsets, then our approximation can stop when the members of top-k FIs are fixed with high probability (i.e., at least $1-\delta$). In particular, if with probability at least $1-\delta$, the true frequency of any itemset will not surpass the middle point of the k-th and $(k+1)$-th largest approximated frequencies, the k itemsets with largest approximated frequencies should probably be the correct top k ones. By Hoeffding's inequality and union bounds, given the approximate frequencies with n samples, the probability p that there exists an itemset, whose approximated frequency and true frequency are on the different sides of the middle point of the k-th and $(k+1)$-th largest approximated frequencies, can be upper bounded as follows:

$$\begin{aligned} p &= \Pr[\bigvee_{A\in\mathbf{Ob}} (f_{\mathcal{D}}(A) < m < \hat{f}_{\mathcal{D}}(A)) \vee (f_{\mathcal{D}}(A) > m > \hat{f}_{\mathcal{D}}(A))] \\ &\leq \sum_{A\in\mathbf{Ob}} \exp\left(-2n(\hat{f}_{\mathcal{D}}(A) - m)^2\right), \end{aligned}$$

where m is the frequency middle point as described above. Hence we can let the sampling stop when the upper bound of p is less than δ. Combining these ideas, a progressive sampling approximation algorithm can be given as follows:

Top-k Frequent Itemsets Approximation Algorithm with Observed Itemsets

Input: all the items $\mathcal{I}$, the observed itemsets $\mathbf{Ob} \subseteq 2^{\mathcal{I}}$, transactional dataset $\mathcal{D} \subseteq 2^{\mathcal{I}}$, sampling increase Δn, $k > 0$, $\epsilon > 0$, $\delta > 0$.

Output: k itemsets among **Ob** that have the highest approximate frequencies s.t. with probability at least $1-\delta$, the approximate frequencies have worst-case error less than ϵ.

1. $n \leftarrow 0$. $\hat{f}_{\mathcal{D}}(A) \leftarrow 0$ for any $A \in \mathbf{Ob}$.
2. $\mathcal{S} \leftarrow \emptyset$, $N \leftarrow \min\{|\mathcal{D}|, \lceil \frac{1}{2\epsilon^2}(\log(2|\mathbf{Ob}|) + \log(1/\delta))\rceil\}$.
3. Choose Δn itemsets from $\mathcal{D}$ at uniformly random, and denote by $\Delta\mathcal{S}$ the chosen Δn itemsets.
4. Label the transactions in $\Delta\mathcal{S}$: $\Delta\mathcal{S} = \{\tau_1, \cdots, \tau_{\Delta n}\}$.
5. For each $A \in \mathbf{Ob}$, compute $\hat{f}_{\mathcal{D}}(A) \leftarrow \frac{1}{n+\Delta n}\left(n \cdot \hat{f}_{\mathcal{D}}(A) + \sum_{i=1}^{\Delta n} \phi_A(\tau_i)\right)$.
6. $n \leftarrow n + \Delta n$. $\mathcal{S} \leftarrow \mathcal{S} \cup \Delta\mathcal{S}$
7. Let $\hat{f}[k]$ and $\hat{f}[k+1]$ be the k-th and $(k+1)$-th largest approximate frequencies in **Ob** respectively. Compute their middle point $m \leftarrow \frac{1}{2}(\hat{f}[k] + \hat{f}[k+1])$.
8. If $n > N$, then return the approximate top-k frequent itemsets $\hat{f}_{\mathcal{D}}(A)$ for $A \in \mathbf{Ob}$. Otherwise, go to the next step.
9. If the following stopping condition is satisfied:

$$\sum_{A\in\mathbf{Ob}} \exp\left(-2n(\hat{f}_{\mathcal{D}}(A) - m)^2\right) < \delta,$$

then return the approximate top k frequent itemsets in **Ob**. Otherwise, go back to Step 3.

Note that in out top-k approximation algorithm, the stopping condition depends on the k-th and $(k+1)$-th largest frequencies. If these two frequencies tie, it is likely that many samples will be needed since we cannot distinguish them based on approximated frequencies. Hence we require the number of samples should

not exceed N, the number of samples that can guarantee the (ϵ, δ)-approximation. If more than N samples are needed, we can assume that the k-th and $(k+1)$-th largest frequencies tie or are very close. Then we directly output the approximated top FIs, since further computations to distinguish the very close FIs on the boundary are often unnecessary in practice.

To be efficient, we must ensure the size of **Ob** is small enough. One possible way, which is similar to A-Priori algorithm [1], is given as follows:

1. We first only consider the itemsets with single item. The item size is usually small enough (i.e., $|\mathcal{I}|$) that can be put in main memory. We approximate their frequencies, and take the threshold T as the k-th largest frequency among the single items. Usually people are only interested in small k, e.g., 10 to 100, which is much less than $|\mathcal{I}|$.

2. Then we use SON algorithm [21] to exactly find the itemsets with frequencies at least T. For efficiency, in SON we only consider the itemsets with 2 items, like what [10] did. The reason is that the itemsets of sizes larger than 2 usually have much lower frequencies than pairs. Clearly SON algorithm can find at most k^2 candidate itemsets.

3. We use our (top-k) approximation algorithms to estimate the frequencies of the candidate itemsets (or select the top-k ones).

Since we only consider frequent pairs, we can also build our scheme on PCY or Multistage algorithms. The major difference between PCY, Multistage and A-Priori is how to fully use the main memory for the passes, in which we select the most frequent items or itemsets. Hence the difference does not affect our sampling. Also we can use Toivonen's algorithm instead of SON to mine frequent itemsets with more than 2 items.

6 Evaluation

In this section we present our evaluation results. We try to find the top-K frequent itemsets (in pairs) by two algorithms proposed in this paper:

- **A-Priori + Precomputed Sample Size**. We first use A-Priori algorithm to find the top-K most frequent items, and then use the algorithm (discussed in Section 5.1) to approximate the frequencies of all the pairs formed by the K items. At last we sort the frequencies and find the top-K pairs with the highest approximate frequencies. Note that the sample size can be precomputed.

- **A-Priori + Progressive Sampling**. This is given in Section 5.2. Note that the sampling is progressive, i.e., there is no precomputed sample size.

We compare our algorithms with the state-of-art [18], which is a progressive sampling approximation. For each comparison, our code for each algorithm is similarly organized except that the bounds are different. For the performance, we consider time complexity (running time), sample size, and precision/recall.

6.1 Setup

We implemented our algorithms by Python 3.4.3 and ran the programs on knot cluster (one DL580 nodes with 4 Intel X7550 eight core processors and 512GB RAM) at UCSB Center for Scientific Computing. To reduce the entire running time in the experiments (it is very time consuming to compute the exact solutions), for each dataset, we selected the first 70 items and then approximate the top 10 frequencies of their pairs. Each approximation was repeated by 10 times and we took the averaged results. By default we chose that the sample size increases by 100 for each round, and $\epsilon = 0.05$, $\delta = 0.01\%$.

Name	No. of Transactions	No. of items
accidents	340183	468
chess	3196	75
connect	67557	129
kosarak	990002	41270
mushroom	8124	119
pumsb	49046	7116
pumsb star	49046	7116
retail	88162	16470

Table 1: Dataset characteristics

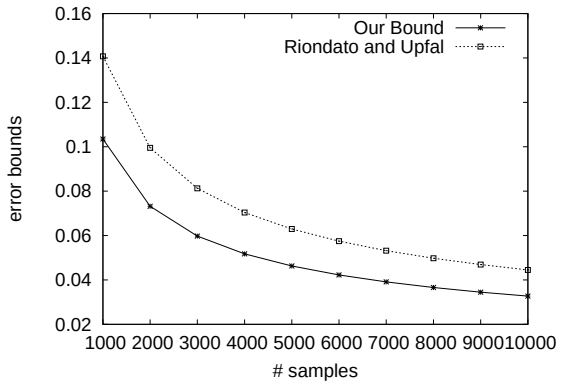

Figure 1: $\epsilon = 0.01$, $\delta = 0.0001$.

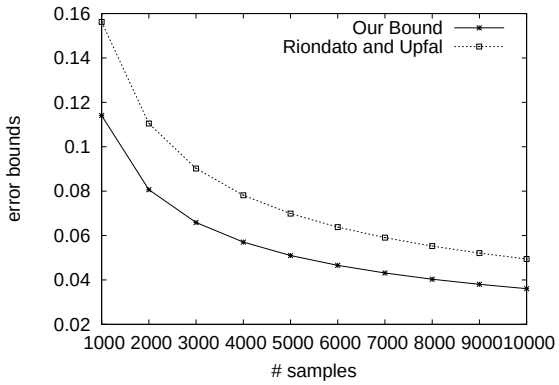

Figure 2: $\epsilon = 0.001$, $\delta = 0.00001$.

6.2 Datasets

For consistency, we choose FIMI'03 data repository [7], the real-world data set from [18] (the data repository is available at `http://fimi.ua.ac.be/data/`). The item and transaction data sizes of the FIMI datasets are given in Table 1. We will use these data sets to evaluate the samples sizes and worst case errors of our algorithms, and compare our results with the state-of-art algorithms.

6.3 Worst-case Error Upper Bound Comparison

We compare our new worst-case error bound with the state-of-art [18]. Figure 1 and Figure 2 give our bounds on worst-case errors and [18]'s for different combinations of ϵ, δ and sample size. Clearly our new bound outperforms the state-of-art, implying that to achieve a certain degree of accuracy, compared to [18]'s estimation, actually much fewer samples are needed. This key result gives the basic motivation of our algorithms.

6.4 Comparison Results

We compare the performance between our methods and the state-of-art [18]. Table 2 gives the evaluation results for our algorithms discussed in Section 5.1. The algorithms try to find the top-100 most frequent itemsets with two items. Our algorithm will first find the 100 most frequent items, and then approximate the frequencies of the pairs between the 100 items. Thus the number of the observed itemsets is $|\mathbf{Ob}| = 100 * 99/2 = 4950$ for the second pass. Given the default ϵ and δ, the precomputed sample size n for the second pass is fixed, except the dataset chess, whose transactional size is less than the precomputed sample size. Note that we have significantly confined the observed itemsets, and thus it is efficient to compute the

Item	Used time (sec) (ours)	Sample size (ours)	Precision (ours)	Used time (sec) [18]	Sample size [18]	Precision [18]
accidents	48.78	6894	97%	2683.74	387782	99%
chess	21.44	3196	100%	21.86	3196	100%
connect	50.34	6636	98%	1097.70	132751	99%
kosarak	9.17	3539	91%	1254.18	502924	99%
mushroom	43.32	6620	98%	125.56	18646	99%
pumsb	193.39	7438	98%	2998.37	115939	98%
pumsb star	155.15	7438	95%	2595.17	133668	99%
retail	102.93	7606	91%	1909.02	128544	96%

Table 2: Results of our approximation algorithms with precomputed sample size and [18].

Item	Used time (sec) (ours)	Sample size (ours)	Precision (ours)	Used time (sec) [18]	Sample size [18]	Precision [18]
accidents	18.69	3600	98%	357.37	60600	99%
chess	21.96	3200	98%	22.33	3200	99%
connect	43.04	6700	99%	748.37	110200	100%
kosarak	21.54	3600	89%	168.22	32400	96%
mushroom	46.13	6700	98%	100.68	16400	98%
pumsb	170.12	7500	98%	2225.19	98200	98%
pumsb star	130.49	7500	98%	1708.80	98200	98%
retail	98.24	7700	94%	1152.59	83700	97%

Table 3: Comparisons on our progressive approximation algorithms and [18].

upper bound in [18], i.e., we do not need to consider each subset of $\mathcal{I}$. In the table we have included all the samples taken in both the first and second passes. By using our new error upper bound, the running time and sample size are significantly reduced compared to [18], while the accuracy is still quite competitive, i.e., all are larger than 90%. This is natural since [18] takes more samples and thus the estimation should be more accurate. In the dataset chess, every transaction is sampled since its data volume is very small. In the dataset connect, our sample size is only 1/20 of [18]'s, but the accuracy is almost the same.

Table 3 gives the evaluation results for our algorithms discussed in Section 5.2. Similarly to the above non-progressive version, our progressive method is still quite efficient and accurate. The comparison between ours and [18] show that our method is competitive.

7 Conclusion

We have proposed a new upper bound for the worst-case errors on sampling-based approximate frequent itemsets mining. Our new bound is tighter than the state-of-art result. Based on our new bound, two approximation algorithms have been proposed. We have used real-world datasets to evaluate our results and the performance of our algorithms. The evaluation results have shown that our algorithms are not only competitive but also efficient.

References

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

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

[3] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer, 2004.

[4] Bin Chen, Peter Haas, and Peter Scheuermann. A new two-phase sampling based algorithm for discovering association rules. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–468. ACM, 2002.

[5] Kun-Ta Chuang, Ming-Syan Chen, and Wen-Chieh Yang. Progressive sampling for association rules based on sampling error estimation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 505–515. Springer, 2005.

[6] Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, and Jeffrey D Ullman. Computing iceberg queries efficiently. In *Internaational Conference on Very Large Databases (VLDB'98), New York, August 1998*. Stanford InfoLab, 1999.

[7] Bart Goethals and Mohammed J Zaki. Advances in frequent itemset mining implementations: report on fimi'03. *ACM SIGKDD Explorations Newsletter*, 6(1):109–117, 2004.

[8] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[9] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[10] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[11] Pascal Massart. Some applications of concentration inequalities to statistics. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 9, pages 245–303, 2000.

[12] Jong Soo Park, Ming-Syan Chen, and Philip S Yu. *An effective hash-based algorithm for mining association rules*, volume 24. ACM, 1995.

[13] Srinivasan Parthasarathy. Efficient progressive sampling for association rules. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 354–361. IEEE, 2002.

[14] Andrea Pietracaprina, Matteo Riondato, Eli Upfal, and Fabio Vandin. Mining top-k frequent itemsets through progressive sampling. *Data Mining and Knowledge Discovery*, 21(2):310–326, 2010.

[15] Matteo Riondato, Justin A DeBrabant, Rodrigo Fonseca, and Eli Upfal. Parma: a parallel randomized algorithm for approximate association rules mining in mapreduce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 85–94. ACM, 2012.

[16] Matteo Riondato and Eli Upfal. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 25–41. Springer, 2012.

[17] Matteo Riondato and Eli Upfal. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(4):20, 2014.

[18] Matteo Riondato and Eli Upfal. Mining frequent itemsets through progressive sampling with rademacher averages. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM, 2015.

[19] Matteo Riondato and Eli Upfal. Abra: Approximating betweenness centrality in static and dynamic graphs with rademacher averages. *arXiv preprint arXiv:1602.05866*, 2016.

[20] Matteo Riondato and Fabio Vandin. Finding the true frequent itemsets. In *SDM*, pages 497–505. SIAM, 2014.

[21] Ashok Savasere, Edward Robert Omiecinski, and Shamkant B Navathe. An efficient algorithm for mining association rules in large databases. *College of Computing Technical Report, Georgia Institute of Technology*, 1995.

[22] Tobias Scheffer and Stefan Wrobel. Finding the most interesting patterns in a database quickly by using sequential sampling. *Journal of Machine Learning Research*, 3(Dec):833–862, 2002.

[23] Hannu Toivonen et al. Sampling large databases for association rules. In *VLDB*, volume 96, pages 134–145, 1996.

[24] Vladimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[25] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.