Sampling for Top-K Frequent Itemsets

Santosh Ravi Kiran P
santosh.kiran@csa.iisc.ernet.in
Dept. of Computer Science and Automation, IISc.
SR No: 04-04-00-10-41-12-1-09183

Abstract

Sampling is an efficient alternative to mine top-K frequent itemsets on massive datasets. Recent research [13] has presented probabilistic bound on number of random samples required to get approximate solutions. The bound is linearly dependent on number of items present in dataset to be mined. Although bound is independent of dataset row-cardinality, it is dependent on maximum length of itemsets (let it be w) of interest. Users may find it difficult to come up with right w.

Our experimental results show that we can get approximate solutions with smaller number of samples (atleast an order of magnitude smaller) than the value given by the bound. We derived an alternative bound that uses the upper bound on number of "negative border" itemsets and made it independent of maximum length of itemsets w. Our new bound is logarithmic in number of items present in dataset. As the bound still remained loose, we explored the possibility of obtaining better bounds by assuming the knowledge of dataset statistics. Specifically, we made strong assumption of having count of "maximal frequent" itemsets, but the advantage is marginal.

This suggests our limitations of using theoretical tools in deriving better bounds, which motivated us to develop a voting based iterative sampling algorithm ItopK. Our experimental results show that it produces a relative approximate solution with number of samples close to optimal, in most of the cases it is *within twice of optimal*.

1 Introduction

Many chain stores collect data out of their customer transactions, so that they can process it and extract useful information which is crucial in making business decisions like design of promotional pricing, product placement, etc. One such analysis is association rule mining [3] for finding correlations between existence of different items in a transaction. It has applications in web usage mining, intrusion detection, and bioinformatics. The aim of association rule mining is to find rules like "whenever a customer buys jam and butter, it is likely that he is going to buy bread". The first step of association rule mining is extracting all frequent itemsets followed by rule generation. Though rule generation is trivial, first step of mining frequent itemsets should be done efficiently, because extracting all frequent itemsets in a dataset involves searching all possible itemsets (combination of items, which is exponential in size). Apriori, the classical data mining algorithm for mining frequent itemsets was presented in [1], which exploits the properties like "All subsets of a frequent itemset are frequent" and "All supersets of infrequent itemsets are infrequent" to reduce search space. Eventually, a lot of research has happened on this problem, an extensive overview of it is presented in [6]. We are concerned with the problem of mining top-K frequent itemsets which is variant of mining frequent itemsets.

1.1 Top-K frequent itemsets Mining

For a given dataset D of transactions over set of items I, an itemset (non-empty subset of I) is considered to be frequent if it appears in atleast fixed fraction of transactions. We call this fraction threshold as support threshold (*min_support*). In classical data mining problem user specifies min_support, which determines the output size i.e., cardinality of set of all frequent itemsets. But to choose an appropriate min_support users should have knowledge about mining query and nature of dataset. Setting min_support is subtle: a small min_support might generate thousands of itemsets while a large one may not generate itemsets at all. Small min_support for one dataset might be a bigger one for other dataset. Coming up with right min_support is not an easy task, which is also emphasized in [14, 18]. To have better control on output size users tend to ask for top-K itemsets rather than spec-

ifying min_support. So we deal the problem of mining top-K frequent itemsets i.e., the itemsets whose frequency is atleast as much as K-th most frequent itemset.

1.2 Sampling for Top-K frequent itemsets mining

With technology advancements dataset sizes are increasing enormously in terms of number of transactions. For example, if we consider the Walmart retail chain their total sales per week across all stores are in order of 200 million [17]. Mining top-K frequent itemsets from these huge datasets is computationally intensive and time consuming, because existing algorithms to extract top-K frequent itemsets have to scan entire dataset more than once. Most of times entire dataset may not fit in main memory which increases disk accesses and becomes computationally infeasible. So we use sampling to approximate top-K frequent itemsets within the desired levels of confidence (which determines size of random sample required). In this report, whenever we refer to sample, we mean to refer a random sample with replacement, unless we mention it explicitly.

Sampling has other benefits: Firstly when data is owned by different entities, the financial cost of data acquisition depends on size of sample needed. Size of the sample determines the network delay when data is present at remote location, which affects the performance of data mining task. It has very important application in the field of *privacy-preserving data mining* [2], in which only small portion of perturbed data is given to third party data miner, to prevent privacy breaches of owned data.

Probabilistic bound on number of random samples required to get an absolute approximation (see Section 2.2 for formal definition) is given in [13]. Henceforth, we refer to this bound as the PRUV (abbreviated from name of authors) bound. Although PRUV bound is independent of dataset row-cardinality, it depends on user specified threshold w. Users may find it difficult to come up with right w, especially when they use top-K itemsets to generate association rules having longer consequents. We did experiments on large instances of real world datasets and found that it is loose by atleast an order of magnitude when compared to optimal sample size (which we computed by checking ϵ-closeness of samples of all possible sizes). These experiments motivated us to look into new techniques to reduce the bound on sample size and bring it close to optimal.

1.3 Our Contributions

- We derived an alternative bound that uses the upper bound (function of K and number of items present in dataset) on number of negative border (defined in Section 4.2) itemsets and made it independent of w, a restriction imposed in original bound.
- We derived a bound that assumes prior knowledge about the result i.e., count of maximal frequent itemsets (defined in Section 4.2), which helps to get a tighter upper bound (function of number of maximal frequent itemsets and number of items present in dataset) on number of negative border itemsets. Our experimental results show that this bound is several times more than optimal sample size. This shows the limitations of using theoretical tools in deriving better bounds.
- Absolute approximation is not good for sparse datasets i.e., users have to specify an appropriate ϵ (error threshold) depending upon nature of dataset. So, we also derived a bound for relative approximation (see Section 2.3 for formal definition) and encountered the same limitations on deriving better bounds.
- With this motivation, we developed an iterative sampling algorithm that produces relative approximation with close-to-ideal sample size.

Organization In Section 2 we formally define top-K frequent itemsets and definition of an absolute approximation of it. In Section 3 we describe the experimental setup which includes the procedure of generating large datasets and method of computing optimal sample size, using which the tightness of every bound is evaluated. Section 4 deals with absolute approximation, at first we present the existing upper bound on sample size required to get an approximation and its tightness evaluation followed by another bound assuming count of maximal frequent itemsets and its tightness evaluation. Section 5 highlights the importance of relative approximation and repeats similar analysis of Section 4 for it. In Section 6 iterative sampling algorithm is described, which produces approximate solution with close-to-ideal sample size. Section 7 describes related work. Finally we conclude the report in Section 8 and present some directions for future work.

Dataset Parameters	**Notation**
Input Dataset	D
Set of items present in dataset D	I
Cardinality of set of items	n
Set of possible itemsets of I	$\mathcal{P}(I)$
Mining Parameters	**Notation**
Number of top frequent itemsets desired by user	K
Tolerance threshold specified by user	ϵ
Probability of success specified by user	1 - δ
Restriction on max. length of itemsets of result	w
Terms used in the analysis	**Notation**
Frequency of itemset x with respect to dataset D	$f_D(x)$
Frequency of K-th most frequent itemset of $\mathcal{P}(I)$ in dataset D	$f_D^{(K)}$
Set of top-Kfrequent itemsets of $\mathcal{P}(I)$ in dataset D	F_D^K
Set of maximal frequent itemsets among F_D^K	M_D^K
Set of negative border itemsets of F_D^K	NB_D^K
Subset of itemsets of $\mathcal{P}(I)$ having max. length w	$U(I,w)$
Cardinality of set $U(I,w)$	m
Frequency of K-th most frequent itemset of $U(I,w)$ in dataset D	$f_D^{(K_w)}$
Set of top-K frequent itemsets of $U(I,w)$ in dataset D	$F_D^{K_w}$
Generic random sample of transactions	S
Frequency of itemset x with respect to sample S	$f_S(x)$
Frequency of K-th most frequent itemset of $\mathcal{P}(I)$ in sample S	$f_S^{(K)}$
Set of top-Kfrequent itemsets of $\mathcal{P}(I)$ in sample S	F_S^K
Set of maximal frequent itemsets among F_S^K	M_S^K
Set of negative border itemsets of F_S^K	NB_S^K
Frequency of K-th most frequent itemset of $U(I,w)$ in sample S	$f_S^{(K_w)}$
Set of top-K frequent itemsets of $U(I,w)$ in sample S	$F_S^{K_w}$

Table 1: Summary of Notations

2 Problem Formulation

In this section, we formally define top-K frequent itemsets and approximations of it. We also define a variant of it and its absolute ϵ-approximation that depends on w.

Notations used in upcoming sections of this report are summarized in table 1.

2.1 Top-K Frequent Itemsets

Consider a dataset D of transactions, where each transaction τ is an element of $\mathcal{P}(I)$, where $\mathcal{P}(I)$ is powerset over all items I. $\mathcal{P}(I)$ is usually represented as lattice structure (see fig. 3 for $\mathcal{P}(I)$ of 4 items). For any non-empty itemset $x \in \mathcal{P}(I)$, we denote its frequency with respect to dataset D with $f_D(x)$ i.e., fraction of transactions containing x. Let us assume that the itemsets of $\mathcal{P}(I)$ are arranged in increasing order of their dataset frequencies (ranging from 0 to 1). Set of top-K frequent itemsets can be defined as

$$\text{TOPK}(D,I,K) = \Big\{(x, f_D(x)) : x \in \mathcal{P}(I)\backslash\{\emptyset\}, f_D(x) \geq f_D^{(K)}\Big\}$$

where $f_D^{(K)}$ is frequency of K-th most frequent itemset in dataset D. Let F_D^K denotes TOPK(D,I,K). F_D^K can have K or more ordered pairs because there can be multiple itemsets with same frequency as that of K-th most frequent itemset.

Let $U(I,w)$ denote set of all itemsets having length at most w. Set of top-K frequent itemsets of length at most w can be defined as

$$\text{TOPK}(D,I,K,w) = \Big\{(x, f_D(x)) : x \in U(I,w), f_D(x) \geq f_D^{(K_w)}\Big\}$$

where $f_D^{(K_w)}$ is frequency of K-th most frequent itemset of $U(I,w)$ with respect to dataset D. Let $F_D^{K_w}$ denotes TOPK(D,I,K,w).

Subsequently whenever we say frequent itemsets, we are referring to top-K frequent itemsets unless we mention explicitly.

2.2 Absolute ϵ-approximation to Top-K frequent itemsets

Consider a sample S ($\subseteq D$) of transactions from dataset D. Let F_S^K denote set of top-K frequent itemsets TOPK(S,I,K) from S. For any itemset x, we denote its support with respect to sample S with $f_S(x)$

i.e., fraction of S's transactions containing x and $f_S^{(K)}$ denotes frequency of K-th most frequent itemset in sample S.

An absolute ϵ-approximation to top-K frequent itemsets was defined in [13] (Definition 2).

Definition 1. Let $\epsilon \in (0,1)$ be a real valued parameter. The set F_S^K containing ordered pairs $(x,f_S(x))$ is defined as absolute ϵ-approximation to TOPK(D,I,K), if the following conditions are satisfied

C1: for each $(x,\ f_S(x)) \in F_S^K$, $f_D(x) \geq f_D^{(K)} - \epsilon$;
C2: for each $(x,\ f_S(x)) \notin F_S^K$, $f_D(x) < f_D^{(K)} + \epsilon$;
C3: for each $(x,\ f_S(x)) \in F_S^K$, $|f_D(x) - f_S(x)| \leq \epsilon$.

where $x \in \mathcal{P}(I) \backslash \{\emptyset\}$.

Condition C1 (horizontal lined region of fig. 1) implies that F_S^K should exclude all itemsets whose dataset frequency $f_D(x)$ is less than $f_D^{(K)} - \epsilon$, condition C2 (vertical lined region of fig. 1) implies that F_S^K should include all itemsets whose dataset frequency is atleast $f_D^{(K)} + \epsilon$, and condition C3 implies that the sample frequency $f_S(x)$ of every itemset x in the approximation F_S^K must be in ϵ-range of its dataset frequency $f_D(x)$.

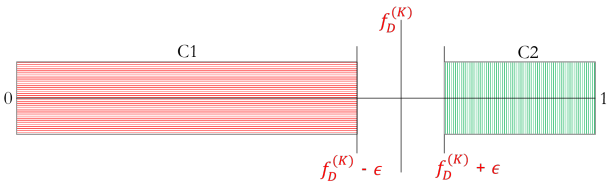

Figure 1: Absolute ϵ-approximation

Similarly we can define an absolute ϵ-approximation to TOPK(D,I,K,w). Let $F_S^{K_w}$ denotes set of top-K frequent itemsets TOPK(S,I,K,w) and $f_S^{(K_w)}$ denotes frequency of K-th most frequent itemset of $U(I,w)$ with respect to sample S.

Definition 2. The set $F_S^{K_w}$ containing ordered pairs $(x,f_S(x))$ is defined as absolute ϵ-approximation to TOPK(D,I,K,w), if the following conditions are satisfied

C1: for each $(x,\ f_S(x)) \in F_S^{K_w}$, $f_D(x) \geq f_D^{(K_w)} - \epsilon$;
C2: for each $(x,\ f_S(x)) \notin F_S^{K_w}$, $f_D(x) < f_D^{(K_w)} + \epsilon$;
C3: for each $(x,\ f_S(x)) \in F_S^{K_w}$, $|f_D(x) - f_S(x)| \leq \epsilon$.

where $x \in U(I,w)$.

2.3 Relative ϵ-approximation to Top-K frequent itemsets

Consider a sample S ($\subseteq D$) of transactions from dataset D. A relative ϵ-approximation to top-K frequent itemsets can be defined as

Definition 3. Let $\epsilon \in (0,1)$ be a real valued parameter. The set F_S^K containing ordered pairs $(x,f_S(x))$ is defined as relative ϵ-approximation to TOPK(D,I,K), if the following conditions are satisfied

C1: for each $(x,\ f_S(x)) \in F_S^K$, $f_D(x) \geq f_D^{(K)}(1-\epsilon)$;
C2: for each $(x,\ f_S(x)) \notin F_S^K$, $f_D(x) < f_D^{(K)}(1+\epsilon)$;
C3: for each $(x,\ f_S(x)) \in F_S^K$,
$|f_D(x) - f_S(x)| \leq \epsilon * f_D(x)$.

where $x \in \mathcal{P}(I) \backslash \{\emptyset\}$.

Condition C1 (horizontal lined region of fig. 2) implies that F_S^K should exclude all itemsets whose dataset frequency $f_D(x)$ is less than $f_D^{(K)}(1-\epsilon)$, condition C2 (vertical lined region of fig. 2) implies that F_S^K should include all itemsets whose dataset frequency is atleast $f_D^{(K)}(1+\epsilon)$, and condition C3 implies that the sample frequency $f_S(x)$ of every itemset x in the approximation F_S^K must be in $\epsilon * f_D(x)$ - range of its dataset frequency $f_D(x)$.

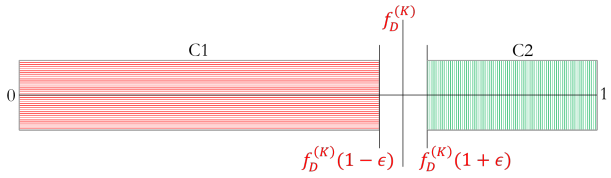

Figure 2: Relative ϵ-approximation

3 Experimental Setup

In upcoming sections we present different bounds, and evaluate their tightness by comparing them with optimal sample size. So, in this section we discuss about the datasets used in experiments followed by procedure to identify optimal sample size.

3.1 Datasets used in Experiments

We have used real-life datasets from FIMI repository [9]. The real power of sampling is realized only in case

of large datasets having millions of transactions. As original sizes of FIMI datasets are quite small, we generated scaled versions by keeping their statistical characteristics intact. We selected four datasets from FIMI repository with diverse characteristics. They are (a) RETAIL, a sparse dataset (b) MUSHROOM, in which all transactions have same length (c) PUMSB_STAR (d) KOSARAK, a sparse dataset. We scaled up all datasets to 100 million transactions. Details of datasets used in experiments are provided in table 2.

3.2 Generation of Large Datasets

Large datasets are generated using the procedure given in [11]. We generated them using random instances of original datasets, without affecting their distribution statistics (i.e., frequencies of every itemset should be same for original and larger instances). We did this efficiently by dividing original dataset into C partitions each containing X transactions (we set X to 10). Let GS be the number of transactions we want in large dataset. We generated GS/X array of random integers in the range [1, C], such that all values from 1 to C are uniformly distributed (balancing partition counts). Let $\pi(i)$ denote the ith value of above array. Starting from $i = 1$ to GS/X we read partition $P_{\pi(i)}$ from original dataset and write it to file of large dataset.

Dataset	**#Items**	**Avg. Trans. Length**	**#Trans.**
Retail	16,470	10	100,063,870
Pumsb_star	2,088	50	100,004,794
Kosarak	41,270	8	100,980,204
Mushroom	119	23	100,006,440

Table 2: Details of FIMI datasets

3.3 Computing Optimal Sample Size

PRUV bound on size of one-shot sample required to produce absolute ϵ-close solution was recent and there is no comprehensive study on its tightness. Empirically we compared its gap with the optimal one-shot sample S_{opt}. We ran fpgrowth algorithm [5] on entire dataset to obtain exact top-K frequent itemsets, which is used to verify whether a sample produces ϵ-close approximation or not.

Procedure to identify optimal sample size is given in [11]. Optimal sample size is found by evaluating all possible sample sizes and selecting the minimum sample size which produces ϵ-close solution. They did this by taking sequence of samples $S, S/2, S/2^2, S/2^3, ...$, with S being set to bound's value, and $S_{i+1} = S_i/2$ until we hit a sample $S/2^k$ which doesn't produce an ϵ-close solution. Then they do a binary search between $S/2^k$ and $S/2^{k-1}$ until they end up with an optimal sample S_{opt}. Let us call this algorithm as *halving algorithm*.

We have used a different approach that improves the efficiency of finding optimal sample size without affecting the end result. In our approach we start with a sample size of $S/2^p$, where p is maximum value for which $S/2^p < 50$, and double it in successive iterations (i.e., we considered sample sequence $S/2^p$, $S/2^{p-1}$, $S/2^{p-2}$, $S/2^{p-3}$, ...) until we hit a sample $S/2^{k-1}$ which produces ϵ-close solution. Then we do a binary search between last two sample sizes $S/2^k$ and $S/2^{k-1}$ to determine optimal sample size. Let us call this a *doubling algorithm*.

Both algorithms will converge to same k value, and the sample sizes of last two iterations considered for binary search will be same.

Optimal sample size is chosen in such a way that it produces approximation with 100% success probability. It is computed according to specific bound's definition (absolute or relative ϵ-approximation) and restriction (maximum length of itemsets w).

Performance analysis Total number of transactions taken in all samples during halving algorithm's execution is

$$S + \frac{S}{2} + \frac{S}{2^2} + ... + \frac{S}{2^{k-2}} + \frac{S}{2^{k-1}} + \frac{S}{2^k} = \frac{S}{2^k}\left(2^k + 2^{k-1} + 2^{k-2} + ... + 4 + 2 + 1\right) = \frac{S}{2^k}\left(2^{k+1} - 1\right)$$

Total number of transactions taken in all samples of our doubling algorithm's execution is

$$1 + 2 + 4 + ... + \frac{S}{2^{k+1}} + \frac{S}{2^k} + \frac{S}{2^{k-1}} = \frac{S}{2^{k-1}}\left(\frac{2^{k-1}}{S} + \frac{2^k}{S} + \frac{2^{k+1}}{S} + ... + \frac{1}{4} + \frac{1}{2} + 1\right) = \frac{S}{2^{k-2}}$$

The above sum is overestimated as we included sample sizes upto 1. The fraction of transactions drawn in doubling approach when compared to halving approach is

$$\frac{\frac{S}{2^{k-2}}}{\frac{S}{2^k}\left(2^{k+1}-1\right)} = \frac{4}{2^{k+1}-1}$$

Assuming that specific bound always gives ϵ-close solution, so $k \geq 1$. In worst case of doubling algorithm, where it takes samples up to bound's size (never happened in our experiments) then $k = 1$ and the fraction is 1.33 i.e., doubling algorithm takes 33% more samples than halving algorithm. Considering the looseness of PRUV bound, which is atleast an order of

magnitude larger than optimal sample size (see Section 4.1.2), k value will be atleast 3 and the fraction is less than or equal to 0.25 (i.e., doubling algorithm takes one fourth number of samples taken by halving algorithm), as we overestimated number of transactions taken in doubling algorithm. Although identification of optimal sample size is only for the purpose of comparison, it hugely affects our current experiments, because for a selected dataset we need to repeat this process for different combinations of K and w. We sampled multiple times, each time with different seed of pseudo random number generator to ensure accurate results.

3.4 Performance Metric

We use ratio of sample size given by specific bound to optimal sample size as performance metric.

$$\text{Competitive-factor} = S_{<bound-name>} \ / \ S_{opt}$$

where $S_{<bound-name>}$ represents upper bound on sample size computed for specific bound <bound-name> and S_{opt} denote optimal sample size required to get an ϵ-approximation.

4 Absolute ϵ-approximation

This section deals with bounds related to absolute ϵ-approximation. Firstly, we present an already existing bound and show that it is experimentally loose. Then we discuss about alternate bound (independent of maximum length of itemsets w) that uses the upper bound on number of negative border itemsets. Later we derive a bound which makes strong assumption of prior knowledge about count of maximal frequent itemsets and emphasize the limitations of theoretical tools on deriving better bounds.

4.1 PRUV bound

4.1.1 Upper bound on the sample size

Consider a sample S drawn at random from dataset D with replacement. An upper bound on sample size required to generate an absolute ϵ-approximation (Definition 2) to TOPK(D,I,K,w) is given by [13].

$$|S_{PRUV}| = \frac{2}{\epsilon^2} \ln \frac{2m+K(m-K)}{\delta}$$

where $m = |U(I,w)| = \sum_{i=1}^{w} \binom{n}{i}$ is cardinality of set of possible itemsets of length at most w, $n = |I|$. If we draw a sample of size $|S_{PRUV}|$, then TOPK(S_{PRUV},I,K,w) will be an absolute ϵ-approximation to TOPK(D,I,K,w) with probability atleast 1 - δ.

4.1.2 Tightness evaluation

We evaluated empirical tightness of PRUV bound on four real-life diverse datasets from FIMI repository. For different values of K and corresponding values of w, we computed optimal sample sizes for all four datasets. In all cases we observed that optimal sample size is atleast an order of magnitude less than PRUV bound.

We did experiments with ϵ set to 0.02 and δ set to 0.01. Tables 3 to 6 shows the experimental results for different datasets. Each cell in a table shows competitive-factor of PRUV bound for corresponding K (column) and w (row) values.

For higher values of w and smaller values of K, competitive-factor is close to two orders of magnitude, which shows the negative impact of w on the bound. Even at higher values of K and smaller values of w, it is atleast an order of magnitude.

K	1	5	10	50
$w = 2$	40	28	23	33
$w = 3$	46	43	41	41
$w = 5$	76	55	49	58
$w = 10$	145	111	98	102

Table 3: Retail Dataset Results

K	1	5	10	50	100
$w = 2$	36	24	15	16	14
$w = 3$	51	33	34	32	25
$w = 5$	99	41	32	39	44
$w = 10$	169	83	66	67	57

Table 4: Pumsb_star Dataset Results

K	1	5	10	50
$w = 2$	32	27	29	27
$w = 3$	44	45	33	45
$w = 5$	83	53	68	58
$w = 10$	132	94	107	111

Table 5: Kosarak Dataset Results

K	1	5	10	50	100
$w = 2$	378	150	39	11	12
$w = 3$	550	186	63	14	14
$w = 5$	774	322	76	17	23
$w = 10$	1154	608	127	33	36

Table 6: Mushroom Dataset Results

Overall, competitive-factor (see Section 3.4) for different datasets is atleast an order of magnitude ranging from 11 to 1154, which proves that PRUV bound is much larger than optimal sample size. This motivated us to work on lines of deriving better bounds.

4.2 Bound based on Negative Border

This section derives a new bound that uses the upper bound on number of negative border [16] itemsets, which in turn makes it independent of w, a parameter which has to be specified by user. It is not easy for user to come up with right w.

In this subsection, at first we define maximal frequent itemsets and negative border itemsets which are used in improving the bound.

Maximal frequent itemsets An itemset is said to be maximal frequent if no proper superset of it is frequent.

e.g. Consider a query of mining top-4 itemsets from dataset having four items (Milk, Bread, Butter and Eggs). These top-4 itemsets are Milk, Bread, Butter and Milk-Bread (see fig. 3). Butter and Milk-Bread (grid shaded region of fig. 3) are maximal frequent because none of their proper supersets are frequent.

Let M(D,I,K) be set of maximal frequent itemsets among top-K frequent itemsets with respect to D, such that no proper superset of it belongs to TOPK(D,I,K). Let M_D^K denotes M(D,I,K). Clearly $M_D^K \subseteq$TOPK(D,I,K). So,

$$|M_D^K| \leq |F_D^K| = K$$

Every itemset belonging to F_D^K, either belongs to M_D^K or a proper subset of atleast one itemset belonging to M_D^K. Hence,

$$\underset{x \in F_D^K}{\forall} f_D(x) \geq \min_{p \in M_D^K} f_D(p)$$

and

$$\underset{x \in F_S^K}{\forall} f_S(x) \geq \min_{p \in M_S^K} f_S(p)$$

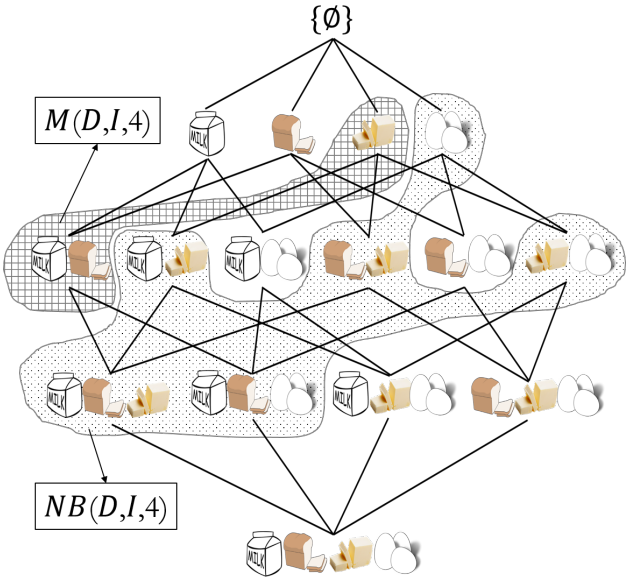

Figure 3: Maximal Frequent and Negative Border Itemsets

Negative border itemsets These are just infrequent itemsets i.e., minimal non-top-K itemsets. These are the itemsets which are minimal proper supersets of itemsets belonging to M(D,I,K), and they also include 1-items that does not belong to TOPK(D,I,K).

e.g. Dotted region of Figure 3 shows negative border itemsets for top-4 frequent itemsets.

Let NB(D,I,K) or NB_D^K denote set of negative border itemsets (NBI) of TOPK(D,I,K). Every infrequent itemset that does not belong to F_D^K, either belongs to NB_D^K or a proper superset of atleast one itemset belonging to NB_D^K. Therefore,

$$\underset{y \notin F_D^K}{\forall} f_D(y) \leq \max_{q \in NB_D^K} f_D(q)$$

and

$$\underset{y \notin F_S^K}{\forall} f_S(y) \leq \max_{q \in NB_S^K} f_S(q)$$

Size of negative border set depends on identities of maximal frequent itemsets. But every itemset of maximal frequent itemsets can have at most n minimal

proper supersets and there can be at most n 1-items which are infrequent. So, size of negative border can be upper bounded by $n|M_D^K| + n$.

$$|NB_D^K| \leq n(|M_D^K| + 1)$$

4.2.1 Absolute bound NBI^a

Derivation of PRUV bound (Section 3, Theorem 1 of [13]) uses the following facts.

Fact 1. Consider a sample S of transactions drawn at random with replacement from dataset D. For any fixed $\epsilon \in (0,1)$ and any non-empty itemsets $x, y \in \mathcal{P}(I)$ such that $f_D(x) \geq f_D(y) + \epsilon$ we have:

$$\Pr(f_S(y) > f_S(x)) \leq e^{\frac{-\epsilon^2}{2}|S|} \text{ and}$$
$$\Pr(|f_S(x) - f_D(x)| \geq \epsilon) \leq 2e^{\frac{-\epsilon^2}{2}|S|}$$

Let x be any frequent itemset ($\in$TOPK(D,I,K)) and y be any infrequent itemset ($\notin$TOPK(D,I,K)). PRUV bound's derivation is on the lines that if sample frequency of any y is greater than sample frequency of any x, then it is a failure. Number of such possible failures are $K(m-K) = K(2^n - 1 - K)$ (if we set w as n). Probability of each such failure can be upper bounded using the first inequality of above fact and probability that any such failure occurs can be upper bounded using union bound on $K(m-K)$ events.

From the definitions of maximal frequent itemsets and negative border itemsets, if sample frequency of every negative border itemset of D is less than sample frequency of every maximal frequent itemset of D, then sample frequency of every non frequent itemset of D is less than sample frequency of every frequent itemset of D. Formally,

$$if \quad \forall_{q \in NB_D^K} f_S(q) < \forall_{p \in M_D^K} f_S(p)$$
$$then \quad \forall_{y \notin F_D^K} f_S(y) < \forall_{x \in F_D^K} f_S(x)$$

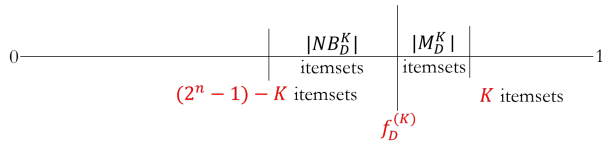

Figure 4: Maximal Frequent and Negative Border Itemset pairs

Above claim reduces the number of failure pairs from $K(2^n - 1 - K)$ to $|M_D^K| * |NB_D^K|$ (see fig. 4). Using the upper bounds of $|M_D^K|, |NB_D^K|$ from Section 4.2, number of failure pairs can be at most $nK(K+1)$. So, the term $K(m-K)$ of PRUV bound will get replaced by $nK(K+1)$. With high probability, we need to ensure that the event of Fact 1's second inequality doesn't happen for top-K itemsets. Therefore, the term $2m$ of PRUV bound will be replaced by $2K$. Hence the bound (lets call it NBI^a bound) becomes

$$|S_{NBI^a}| = \frac{2}{\epsilon^2} \ln \frac{2K + nK(K+1)}{\delta}$$

If we draw $|S_{NBI^a}|$ number of samples, then TOPK(S_{NBI^a},I,K) will be an absolute ϵ-approximation (Definition 1) to TOPK(D,I,K) with probability atleast 1 - δ. Clearly NBI^a bound is independent of w. Users may find it difficult to come up with right w, especially when they use top-K itemsets to generate association rules having longer consequents. Therefore, it is not a good idea restrict maximum length of itemsets of top-K.

4.2.2 Tightness evaluation

We used same experimental parameters (of Section 4.1.2) throughout the report. Tables 7 to 10 show competitive-factor of NBI^a bound for different values of K.

K	1	5	10	50
S_{NBI^a}/S_{opt}	21	25	24	24

Table 7: Retail Dataset Results

K	1	5	10	50	100
S_{NBI^a}/S_{opt}	29	17	21	20	17

Table 8: Pumsb_star Dataset Results

K	1	5	10	50
S_{NBI^a}/S_{opt}	23	20	24	26

Table 9: Kosarak Dataset Results

K	1	5	10	50	100
S_{NBI^a}/S_{opt}	261	148	62	15	13

Table 10: Mushroom Dataset Results

We can observe significant improvement in competitive-factor values when compared to original PRUV bound. Over all datasets competitive-factor is

atleast an order of magnitude ranging from 13 to 261. Worst case competitive-factor of PRUV bound is three orders of magnitude (1154), while it is two orders of magnitude (261) for NBI^a bound.

4.3 Bound assuming Maximal Frequent Itemsets

Even after incorporating the upper bound on number of negative border itemsets, the bound is loose by atleast an order of magnitude. So we worked on deriving better bounds assuming strong statistical knowledge of having count of Maximal Frequent Itemsets (MFI) of result, as did in [11].

4.3.1 MFI^a bound

As mentioned in Section 4.2.1, instead of applying union bound on $K(2^n - 1 - K)$ failure pairs, we can apply it for only $|M_D^K| * |NB_D^K|$ pairs. Using the upper bound of $|NB_D^K|$ from Section 4.2, number of pairs can be at most $n|M_D^K|(|M_D^K| + 1)$. Therefore the bound (call it MFI^a) becomes

$$|S_{MFI^a}| = \frac{2}{\epsilon^2} \ln \frac{2K + n|M_D^K|(|M_D^K|+1)}{\delta}$$

Here we used tighter upper bound on number of negative border itemsets, which assumes the count of maximal frequent itemsets.

4.3.2 Tightness evaluation

Tables 11 to 14 show competitive-factor of MFI^a bound for different values of K.

K	1	5	10	50
S_{MFI^a}/S_{opt}	21	23	22	22

Table 11: Retail Dataset Results

K	1	5	10	50	100
S_{MFI^a}/S_{opt}	29	16	19	17	13

Table 12: Pumsb_star Dataset Results

K	1	5	10	50
S_{MFI^a}/S_{opt}	23	18	22	23

Table 13: Kosarak Dataset Results

K	1	5	10	50	100
S_{MFI^a}/S_{opt}	261	129	49	11	10

Table 14: Mushroom Dataset Results

On the whole, competitive-factor for different datasets is atleast an order of magnitude ranging from 10 to 261. MFI^a bound performed marginally better than NBI^a bound in case of higher values of K. Results of MFI^a shows our limitations in using Chernoff's bounds in deriving better bounds even with knowledge of $|M_D^K|$ (which cannot be computed unless we mine entire dataset D). In next section, we show that same claim holds for relative ϵ-approximation also.

5 Relative ϵ-approximation

In case of sparse datasets, absolute ϵ-approximation doesn't work. For instance, consider a query of mining top-100 itemsets from Retail dataset with $\epsilon = 0.02$. Frequency of 100-th most frequent itemset is 0.0135. C2 of Definition 1 says that all itemsets whose frequency greater than or equal to $f_D^{(100)} + \epsilon = 0.0335$ should not be missed. Number of itemsets with frequency atleast 0.0335 are 24. So, absolute ϵ-close solution can only guarantee that 24 are true top-100 itemsets. This is because of low $f_D^{(100)}$ value of Retail dataset. This motivated us to work with relative ϵ-approximation also.

In this section, at first we show that sample size given by absolute bound cannot be used to generate relative ϵ-approximation. Then, we repeat the similar analysis of Section 4 for relative approximation.

5.1 Absolute bound for relative ϵ-approximation

For fixed ϵ, effective error (ϵ) allowed in absolute approximation is more than effective error ($\epsilon * f_D^{(K)}$) allowed in relative approximation. As the absolute bound is loose, it may indeed generate relative approximation. Experimentally we show that it cannot generate relative approximation in all the cases.

Tables 15 to 18 show the success probability of absolute bound in generating relative approximation for different datasets. Absolute bound values are computed for success probability of 99% ($\delta = 0.01$). Because of looseness, absolute bound is able to produce relative approximation at lower values of K. But for sparse datasets, at higher values of K, its success probability in producing relative approximation is very

low. This highlights the need of deriving bound for relative approximation.

K	1	5	10	50
Success Prob.(%)	100	100	93	0

Table 15: Retail Dataset Results

K	1	5	10	50	100
Success Prob.(%)	100	100	100	100	100

Table 16: Pumsb_star Dataset Results

K	1	5	10	50
Success Prob.(%)	100	100	94	11

Table 17: Kosarak Dataset Results

K	1	5	10	50	100
Success Prob.(%)	100	100	100	100	100

Table 18: Mushroom Dataset Results

5.2 Bound for relative ϵ-approximation

We start this subsection stating multiplicative Chernoff's bounds [4] used in deriving bounds for relative ϵ-approximation.

Multiplicative Chernoff's bounds Let Y_1,...,Y_r be i.i.d random variables. Define Y to be the sum of these r random variables. For any $0 < \alpha < 1$,

$$\Pr(Y \leq E[Y](1-\alpha)) \leq e^{-\frac{\alpha^2}{2}E[Y]} \text{ and}$$
$$\Pr(Y \geq E[Y](1+\alpha)) \leq e^{-\frac{\alpha^2}{3}E[Y]}$$

The following fact is used in analysis of upcoming subsection.

Fact 2. Consider a sample S of transactions drawn at random with replacement from dataset D. For any fixed $\epsilon \in (0,1)$ and any non-empty itemsets $x, y \in \mathcal{P}(I)$ such that $f_D(x) \geq f_D(y) + \epsilon * f_D^{(K)}$ we have:

$$\Pr(f_S(y) > f_S(x)) \leq e^{-\frac{\epsilon^2 * f_D^{(K)2}}{2}|S|} \text{ and}$$
$$\Pr(|f_S(x) - f_D(x)| \geq \epsilon * f_D(x)) \leq 2e^{-\frac{\epsilon^2}{3}f_D(x)|S|}$$

First inequality is obtained by substituting $\epsilon * f_D^{(K)}$ for ϵ in first inequality of Fact 1. Second inequality is obtained by taking $C_S(x)$ for Y in multiplicative Chernoff's bounds.

5.2.1 Relative bound NBI^r

Using the above two inequalities we can get an upper bound (call it NBI^r) on sample size required to get a relative ϵ-approximation to TOPK(D,I,K) as

$$|S_{NBI^r}| =$$
$$Max\left\{\frac{2}{\epsilon^2 * f_D^{(K)2}} ln\left(\frac{2nK(K+1)}{\delta}\right), \frac{3}{\epsilon^2 * f_D^{(K)}} ln\left(\frac{4K}{\delta}\right)\right\}$$

First term is obtained by applying union bound on $nK(K+1)$ failure pairs of first inequality and upper bounding the probability by $\delta/2$. Second term is obtained by applying union bound on K itemsets for second inequality and upper bounding the probability by $\delta/2$. Net failure probability is $\delta/2 + \delta/2 = \delta$.

If we draw $|S_{NBI^r}|$ number of random samples, then TOPK(S_{NBI^r},I,K) will be a relative ϵ-approximation to TOPK(D,I,K) with probability atleast 1 - δ.

NBI^r bound depends on $f_D^{(K)}$, which depends on dataset D. We use NBI^a bound to estimate this.

5.2.2 Two Phase Sampling

In phase-I $f_D^{(K)}$ is estimated, and in phase-II relative bound NBI^r is computed using this estimate.

Phase-I: Let ϵ, δ, n and K be input parameters to compute NBI^r bound. Let S_a be square root of NBI^a bound with ϵ^2, δ, n and K as input parameters. We draw $|S_a|$ number of random samples from D and compute $f_{S_a}^{(K)}$, which is used as estimate of $f_D^{(K)}$ in the next phase.

Phase-II: NBI^r is computed using $f_{S_a}^{(K)}$ of phase I as $f_D^{(K)}$. $|S_{NBI^r}|$ number of samples are required, so that TOPK(S_{NBI^r},I,K) will be a relative ϵ-approximation to TOPK(D,I,K) with success probability of $1 - \delta$.

5.2.3 Tightness evaluation

Tables 19 to 22 show competitive-factor of NBI^r bound for different values of K. Two phase sampling is used to compute NBI^r bound.

K	1	5	10	50
S_{NBI^r}/S_{opt}	20	25	80	258

Table 19: Retail Dataset Results

K	1	5	10	50	100
S_{NBI^r}/S_{opt}	30	27	21	25	27

Table 20: Pumsb_star Dataset Results

K	1	5	10	50
S_{NBI^r}/S_{opt}	24	38	95	155

Table 21: Kosarak Dataset Results

K	1	5	10	50	100
S_{NBI^r}/S_{opt}	385	212	66	20	14

Table 22: Mushroom Dataset Results

Overall, competitive-factor for different datasets is atleast an order of magnitude ranging from 14 to 385. NBI^r bound performed much worse than NBI^a, this is because the effective error ($\epsilon * f_D^{(K)}$) allowed in relative approximation is less than effective error (ϵ) allowed in absolute approximation. In case of Retail dataset, at $K = 50$, NBI^r crossed dataset size (100M). This is because of low value of dataset frequency of 50-th most frequent itemset.

5.3 Bound assuming maximal frequent itemsets

5.3.1 MFI^r bound

Using similar analysis of Section 4.3, we can derive maximal frequent itemsets bound (call it MFI^r) for relative ϵ-approximation as

$$|S_{MFI^r}| = Max\left\{\frac{2}{\epsilon^2 * {f_D^{(K)}}^2} ln\left(\frac{2n|M_D^K|(|M_D^K|+1)}{\delta}\right), \frac{3}{\epsilon^2 * f_D^{(K)}} ln\left(\frac{4K}{\delta}\right)\right\}$$

5.3.2 Tightness evaluation

Tables 23 to 26 show competitive-factor of MFI^r bound for different values of K.

K	1	5	10	50
S_{MFI^r}/S_{opt}	20	24	75	237

Table 23: Retail Dataset Results

K	1	5	10	50	100
S_{MFI^r}/S_{opt}	30	26	20	21	21

Table 24: Pumsb_star Dataset Results

K	1	5	10	50
S_{MFI^r}/S_{opt}	24	35	85	138

Table 25: Kosarak Dataset Results

K	1	5	10	50	100
S_{MFI^r}/S_{opt}	358	190	54	15	10

Table 26: Mushroom Dataset Results

On the whole, competitive-factor for different datasets is atleast an order of magnitude ranging from 10 to 358. At higher values of K, MFI^r bound performed marginally better than NBI^r bound. Results of MFI^r shows our limitations in using Chernoff's bounds in deriving better bounds even with knowledge of $|M_D^K|$ (which cannot be computed unless we mine entire dataset D). This motivated us to look for algorithmically obtaining a relative ϵ-close solution, which is discussed in next section.

6 The Itop$K$ Algorithm

In this section, we present an iterative sampling algorithm which empirically approaches to a right sample size that produces a relative ϵ-approximation to top-K itemsets. This algorithm is a variant of VISTA [11], which produces a relative ϵ-approximation to Frequent Itemsets Mining (min_support) problem.

6.1 Overview

It is an iterative sampling algorithm. In every iteration a batch of samples, S_{small} (called as *samplet*), are obtained. Every samplet of an iteration is mined for top-K frequent itemsets, based on which two sets L and F are updated. L contains set of candidate frequent itemsets i.e., itemsets that are frequent in at least one iteration. F contains set of clearly evident frequent itemsets i.e., top-K voted itemsets of L. The algorithm has two phases *Initialization phase* and *Stabilization phase*. It terminates when F *stabilizes*.

Initialization phase runs for $INIT$ (parameter) number of iterations. In every iteration L is updated. Whenever an itemset appears in an iteration, we check whether it is present in L or not. If it is present, its vote is incremented, otherwise it is added to L and its vote is initialized to 1. After initialization phase F is initialized to top-K voted itemsets of L. Before stabilization phase begins F is copied into F_{cur}. Stabilization phase runs until F_{cur} stabilizes. L and F are up-

dated in every iteration of stabilization phase. Whenever F_{cur} changes new *stabilization sequence* starts, it serves as representative set to measure the degree of change in F. It is updated only when there are significant changes to F (determined by parameter t). F_{cur} is the F of previous stabilization sequence. Stabilization phase terminates when F_{cur} doesn't change for fixed number of consecutive iterations (determined by parameter l_0). Samplet size S_{small} is determined by taking two times square root of NBI^r bound for given input parameters using two phase sampling. Pseudocode of complete algorithm is shown in Algorithm 1.

Algorithm Parameters Apart from input parameters (D, I, K, ϵ, δ), algorithm has three design parameters $INIT$, l_0 and t. $INIT$ determines the number of iterations for which L has to be warmed up. Stabilization phase ends when stabilization sequence length reaches l_0. In every iteration of stabilization phase difference between F (current set of clearly evident frequent itemsets) and F_{cur} is computed. F_{cur} is updated (new stabilization sequence starts) when the difference is more than fraction t of F_{cur} size.

6.2 Experimental evaluation of Itop$K$ Algorithm

Tables 27 to 30 shows number of samples (added over all iterations) taken by Itop$K$ algorithm and its performance when compared to S_{opt}.

K	1	5	10	50
S_{opt}	12,614	131,074	144,384	1,047,424
S_{ItopK}	18,810	79,156	122,854	833,404
S_{ItopK}/S_{opt}	1.49	0.6	0.85	0.8

Table 27: Retail Dataset Results

K	1	5	10	50	100
S_{opt}	4,082	5,995	8,318	10,247	11,389
S_{ItopK}	13,262	15,276	15,960	19,038	20,596
S_{ItopK}/S_{opt}	3.25	2.55	1.92	1.86	1.8

Table 28: Pumsb_star Dataset Results

K	1	5	10	50
S_{opt}	9,859	38,027	130,469	633,024
S_{ItopK}	18,696	43,852	127,984	393,110
S_{ItopK}/S_{opt}	1.95	1.15	1.04	0.62

Table 29: Kosarak Dataset Results

Algorithm 1: ItopK Algorithm

Data: Dataset D, I, ϵ, K, $INIT$, δ, stabilization length l_0 and tolerance t
Result: Relative ϵ-approximation to Top-K Frequent Itemsets
Extract a sample S_a of size $\sqrt{NBI^a(\epsilon^2, |I|, K, \delta)}$;
Compute $f_{S_a}^{(K)}$ from S_a;
Let $S_{small} = 2 * \sqrt{NBI^r(\epsilon^2, |I|, K, \delta, f_{S_a}^{(K)})}$
where NBI^a denotes improved absolute bound and NBI^r denotes relative bound using $f_{S_a}^{(K)}$ as approximation to $f_D^{(K)}$;
`// Initialization Phase`
Initialize L as empty set;
for $i \leftarrow$ *1 to INIT* **do**
  Obtain a samle S_i of size S_{small};
  Let L_i be top-K frequent itemsets of *samplet* S_i and update $L = L_i \bigcup L$;
end
Initialize F with set of top-K itemsets from L (with respect to number of votes);
`// Stabilization Phase`
Stab = false; curLen = 1; $F_{cur} = F$;
while *Stab == false* **do**
  Obtain a samplet of size S_{small} and let $L_{current}$ be top-K frequent itemsets of it;
  Increment the vote of each itemset in L that appears in $L_{current}$;
  Add the itemsets in $L_{current} \setminus L$ to L and initialize their vote to 1;
  F = Set of itemsets in L whose votes is more than K-th most voted itemset of L;
  $F_{diff} = F \setminus F_{cur}$;
  if $|F_{diff}| > t.|F_{cur}|$ **then**
    curLen = 1; `// New stabilization`
    $F_{cur} = F$; `// sequence starts`
  else
    curLen += 1;
  end
  if *curLen* $\geq l_0$ **then**
    Stab = true;
  end
end
return F_{cur};

K	1	5	10	50	100
S_{opt}	169	397	1,596	13,216	24,516
S_{ItopK}	9,690	10,982	12,312	19,152	23,000
S_{ItopK}/S_{opt}	57.34	25.67	7.71	1.45	0.94

Table 30: Mushroom Dataset Results

Experimental Parameters We set $INIT = 10$, $l_0 = 10$ and $t = 0.01$ for Itop$K$ algorithm, while ϵ and δ remain intact.

Observations Overall, competitive-factor of Itop$K$ algorithm ranges between 0.6 to 57.34, which shows that our algorithm outperformed the theoretical bound. It is always *within twice of optimal*, except in the cases where S_{opt} is very small. In cases where it is greater than two, absolute number of samples drawn by our algorithm is less than 10K number of samples (0.01% of dataset size). In some cases our algorithm performed better than optimal sample size, because our algorithm draws samples in batches while S_{opt} draws samples in one-shot and checks whether it produces an ϵ-close solution or not.

7 Related Work

In this section, whenever we mention frequent itemsets mining, we are referring to problem which takes min_support as input.

Rich body of literature exists on importance of sampling for extracting frequent itemsets. Sampling for frequent itemsets mining was first introduced in [10]. Along with sampling they dealt with many issues, their experimental investigation points out sampling as efficient technique, they did not emphasize on effectiveness of sampling for association rule mining.

Toivonen [16] is the first person to study sampling exclusively in the context of frequent itemsets mining. He gave a bound on sample size required to ensure that the frequency of itemset in sample is approximately equal to its true dataset frequency. Through empirical evaluation on synthetic dataset of size 100K transactions he showed that sample of size ranging from 20k to 80k produces accurate results.

An experimental evaluation of sampling for frequent itemsets was carried out in [19]. They argued that sample sizes computed using Chernoff bounds were loose. They showed that in some cases Chernoff bounds exceed dataset size, by considering an example dataset of 400K transactions with reasonable accuracy guarantee. With rigorous experimentation they gave a rule of thumb that "samples of sizes ranging from 10% to 20% of dataset are enough to give a reasonable approximation to frequent itemsets". But sampling 10% of dataset is expensive for massive datasets like that of Walmart chain store. In [7] authors claimed that sample of size 10% is larger (for massive datasets) when compared to Chernoff bounds.

A sampling based frequent itemsets mining algorithm, called FAST was presented in [8]. Given a sample size S, they concentrated on obtaining a representative sample over which frequent itemsets mining yields better results when compared to random sample of same size. Sub-sampling based heuristics were developed and used, in which a sub-sample of size S is extracted from a random sample of larger size, and then frequent itemsets mining is performed on extracted sub-sample. Through detailed experimentation, they evaluated accuracy of their heuristics, but they haven't came up with any theoretical upper bound on sample size or accuracy guarantees on output.

A progressive sampling algorithm in which samples were drawn with number of samples increasing geometrically over consecutive iterations was presented in [12]. In each iteration a representative set was selected from the most frequent 1-itemsets of the sample taken in current iteration. Algorithm's termination was based on similarity between representative sets obtained in last two iterations. Their empirical evaluation did not assess the accuracy of the final output.

In [11], the authors empirically showed that existing theoretical bounds on sample size required to mine frequent itemsets are loose (close to 2 orders of magnitude). Assuming prior knowledge of maximal frequent itemsets (frequent itemsets for which no superset is frequent), they derived new bound on sample size required to obtain relative ϵ-close solution (as defined in [7]) for frequent itemsets mining. They showed that even with such high detailed information of maximal frequent itemsets the sample size required is much larger when compared to optimal sample size (by factor of 5). They came up with an iterative sampling algorithm, called VISTA, on lines of progressive sampling algorithm, in which size of sample drawn at each iteration is same, with global convergence criterion, based on voting scheme and notion of stable sequence of iterations. Empirically they showed that sample size required by VISTA is always within the twice of optimal.

In the context of top-K frequent itemsets mining, attempts were made in designing algorithms to mine exact top-K frequent itemsets from entire dataset, restricting output to closed frequent itemsets (no proper superset of it will have same support as itself)[14, 18].

In [13], authors of [14] claimed that these algorithms have limited scalability and are not capable of handling massive datasets.

In [13] authors developed a progressive sampling algorithm, in which number of samples drawn in consecutive iterations increases. Their termination condition is dependent on the sample frequencies of the itemsets obtained from the sample of current iteration. This algorithm performs better than their bound only at higher values of w (restriction on maximum length of interested itemsets) and smaller values of K.

8 Conclusions

In this report, at first we emphasized on advantages of sampling for extracting top-K frequent itemsets, then we empirically evaluated the tightness of theoretical bounds and showed that the bounds are loose (atleast an order of magnitude) when compared to optimal.

We derived a bound that uses upper bound on number of negative border itemsets. We showed the limitations of using Chernoff's bounds in deriving better bounds by making stronger assumption of knowing count of maximal frequent itemsets apriori. In the end, we described an iterative sampling algorithm which produces ϵ-close solution with close-to-ideal sample sizes, specifically in most of the cases it is *within twice of optimal*.

In future, we shall try to give accuracy and confidence guarantees on output of Itop$K$ algorithm. We shall compare our approaches against the bounds given in recent work [15]. Future research could target sampling approaches for problems like classification, clustering etc.

References

[1] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases". In: VLDB '94 (1994), pp. 487–499.

[2] R. Agrawal and R. Srikant. "Privacy-preserving data mining". In: SIGMOD '00 (2000), pp. 439–450.

[3] R. Agrawal, I. Tomasz, and A. Swami. "Mining association rules between sets of items in large databases". In: SIGMOD '93 22.2 (1993), pp. 207–216.

[4] N. Alon and J. Spencer. "The Probabilistic Method". In: (1992).

[5] C. Borgelt. "An implementation of the FP-growth algorithm". In: OSDM '05 (2005), pp. 1–5.

[6] A. Ceglar and J. Roddick. "Association mining". In: ACM Computing Surveys 38.2 (2006).

[7] V. Chakaravarthy, V. Pandit, and Y. Sabharwal. "Analysis of sampling techniques for association rule mining". In: ICDT '09 (2009), pp. 276–283.

[8] B. Chen, P. Haas, and P. Scheuermann. "A new two-phase sampling based algorithm for discovering association rules". In: KDD '02 (2002), pp. 462–468.

[9] B. Goethals and M. Zaki. "Advances in frequent itemset mining implementations: report on FIMI'03". In: SIGKDD Explorations 6.1 (2004), pp. 109–117. URL: `http://fimi.ua.ac.be/data/` (visited on 09/06/2014).

[10] H. Mannila, H. Toivonen, and A. Verkamo. "Efficient Algorithms for Discovering Association Rules". In: (1994), pp. 181–192.

[11] V. Pandit, J. Haritsa, and J. Mudireddy. "VISTA: A View of Effective Sampling for Frequent Itemset Mining." In: Unpublished Draft (2014). URL: `http://dsl.serc.iisc.ernet.in/~course/DBMS/papers/vista.pdf` (visited on 09/06/2014).

[12] S. Parthasarathy. "Efficient Progressive Sampling for Association Rules". In: ICDM '02 (2002), pp. 354–361.

[13] A. Pietracaprina, M. Riondato, E. Upfal, and F. Vandin. "Mining top-K frequent itemsets through progressive sampling". In: Data Mining and Knowledge Discovery 21.2 (2010), pp. 310–326.

[14] A. Pietracaprina and F. Vandin. "Efficient incremental mining of top-K frequent closed itemsets". In: DS '07 (2007), pp. 275–280.

[15] M. Riondato and E. Upfal. "Efficient Discovery of Association Rules and Frequent Itemsets Through Sampling with Tight Performance Guarantees". In: ECML PKDD'12 (2012), pp. 25–41.

[16] H. Toivonen. "Sampling Large Databases for Association Rules". In: VLDB '96 (1996), pp. 134–145.

[17] *Walmart Corporate and Financial facts.* URL: `http://news.walmart.com/walmart-facts/corporate-financial-fact-sheet` (visited on 09/06/2014).

[18] J. Wang, J. Han, Y. Lu, and P. Tzvetkov. "TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets". In: IEEE Transactions on Knowledge and Data Engineering 17.5 (2005), pp. 652–664.

[19] M. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. "Evaluation of sampling for data mining of association rules". In: RIDE '97 (1997), pp. 42–50.