VISTA: A View of Effective Sampling for Frequent Itemset Mining

A Project Report

Submitted in partial fulfilment of the

requirements for the Degree of

Master of Engineering

in

Computer Science and Engineering

by

Mudireddy Jagadesh Babu

Computer Science and Automation

Indian Institute of Science

Bangalore – 560 012

July 2011

TO

My Parents

Acknowledgements

It is a pleasure to thank those who made this thesis possible. I owe my deepest gratitude to my advisor, Prof. Jayant R Haritsa. In spite of very busy schedule he did not hesitate to spend his valuable time and supported me throughout my research work with his patience and knowledge. I feel extremely fortunate to have him as my advisor. It had been a great experience to work with him.

Also, I would like to thank Vinayaka Pandit (IBM India Research Labs) for introducing me to this problem and constantly supporting me to solve this problem.

I dedicate this thesis to my parents who have been constantly supportive and source of inspiration. I want to express my gratitude to my friends due to which I had a wonderful stay in IISc.

I owe a lot to all the teachers, faculty from IISc, who have helped me and inspired me along the way. I am grateful to all the staff at the Department of Computer Science and Automation for their support and timely help.

Abstract

Sampling is a well established technique to speed up the process of discovering frequent itemsets. While the early literature focused on heuristic techniques, mathematical bounds on the sample size required to probabilistically achieve approximately correct results were recently presented in [6], [12]. A particularly appealing feature of these bounds is that they are independent of the database row-cardinality.

In this report, we demonstrate through an extensive empirical evaluation that the bounds, although theoretically elegant, are loose by as much as one to two orders of magnitude in practice. We therefore investigate the possibility of obtaining better bounds through prior knowledge of statistics on the datasets. In particular, we assume that the number of maximal frequent itemsets in the data mining result is known in advance. However, even with such a strong assumption, the revised bound turns out to be several multiples of the required sample size. This motivates us to consider the question of algorithmically identifying a reduced sample size that is sufficient to obtain accurate results. To address this issue, we present VISTA, a voting-based iterative sampling algorithm for accurately discovering frequent itemsets, whose sampling overheads are comparable to the ideal sample size for one-shot frequent itemset mining. VISTA incrementally mines samples in small batches and uses the presence or absence of a frequent itemset in each batch to determine its voting characteristics. The stopping condition is the reaching of a fix point in the identities of frequent itemsets that receive a clear majority of the votes across the batches. All results presented here are validated through extensive experimental evaluation on massive synthetic and real datasets.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recent advances in technology have enabled many organizations to collect massive amount of data from their businesses. These datasets can be seen as valuable data for their company as they can find unknown knowledge by mining these datasets. Mining association rules from datasets is an important research topic these days. Consider for example, the database of all the transactions that take place in a retail chain. The goal of association rule mining is to discover association rules of type, "whenever onion and potato are included together in a transaction, it is likely that burger is also included in the transaction". In association rule mining, main computational step was to find the frequent patterns from which association rules are derived. The problem of finding frequent patterns from database is called as Frequent Itemset Mining(FIM).

1.1 Frequent Itemset Mining

The problem of frequent itemset mining over the so-called basket data model was first introduced in [1]. Basket data consists of a set of individual records called transactions, and each transaction is comprised of a list of items that feature in the transaction. The computational goal is to identify all itemsets whose frequency, called support, in the database exceeds a user-defined threshold (θ), these results are relevant to a diverse suite of applications, including design of promotional pricing, product placement, web

usage mining, intrusion detection, and bioinformatics. Apriori, the classical algorithm for efficiently mining frequent itemsets, was proposed by Agrawal and Srikant in [2], and made use of the monotonicity property that all subsets of any frequent itemset must themselves be frequent. A substantial body of prior work deals with frequent itemset and association rule mining. We refer the reader to the survey by Ceglar and Roddick [5] for more details.

1.2 Sampling for Frequent Itemset Mining

All the frequent itemset mining algorithms used in the literature have drawback that, in addition to significant computations, they need to scan the entire database at least once. Considering how rapidly the sizes of the transactions are growing, it is desirable to avoid scanning the entire database. In this work, we consider various aspects of sampling as a technique for frequent itemset mining.

The use of sampling in identifying frequent itemsets has been popular since inception of association rule mining paradigm. A representative list of the initial work in this direction can be seen in [14], [10], [16], [7]. Their main focus was on experimentally studying the effectiveness of sampling at different sample sizes. A major limitation, however, was that they considered only small databases, whereas the true power and benefits of sampling come to the fore with very large databases. While sampling for FIM was first broached in [10], their work deals with many issues in addition to sampling and hence, their empirical investigation only points to the possible effectiveness of sampling. Toivonen [14] was the first one to study sampling as a technique in its own right. This study presented an upper bound on the sample size required to ensure that the support of a given itemset in the sample is approximately equal to its support in the database that is, its sample support is approximately equal to its database support. A detailed experimental evaluation of the proposed techniques on synthetic basket databases with 100K records showed that sample sizes from 20K to 80K provide very high accuracy.

An empirical evaluation of sampling for frequent itemset mining was also carried

out in [16]. They argued that the sample sizes suggested by Chernoff bounds [3] can be much larger than that required in practice for FIM. As an example, in the case of a database with 400K rows and a reasonable approximation guarantee, they showed that the Chernoff bounds turn out to be larger than the database size itself! Through extensive experimentation, they came up with a rule of thumb that, in order to obtain reasonable accuracy, samples of sizes between 10% to 25% of the database were required (depending on the desired support threshold). But, a crucial issue that was missed in the above exercise was the question of what happens when the databases are much larger?. As a case in point, consider massive databases like the one operated by Walmart, where just the weekly transactions across all their stores is of the order of 200 million [15]! In such environments, sampling even 10% of the database is itself prohibitively expensive.

A sampling based FIM algorithm, called FAST , was presented in [7]. Given a sample size S, they focussed on obtaining the most representative sample of this size over which FIM can yield better results as compared to a simple random sample of equivalent size. Sub-sampling based heuristics were devised, wherein a cleaned-up subsample of size S is first constructed from an original random sample of larger size, and then the frequent itemset mining is carried out over the sub-sample. However, no quality guarantees on the results were provided in their analysis. Further, experimental results on only a couple of small-sized datasets were presented in the study.

A progressive sampling algorithm in which the sample size increases geometrically between successive samples was presented in [11]. A local convergence criteria which compares the set of characteristic frequent itemsets between the most recent two samples is used to define the termination condition. An empirical evaluation of the convergence rate was presented for a specific notion of closeness between two sets of characteristic frequent itemsets. The local convergence criteria forces the algorithm to run for large number of iterations before reaching convergence. For instance, for a convergence threshold of 0.95 similarity between successive samples, the stabilization occurs at samples of size 10 million. Whereas, using a global voting based criteria, we show that we can reach termination with all the required accuracy guarantees at a sample size which is at

least one order of magnitude lesser. Another limitation of the work is that, there is no characterization of the errors when FIM is carried out on the computed sample.

1.3 Theoretical Bounds

As mentioned before, identifying the sample size required to ensure that, for a given itemset, its sample support approximately equals its database support, was evaluated in [14]. But, none of the early works considered the question of the sample size required to ensure that this property is satisfied for all the itemsets of interest, i.e. the frequent itemsets. Recently, this issue was addressed in [6] by formulating the notion of ϵ-close FIM solutions, defined below.

ϵ-close Solution [6]: Given a database of transactions T, a support threshold θ, and a tolerance threshold ϵ, a set (F_ϵ) consisting of pairs of the form (I, f), where I is an itemset and f is its frequency, is said to be an ϵ-close solution if it satisfies the following conditions:

- Each itemset I whose database support is θ or above is present in F_ϵ.
- If an itemset frequency in the database is α then its reported frequency(f) in F_ϵ is $(1-\epsilon)\alpha \leq f \leq (1+\epsilon)\alpha$. We call its frequency in F_ϵ as ϵ-close frequency.
- No itemset I with a database support lower than $(1-\epsilon)\theta$ is present in F_ϵ. We call itemsets with supports in this range as *Unacceptable False Positives*(UFP).
- No itemset I with a database support α is present in F_ϵ with $f < (1-\epsilon)\alpha$ or $f > (1+\epsilon)\alpha$. These itemsets with supports in this range are also called as *Unacceptable False Positives*.

Note that an ϵ-close solution may contain itemsets whose database support is in the range of $[(1-\epsilon)\theta, \theta)$. We refer to such itemsets as *acceptable false positives* (AFP).

Rather surprisingly, it was shown in [6] that the sample size required for guaranteeing ϵ-close solutions with high probability is independent of the size of the database

(measured in terms of its row cardinality)! Specifically, the sample size bound is given by

$$O(\frac{1}{\epsilon^2\theta}(\Delta + \log\frac{h}{\theta})) \tag{1.1}$$

where Δ is the length of the longest transaction occurring in the database, and $(1-\frac{1}{h})$ is the desired probability of success.

The above result is achieved through (i) a lemma that bounds the number of itemsets of a given support, and (ii) a technique for analyzing the probability of UFP itemsets erroneously appearing in the output. For ease of presentation, we will hereafter refer to this bound as the **longest transaction (LT)** bound.

A notion very similar to ϵ-close solutions was considered by [12] in the context of top-K itemset mining, where the goal is to identify the K itemsets having the highest supports from among all itemsets appearing in the database. They analytically show that a sample of size $O(\frac{1}{\epsilon^2}\log(h(2f + K(f - K))$ is sufficient to guarantee ϵ-close solutions. Here, f is the number of frequent itemsets, K is the number of top itemsets that are required, and $(1-\frac{1}{h})$ is the desired probability of success. In their study, the value of m is artificially curtailed by considering only itemsets within a predetermined size.

1.4 Contributions

The LT bound of [6] is appealing due to its being independent of the database cardinality. For databases hosting hundreds of billion transactions, the bound works out to be just a few million. We have conducted extensive experiments with popular real-life and synthetic datasets to study the tightness of this bound. Based on the experiments, our consistent observation is that there is a gap of atleast an order of magnitude, often even two, between the bounds and the ideal sample sizes, as would have been determined by an Oracle that directly runs Apriori on the sample database (we model the oracle by empiricially evaluating the fewest number of samples for which ϵ-close solutions are obtained).

Given the above observation, we first consider the question of whether it is possible to obtain tighter bounds by assuming advance knowledge of some strong statistical measures on the database. In particular, we derive a new bound that presupposes knowledge of the number of maximal frequent itemsets in the database at the given support threshold (a maximal frequent itemset is one for which no superset is frequent). This is a strong assumption as it implies the knowledge of the histogram of the number of maximal frequent itemsets at different support thresholds. We derive a new bound in terms of this parameter and show that a sample of size is sufficient to obtain ϵ-close solutions. Here, m is the number of items, $(1 - \frac{1}{h})$ is the desired probability of success, and M is the set of maximal itemsets arising with the θ support threshold. Our proof is based on simple observations that relate the supports of itemsets based on their relationships in the itemset lattice structure. We will hereafter refer to this bound as the maximal frequent itemset (MFI) bound.

For meaningful values of the input thresholds, the MFI bound yields a value that is substantially lower (typically, by a factor of between 5 and 10) than the LT bound. However, it still falls short in practice the gap between the MFI bound and the Oracle continues to be somewhat large (typically, as much as 5 times).

The conclusions drawn from the earlier work are also found to be not very useful. For instance, [16] suggests a sample size of at least 10% of the database, but this works out to be much larger than the LT and MFI bounds for industrial strength large databases. Therefore, there is a need to algorithmically arrive at the appropriate sample size for obtaining ϵ-close solutions. We present, in this paper, an iterative sampling algorithm, called VISTA (Voting-based Iterative Sampling of Transactions Algorithm), for this purpose. VISTA is based on the concept of repeated voting on the set of candidate frequent itemsets, based on small-sized samples at each step. Repeat yes votes on a given itemset add credence to its claim for being a frequent itemset, whereas sporadic votes raise the bar on the voting levels required in subsequent iterations. Conceptually, our technique is novel in the context of frequent itemset mining and may be of independent interest for

other sampling problems in data mining. We present an extensive empirical evaluation of VISTA over a rich suite of real and synthetic datasets. Our results indicate that VISTA uses much, much fewer samples than the theoretical bounds in fact, it is always within twice of the Oracle without sacrificing the data mining quality.

In a nutshell, VISTA is the first sampling algorithm, to our knowledge, that is (empirically) capable of delivering ϵ-close solutions while only incurring sampling overheads comparable to that of an idealized one-shot Oracle FIM algorithm.

1.5 Organization

The remainder of this thesis is organized as follows: The problem formulation is given in Chapter 2. Limitation of theoretical bounds are explained in Chapter 3 which also presents the new theoretical bounds which are lower than the LT bound. Chapter 4 presents Iterative Sampling Algorithm VISTA. Chapter 5 presents the Experimental setup and Experimental results are highlighted in chapter 6. Finally, Chapter 7 summarizes the conclusions of our study.

Chapter 2

Problem Formulation

In this chapter, we precisely define the problem that we address here and the notations used in our analysis. The input to frequent itemset mining (FIM) consists of a database T of N transactions, $T = t_1, t_2, \ldots t_N$, that range over a set of m items $\mathcal{I}$=$I_1, I_2, \ldots I_m$. Each transaction is a subset of $\mathcal{I}$ and we use Δ to denote the length of the longest transaction i.e. the transaction with the maximum number of items. We assume that Δ is part of the statistics maintained by the database. A subset $X \subseteq \mathcal{I}$ is called an itemset, and the frequency of an itemset is the ratio of the number of transactions that contain the itemset to the total number of transactions in the database. An itemset X is said to be p-frequent if its frequency is at least p. Given a user defined parameter $0 \geq \theta \geq 1$, called the support threshold, the FIM goal is to discover the set $F_{\geq\theta}$ of all θ-frequent itemsets in T.

The statistical nature of sampling implies that we obviously cannot guarantee to obtain the exact answer $F_{\geq\theta}$ based on just an analysis of a sample of the database. Therefore, the ϵ-close FIM problem is defined as follows: Given a tolerance threshold θ, obtain an ϵ-close solution using the minimum number of transaction samples. In [6], the following LT upper bound on the sample size required to obtain an ϵ-close solution with a confidence probability of $(1 - \frac{1}{h})$ was derived:

$$\frac{24}{\epsilon^2(1-\epsilon)\theta}\left[\Delta + 5 + \log\frac{5h}{(1-\epsilon)\theta)}\right] \tag{2.1}$$

We study here the empirical effectiveness of the LT bound for massive datasets. In particular, we consider 5 diverse real-life datasets from the FIMI repository [8], as well as synthetic datasets generated by the popular IBM QUEST dataset generator [2]. Each of these datasets contain up to 250 million transactions. We execute the Apriori algorithm on a sequence of samples, which starts with the LT bound and then, through a binary reduction technique, identifies the fewest number of samples at which the ϵ-close solution is maintained this is the number that would have been predicted by an Oracle. Each evaluation with a given sample size is repeated several times with different seeds for the random number generator to ensure the statistical soundness of our experiments. Across all the datasets, we consistently observe at least an order of magnitude gap between LT and the Oracle in fact, it is often closer to two orders of magnitude. The details of these experiments are presented in Chapter 6.

Chapter 3

Limitation Of Theoretical Bounds

In this chapter, we investigate whether significantly better theoretical bounds can be obtained if we are allowed to assume prior knowledge of some strong statistics of the database. To bound the sample size for ϵ-close solution, we have to get the sample bound at which all frequent itemsets are present in solution and no unacceptable false positive is present in solution. For this we divide the problem into two sub problems, one to get the sample bound which accepts all the frequent itemsets and second one to get the sample size which rejects all unacceptable false positives. Much of the work required in obtaining the LT bound lies in estimating the number of itemsets on either side of the acceptable support range in ϵ-close solutions. Therefore, it seems intuitive that, perhaps the strongest statistical information that can be advised to a frequent itemset miner is the number of maximal frequent itemsets [4], defined below, at the given support threshold.

Let $F_{\geq\theta}$ denote the set of all θ-frequent itemsets in T . An itemset $X \in F_{\geq\theta}$ is said to be a maximal θ-frequent itemset if $Y \in F_{\geq\theta}$such that $X \in Y$. Let $M_{\geq\theta}$ denote the set of all maximal θ-frequent itemsets in T . We now present a new upper bound on the required sample size based on the size of $M_{\geq\theta}$, which is arrived at by exploiting simple observations on the lattice structure of the itemsets.

We introduce two more notations that will be used in the rest of our exposition. Let S be a sample of T . We denote the set of all θ-frequent itemsets in the sample S by

$F^S_{\geq\theta}$. Similarly, we denote the set of all maximalθ-frequent itemsets in the sample S by $M^S_{\geq\theta}$.

3.1 Relaxed Random Algorithm

We begin with the description of the one-shot sampling algorithm for reporting frequent itemsets given in [6]. This algorithm, that we call as RelaxedRandom (to indicate that the support threshold is appropriately relaxed for mining the random database sample) is as follows:

- Pick a random sample S.
- Execute Apriori on S with support threshold set to a relaxed value of $(1-\frac{\epsilon}{2})\theta$ and report all the frequent itemsets output by the algorithm, i.e, $F^S_{(1-\frac{\epsilon}{2})\theta}$.

Let $X \subseteq \mathcal{I}$ be an itemset, with its "database support" denoted by f_X. The corresponding "sample support" of X in a simple random sample S is denoted by f^S_X. By expressing the counts of an itemset X in the sample and in the database, in terms of f^S_X and f_X, respectively, and using well known Chernoff bounds [3], the following was shown in [6]:

For all $0 < \delta < 1$,

$$\mathbf{Pr}[f^S_X \leq (1-\delta)f_X] \leq e^{-\frac{\delta^2|S|f_X}{2}} \tag{3.1}$$

Similarly, for all $0 < \delta < 1$,

$$\mathbf{Pr}[f^s_X \geq (1+\delta)f_X] \leq e^{-\frac{\delta^2|S|f_X}{3}} \tag{3.2}$$

3.2 Accepting all Itemsets in $F_{\geq\theta}$

We now present the sample size required to ensure that all itemsets in $F_{\geq\theta}$ are reported by Relaxed Random with a probability of $(1-\frac{1}{2h})$. Consider an itemset $X \in F_{\geq\theta}$. It

is not reported by Relaxed Random only if $f_X^S < (1 - \epsilon/2)\theta$. But, by Equation 3.1, it implies

$$\mathbf{Pr}[f_X^S \leq (1 - \epsilon/2) f_X] \leq e^{-\frac{\epsilon^2}{8}|S|\theta} \tag{3.3}$$

Now consider the set of maximal θ-frequent itemsets, i.e, $M_{\geq\theta}$, on which we can prove the following claim:

CLAIM 3.2.1. *Suppose it is true that, for all $X \in M_{\geq\theta}$, $f_X^S \geq (1 - \frac{\epsilon}{2})\theta$, then Relaxed Random reports every itemset in $F_{\geq\theta}$.*

The proof of the claim follows from the observation that every transaction containing X also contains every itemset $Y \subseteq X$. Therefore, it suffices to ensure that Relaxed Random returns all itemsets in $M_{\geq\theta}$. In turn, it means that it suffices if our algorithm ensures that each itemset in $M_{\geq\theta}$ is reported with a probability of at least $(1 - \frac{1}{2h|M_{\geq\theta}|})$. Therefore, we need

$$\mathbf{Pr}[f_X^S \leq (1 - \epsilon/2) f_X] \leq e^{-\frac{\epsilon^2}{8}|S|\theta} \leq \frac{1}{2h|M_{\geq\theta}|} \tag{3.4}$$

Solving for S in Equation 3.4 yields the following lemma:

LEMMA 3.1. *A sample of size $S \geq \frac{8}{\epsilon^2\theta}(\log|M_{\geq\theta}| + \log 2h)$ is sufficient to ensure that the probability of accepting all the elements of $F_{\geq\theta}$ is at least $(1 - \frac{1}{2h})$.*

3.3 Rejecting all Non-Frequent Itemsets

The key in ensuring the acceptance of all itemsets in $F_{\geq\theta}$ was identifying a set of itemsets ($M_{\geq\theta}$) such that, if the algorithm accepts all itemsets in the set, then it also accepts all of $F_{\geq\theta}$. Similarly, we need to identify a similar small enough set for the case of rejecting all non-frequent itemsets. Here, the notion of a "negative border" [14], defined below, comes in handy.

Consider an itemset $X \in M_{\geq\theta} \cup \text{NULL}$. Let $\mathcal{I}_r$ denote the set of individual items not present in X, i.e, $\mathcal{I}_r = \mathcal{I} \setminus \mathcal{X}$. Then, define $NB_X = \cup_{I \in \mathcal{I}_r}(X \cup I))$, i.e, the *minimal*

supersets of X. Finally, the negative border denoted by NB, is defined as $NB = \cup_{X \in M_{\geq\theta}} NB_X$.

CLAIM 3.3.1. *Suppose our algorithm does not report any itemset in NB, then, it is also true that it does not report any non-frequent itemset.* The proof of the claim follows from simply observing that any non-frequent itemset $X \notin NB$ is a superset of at least one itemset in NB. Careful counting shows that,

CLAIM 3.3.2. *The size of NB is at most $(m|M_{\geq\theta}| + |\mathcal{I}|)$.* Note that NB is conservative as it included some itemsets in the region $((1-\epsilon)\theta, \theta)$ which are AFPs. Now by following the steps of Section 3.2 and using the bound in Equation 3.2, we can show that:

LEMMA 3.2. *A sample of size $S \geq \frac{12}{\epsilon^2\theta}(\log m + \log|M_{\geq\theta}| + \log 2h)$ is sufficient to ensure that the probability of rejecting all the non-frequent itemsets is at least $(1 - \frac{1}{2h})$.*

By combining Lemmas 3.1 and 3.2, we obtain the following Maximal Frequent Itemset **(MFI)** bound:

THEOREM 3.3. *A sample of size*

$$S \geq \frac{12}{\epsilon^2\theta}(\log m + \log|M_{\geq\theta}| + \log 2h) \tag{3.5}$$

is sufficient to ensure that we obtain an ϵ-close solution with a probability of at least $(1 - \frac{1}{h})$.

3.4 Balancing the Two Sides

We can slightly improve the MFI bound by exploiting the imbalance in the denominators of the exponents in the Chernoff bounds in Equations 3.1 and 3.2 on the two sides of the deviation. For this, we have to change the Relaxed Random algorithm as follows: After obtaining the sample S, we run Apriori with the relaxed support threshold set to $(1 - \frac{\epsilon}{c})$ for some $c > 2$. So, the allowed deviation for the frequent itemsets reduces from a factor of $\frac{\epsilon}{2}$ to $\frac{\epsilon}{c}$, and the allowed deviation for the non-frequent itemsets in the region

$(1/N, (1-\epsilon)\theta)$ increases from $\frac{\epsilon}{2}$ to $\frac{(c-1)\epsilon}{c}$. As before, it results in two different bounds S_c^1 and S_c^2 for each value of c corresponding to what is shown in Lemmas 3.1 and 3.2, respectively. What we now need is a value of c for which $S_c^1 = S_c^2$. The calculations to derive the required value of c are straightforward. For the sake of brevity, we avoid the calculations and mention directly that a value of c around **2.22** is optimal and improves the required bound in Theorem 3.3 to:

$$S \geq \frac{10}{\epsilon^2\theta}\left(\log m + \log |M_{\geq\theta}| + \log 2h\right) \tag{3.6}$$

We carry out an empirical evaluation of the new MFI bound on the same suite of datasets (and input thresholds) which were used in studying the empirical effectiveness of the LT bound. Our results indicate that in every case the MFI bound is at least 5 to 10 times smaller than the LT bound. But, even then, MFI is still much larger than the Oracle sample size by about a factor of 5. The details of these experimental results are deferred to Chapter 6.

From a close observation of the sample sizes suggested by the LT and MFI theoretical bounds, it seems that their looseness arises from the basic tool used in their derivation, namely, the *Chernoff bounds* employed to estimate the frequency deviation of individual itemsets in the sample. In particular, their inverse dependence on ϵ^2 and θ even to estimate the bound for a single itemset turns out to be a major cause for the looseness. However, we are not aware of much better tools from the statistics literature that can be gainfully utilized in this case.

Comment: Even ignoring its looseness, the MFI bound is difficult to exploit in an algorithmic manner as it requires estimating the number of maximal frequent itemsets, and the complexity of computing this number has been shown to be #P-hard in [9].

Chapter 4

VISTA Algorithm

In the previous chapter, we showed the limitation of the LT and MFI theoretical bounds in accurately predicting the required sample sizes for practical use. We move on now to present a voting-based iterative sampling technique algorithm (VISTA) whose goal is to obtain ϵ-close solutions with a sample of the database whose size is comparable to that of Oracle. We treat the sum of the size of the samples used in each iteration as the total sample size of the VISTA algorithm, denoted by S_{VISTA}. Our focus here is on establishing the ability of the VISTA approach to efficiently obtain highly accurate results. The algorithm is based on an idea of repeated voting to collect enough credible evidence on whether or not an itemset is part of $F_{\geq\theta}$, based on evaluating small-sized samples at each step. These mini samples are henceforth referred to as "samplets" .

4.1 High level Description

Our algorithm is iterative. At any given point in time, we maintain a list L of candidate frequent itemsets and a list F of clearly evident frequent itemsets. These are based on samplets of size S_small whose value is a K-times scaled down version of the LT bound. In each iteration, we pick a simple random sample (with replacement) of size S_{small}. We consider the set of frequent itemsets in the sample and give each constituent a positive vote. The set F of clearly evident frequent itemsets is maintained as the set of itemsets

that have appeared as frequent in more than half of the iterations. We stop when the set F stabilizes, i.e. it does not change over a sufficiently long number of successive iterations (the number is determined dynamically based on the history of changes to F over the samplets processed thus far). The output of the algorithm is the set F at termination. The complete VISTA algorithm is presented in Algorithm 1.

4.2 Algorithm

input : Database T, θ, ϵ, and division factor $K \geq 1000$

output: ϵ-close solution (using minimum total sample)

Let $S_{small} = \frac{S}{K}$ where S denotes the LT bound for the input parameters ;

`// Initialization Phase`

Initialize L and F as empty sets ;

for $i \leftarrow 1$ **to** 20 **do**

  Obtain a sample S_i of size S_{small} ;

  Let L_i be the output of Apriori on S_i with $\theta_r = (1 - \frac{\epsilon}{2})$ and let $L = L_i \cup L'$;

end

Initialize F as the set of itemsets that appeared in at least $\frac{1}{2}$rd of the above iterations ;

`// Stabilization Phase`

while *Not Stabilized* **do**

  Obtain a sample of size S_{small} and let $L_{current}$ be the output of Apriori on 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 ;

  Update F to be itemsets that have appeared in $\frac{2}{3}$rd of the iterations (including Initialization Phase);

end

Output F.

Algorithm 1: The VISTA Algorithm

input: Required Stabilization Sequence Length l_0

Before the starting of stabilization phase, *CurLen =0*;

while $CurLen \leq l_0$ **do**

  Define F_{old} to be the F at the beginning of the current convergence sequence ;

  $F_{diff} = |\frac{F}{F_{old}}|$;

  if *(Cardinality of* F_{diff} *is not more than* 0.1% *of* F_{old}*)* **then**

    $CurLen{+}{+}$;

  else

    $CurLen$=0 ;

    New Convergence Sequence Begins;

  end

end

Procedure Stabilization Phase

4.3 Stabilization Phase

The set L consists of all itemsets that have appeared as frequent at least once. The set F is the set of itemsets that have appeared as frequent in a simple majority of the iterations. The algorithm runs its iterations till the set F stabilizes. We use the number of successive iterations in which the set F does not change as the degree of stabilization. But, the key question is how to choose the required length of such a sequence as a termination criteria (end of stabilization phase). We determine this criterion dynamically depending on progressive changes to F. The procedure is started by letting the required stable sequence length to be a nominal number, such as 10. The initial set F is chosen based on the output of the first few iterations (in our experiments, it is 20), and consists of all itemsets that appeared as frequent in a majority of these iterations. The stabilization phase starts after this initialization is completed. Our stopping condition is when we have a continuous sequence of l0 iterations in each of which the change of F relative to the start of the sequence is less than 0.1%. Note that the logic for the stabilization is inherent in Algorithm 1, but, to improve clarity, we abstract it out and present in

Procedure Stabilization Phase.

Conceptually, our approach has similarities with the "progressive sampling algorithm" of [11]. This algorithm is also iterative, analyzes a fresh samplet at each iteration, and terminates based on a convergence criteria. The most important difference is in the notion of voting that we employ which provides a *global* criteria of convergence. In contrast, the progressive sampling algorithm employs a *local* convergence criteria that is based on a comparison of the "characteristic frequent itemsets" of the previous two iterations. Moreover, the size of the samplets in successive iterations are increased in a geometric manner as suggested in [13], whereas the samplet size is fixed in our approach. The local convergence criteria seems to require a long time to stabilize in the experiments of [11]. Further, it is reported that at the prescribed level of convergence (0.95 proportion of similarity between two successive samplets), for a synthetic database of 250 million transactions (similar to dataset SD1 in Chapter 5), a sample of size 10% was required for stabilization (whereas we terminate at 0.5% of the database). Finally, their technique does not give any assurance of the nature of errors that may be present in the results obtained from the sample, whereas our expectation with VISTA is to have no errors except for AFPs.

4.4 ϵ-Close Frequencies

In previous sections, we explained how VISTA algorithm report the identities of the ϵ-close frequent itemsets. As mentioned in Chapter 1, ϵ-close solution includes the frequency of the itemsets also. We find the frequency of a known ϵ-close frequent itemset as follows: Collect the samplet taken in each iteration of VISTA algorithm and make a larger sample by adding all samplets. Find the frequency of the ϵ-close frequent itemset by making a one iteration over large sample. Report the frequency of the itemset in this sample as frequency of the itemset from VISTA algorithm. We observed that, in all our experiments the VISTA reported frequency is with in the ϵ-range(refer to definition in Chapter 1).

Chapter 5

Experimental Setup

In this chapter, we present the experimental setup for empirical evaluations of the LT bound [6] and the new MFI bound and VISTA algorithm. First we present the details of the datasets used in our experiments in next section. As mentioned in the introduction, the power of the sampling techniques come to fore when working with large datasets. We give details of how to generate large instance of the FIMI datasets such that the correct output does not change. Then we describe the method we followed in deciding θ for different datasets.

5.1 Datasets for the Experiments

The datasets used in our experiments include real-life datasets from the FIMI repository that are standard in evaluation of frequent itemset mining [8], as well as synthetic datasets generated using the IBM QUEST data generator [2]. It is important to note that the LT bound is not useful in the context of small databases as the component $\frac{24}{\epsilon^2\theta}$ itself works out to be much larger than the database size. So, it is useful only in the case of large datasets, and we therefore conduct the bulk of our experiments on datasets that run to 250 million transactions. The details of how we generate large instances of the FIMI datasets such that the distributional statistics of the dataset remain unchanged is given in Section 5.1.3.

5.1.1 Synthetic Datasets

We experimented with several synthetic datasets, and present experimental results on a representative instance that has 250 million transactions. It was generated using the QUEST tool with the parameters set as follows: Maximum transaction length = 35, average transaction length = 10, and total number of items = 1000. The maximum transaction length was limited since the LT bound increases linearly with its value. We call this as the SD1 dataset.

5.1.2 Real Datasets

The FIMI repository consists of several real-life datasets. We pick five that were chosen to have different characteristics. They are: (i) RETAIL, a sparse dataset, (ii) CONNECT, a dense dataset, (iii) MUSHROOM which has maximum transaction length equal to average transaction length, (iv) ACCIDENTS and (v) PUMSB-STAR. These datasets are relatively small and we therefore generate large variants that are statistically identical to the base datasets. The details of the different datasets are captured in Table 5.1. Columns OS and GS refer to their original sizes in the repository and their size in our experiments, respectively.

Dataset	OS	GS (10^6)	Items	T_{avg}	T_{max}
SD2(s,l)	100000	250	1000	10	30
ACCIDENTS	340183	150	468	31	51
CONNECT	67557	195	129	43	43
PUMSB-STAR	49046	196	2088	50	63
MUSHROOM	8124	250	119	23	23
RETAIL	88126	176	16469	10	76

Table 5.1: Details of the FIMI datasets

5.1.3 Generation of Large Datasets

The original sizes of the FIMI datasets are relatively small. One could generate a large instance of a dataset by simply appending multiple copies of it, one after another. But, on such a dataset, a trivial sampling algorithm which picks the first OS transactions gives the exact result! Therefore, it is essential to avoid trivial solutions which exploit the manner in which the large instances are generated. We generate large random instances of the original datasets, but without changing the underlying distributional statistics. The original dataset is partitioned into $P_1, \ldots, P_C$ partitions, each having 10 transactions apiece. The original dataset is repeated GS OS times (refer to Table 5.1) as follows: Every time we pick a random permutation π of (1, . . . ,C), and then write the partitions according to this permutation. Specifically, the partitions are written in the order $(P_{\pi(1)}, P_{\pi(2)}, \ldots, P_{\pi(C)})$. Within each partition, the transactions are again written by picking a random permutation. Note that this process does not change the column distributional statistics in anyway.

5.2 Setting Support Threshold θ

For frequent itemset mining, the most important input parameter is θ, the support level. Since there are many possible values of θ, one needs to choose a value of θ that is challenging from the point of view of producing θ-close solutions. In this section, we describe the method we followed in deciding θ for different datasets.

5.2.1 Synthetic Datasets

For the large synthetic dataset SD1, the number of frequent itemsets at lower support levels is very high, possibly due to random correlations in the data generation. Therefore, we set the θ values to be relatively high at 0.01 and 0.02 at these settings, there are a reasonable number of frequent itemsets (4837 and 1268, respectively), but not impractically many.

Consider a particular θ value, say $\theta = x$. Let the number of frequent itemsets at x

be N_x. Let $N_{x-\delta}$ be the number of frequent itemsets when x is decremented by a small value δ. If the $\frac{N_{x-\delta}}{N_x}$ is close to 1, it is not a difficult point to operate under for a sampling scheme as minor variations in its frequency estimates are unlikely to result in significant errors. On the other hand, if $\frac{N_{x-\delta}}{N_x}$ is high, it is a difficult to point for a sampling scheme to operate under as minor variations in statistical estimates from the sample can lead to significant errors. Starting with 1 and proceeding to lower values, we stop at the first x where the ratio $\frac{N_{x-\delta}}{N_x}$ is above a certain threshold. The rationale for stopping at the first possible instant is, the sample size given at this point is least across all such points (and hence difficult to ensure accuracy).

5.2.2 Real Datasets

For the FIMI datasets, we deliberately choose θ values that make it hard for the sampling techniques to work well that is, we aim to evaluate the tough nut scenarios. Consider a particular θ value, say $\theta = x$. Let the number of frequent itemsets at x be N_x. Let $N_{x-\delta}$ be the number of frequent itemsets when x is decremented by a small value δ. If the $\frac{N_{x-\delta}}{N_x}$ is close to 1, it is not a difficult point to operate under for a sampling scheme as minor variations in its frequency estimates are unlikely to result in significant errors. On the other hand, if both $\frac{N_{x-\delta}}{N_x}$ and $N_{x-\delta} - N_x$ are high, it is a difficult situation for a sampling scheme to operate under as minor variations in statistical estimates from the sample can lead to significant errors. Starting with $\theta = 1$ and proceeding to lower values, we stop at the first x where the ratio $\frac{N_{x-\delta}}{N_x}$ and the difference $N_{x-\delta} - N_x$ are above certain thresholds. The rationale for stopping at the first possible instant is, the sample size given at this setting is the least across all such settings, thereby increasing the scope for making errors.

Chapter 6

Empirical Results

In previous chapters, we presented the LT bound, MFI bound and VISTA algorithm to get the minimum sample bound for ϵ-close solution. To evaluate the relative performance of these bounds and to confirm the claims that we have made about their excepted behavior, we conducted a series of experiments that covered real and synthetic datasets (*detailed in Chapter 5*). The performance metric in these experiments includes number of UFPs, AFPs, TPs present in solution when frequent itemset mining algorithm is executed over certain sample taken from database.

6.1 Evaluation of the LT Bound

In this section, we present an empirical evaluation of the gap between the LT bound and practically found minimum required sample. By practically found minimum required sample we mean, a minimum sample determined by an all powerful oracle which can evaluate all possible sample sizes and output the minimum size at which ϵ-close solutions are still obtained. We refer to this minimum size as S_{Oracle}.

We implement the Oracle for given input thresholds θ and ϵ as follows: Let S denote the sample size given by the bound
$\frac{24}{\epsilon^2(1-\epsilon)\theta}\left[\Delta + 5 + \log\frac{5h}{(1-\epsilon)\theta}\right]$ We execute RelaxedRandom algorithm on the following

sequence of samples: $S_1, S_2, ...$ where $S_i = S2^{i-1}$. We stop the process at the first S_i for which the algorithm returns (i) one or more UFP itemsets OR (ii) misses some itemset in $F_{\geq\theta}$. Note that the minimum required sample lies in the range $[S_i, S_{i-1}]$. We now carry out a similar binary search within this region to further narrow down to the minimum required sample. We denote the size determined by this process as S_{Oracle}, the minimum required sample size to obtain ϵ-close solution based on only one-shot sampling. However, a sampling strategy that uses sub-sampling may be able to obtain ϵ-close solution with a sample smaller than S_{Oracle}.

In the results given below, I_{FN} refers to the false negatives (missing itemsets from $F_{\geq\theta}$), I_{AFP} refers to the of Acceptable False Positives, and I_{UFP} refers to the number of Unacceptable False Positives reported by the algorithm. Each experiment (combination of dataset and input thresholds) was run multiple times (often 10) to make sure the results are credible. However, for brevity, we present the results of just one run, in particular, the one in which S_{i-1} is the highest. Therefore, the actual gap between Oracle and the LT bound may be even larger than what our results suggest below. We present the results of the experiments conducted on synthetic and FIMI real datasets in next subsections.

6.1.1 Synthetic Dataset Results

The synthetic dataset SD1 with 250 million transactions is tested with θ values of 0.01 and 0.02. The other parameters are set as $\epsilon = 0.1$, and $h = 100$ (i.e, probability of success is 0.99). The LT bound works out to be 12 million and 6 million for $\theta = 0.01$, 0.02, respectively. The results of the experiment are shown in Table 6.1 and 6.2, which presents the number of FNs, AFPs, and UFPs at different sample sizes. The actual number of θ-frequent itemsets at 0.01 and 0.02 are 4837and 1268, respectively. The row corresponding to the Oracle is highlighted in bold. Across the ten runs, the first four rows (upto highlighted row) for the two columns FN and UFP are identical. In this case, the theoretical bound is 30 times larger than the Oracle. Note that the number of itemsets in the range $[(1-\epsilon)\theta, \theta)$, i.e, AFPs, are comparable for LT and Oracle.

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
12 Million	5	0	470	0
6 Million	2.5	0	473	0
3 Million	1.3	0	448	0
0.4 Million	**0.16**	**0**	**503**	**0**
0.3 Million	0.1	0	429	2
0.1 Million	0.04	5	507	11

Table 6.1: Evaluation of the LT Bound on SD1 with $\theta = 0.01$

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
6 Million	2.5	0	145	0
3 Million	1.3	0	165	0
0.5 Million	0.2	0	151	0
0.1 Million	**0.04**	**0**	**137**	**0**
0.05 Million	0.02	2	152	1
0.025 Million	0.01	21	148	43

Table 6.2: Evaluation of the LT bound on SD1 with $\theta = 0.02$

6.1.2 Evaluation on FIMI Real-life Repository

We evaluated LT bound on FIMI datasets to test the looseness of the bound over real datasets [8]. For each of the five datasets, we set the value of θ to a challenging value, as described in Chapter 5 The ϵ parameter is set to be 0.01 for all datasets except RETAIL, which is sparse and therefore needs θ to be as low as 0.0045. Further, requiring an accuracy range of $(1 - 0.01)0.0045$ turns to be stringent and the LT bound increases beyond the size of the dataset itself. Therefore, in this case, we relax ϵ to 0.1 and the LT bound works out to be 101 million. The results of one run of the experiment for each of the datasets are captured in Table 6.3 through 6.7. As mentioned before, there are no deviations across experiments, especially in evaluating the Oracle.

Summary : Table 6.7 summarizes the evaluation of the LT bound for the synthetic and different FIMI datasets. For each dataset it captures the support level (θ), number of frequent itemsets at θ, sample size suggested by LT and the Oracle across multiple runs of the experiments. Note that the best instances for LT are those in which the average transaction length is comparable to the maximum transaction length. However, across all the instances we observe huge gaps (from 24 to 200 times) between LT and Oracle. In fact, for very sparse datasets, the ϵ value needs to be increased in order to even get a reasonable bound. The looseness is at least an order of magnitude, if not two.

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
44 Million	17.6	0	79	0
32 Million	12.8	0	79	0
16 Million	6.4	0	79	0
4 Million	1.6	0	79	0
0.6 Million	**0.24**	**0**	**82**	**0**
0.25 Million	0.1	7	79	0

Table 6.3: Evaluation of the LT bound on MUSHROOM at $\theta = 0.4$

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
32 Million	21.33	0	2644	0
16 Million	10.67	0	2614	0
8 Million	5.3	0	2753	0
2 Million	1.35	0	2932	0
0.8 Million	**0.54**	**0**	**3001**	**0**
0.5 Million	0.34	1	3011	0
0.25 Million	0.17	5	2985	0

Table 6.4: Evaluation of the LT bound on ACCIDENTS at $\theta = 0.5$

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
16 Million	9	0	65	0
8 Million	4.5	0	65	0
4 Million	2.26	0	63	0
2 Million	1.13	0	63	0
0.6 Million	**0.37**	**0**	**62**	**0**
0.5 Million	0.31	4	56	0
0.25 Million	0.15	7	54	0

Table 6.5: Evaluation of the LT bound on RETAIL at $\theta = 0.0045$

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
25.6 Million	12.24	0	1913	0
12.8 Million	6.12	0	1913	0
6.4 Million	3.06	0	1723	0
3.2 Million	1.53	0	1702	0
1.6 Million	0.76	0	1536	0
0.8 Million	0.38	0	1458	0
0.3 Million	**0.014**	**0**	**1533**	**0**
0.1 Million	0.0475	3	1952	0
0.05 Million	0.0238	232	2334	0

Table 6.6: Evaluation of the LT bound on PUMSB-STAR at $\theta = 0.40$

6.2 Evaluation of the MFI bound

We now move on to the empirical evaluation of the MFI bound presented in Chapter 3. We evaluated the bound as follows: First, Apriori was executed on the entire dataset and the correct result was obtained. From this, the number of maximal frequent itemsets was computed and fed as the input parameter $|M_{\geq\theta}|$ to the bound. Table 6.9 summarizes the results, wherein the fifth column shows the multiplicative gap between LT and MFI bounds and the sixth column shows the gap between the MFI bound and Oracle. It

Sample size	% of T	I_{FN}	I_{AFP}	I_{UFP}
16 Million	8.16	0	6437	0
8 Million	4.1	0	6421	0
2 Million	1.02	0	6933	0
1 Million	0.5	0	7377	0
0.3 Million	**0.015**	**0**	**9013**	**0**
0.25 Million	0.012	5	11245	0

Table 6.7: Evaluation of the LT bound on CONNECT at $\theta = 0.9$

Benchmark	θ	$\lvert F_{\geq\theta}\rvert$	S_{LT}	S_{Oracle}	$\frac{S_{LT}}{S_{Oracle}}$
SD1	0.01	4837	12	0.4	30
MUSHROOM	0.4	12384	44	1	73
ACCIDENTS	0.5	8099	30.7	1	38
RETAIL	0.0045	580	100.64	0.5	168
PUMSB-STAR	0.4	27348	45.8	0.5	150
CONNECT	0.9	35411	14.7	0.5	49

Table 6.8: Summary of the LT bound Evaluation

is easy to observe in this table that the knowledge of the strong statistical measure significantly improves the tightness of the bound. But, there is still a large gap (as much as 8 times in some instances) between the MFI bound and the Oracle.

6.3 Evaluation of the VISTA Algorithm

In this section, we present results for VISTA algorithm (Chapter 4) to get minimum required sample bound. One of the input parameter for VISTA algorithm is the samplet size. To choose the samplet size, we experimented with different scaling factors for the LT bound, i.e, the input parameter K varied from 1000 to 10000. In each of the cases, although the number of iterations for the stabilization were different, the total sample size across iterations did not vary much. Moreover, the results were also nearly identical.

Dataset	S_{LT}	S_{MFI}	S_{Oracle}	$\frac{S_{LT}}{S_{MFI}}$	$\frac{S_{MFI}}{S_{Oracle}}$
SD1	13.7	1.83	0.4	7.48	4.6
ACCIDENTS	30.7	3.4	0.8	9.03	4.25
CONNECT	14.7	1.7	0.3	8.64	5.7
PUMSB-STAR	45.8	4.4	0.3	10.41	14.7
MUSHROOM	44	7.7	0.6	5.71	12.83
RETAIL	49	4.3	0.6	11.4	7

Table 6.9: Comparison of the MFI bound with LT and Oracle

Therefore, we present the results with $K = 1000$.

With regard to stabilization, the number of iterations required to confirm the algorithm has reached the stabilization is decided by a parameter called required sequence length. This parameter will get incremented depending on the $\frac{|F_{diff}|}{10}$. We denote the total sample used by VISTA by S_{VISTA}. One impressive observation that held across all datasets (and across multiple runs for each dataset) was that at the end of the stabilization phase, the algorithm returned truly ϵ-close solutions, i.e, both FN and UFP were zero. Table 6.9 captures the sample sizes for: LT, MFI bound, VISTA and Oracle. We conclude that VISTA stabilized with total samples much smaller than MFI bound. In fact, as claimed in the introduction, S_{VISTA} is always within twice of S_{Oracle}, the minimum required sample for one-shot sampling.

In Figure 6.1 we also show how the stabilization phase of the algorithm proceeds for the synthetic dataset. At each iteration, we plot the number of true positives, AFPs, UFPs, and FNs. At stabilization both FNs and UFPs are zero. The vertical line around 80th iteration corresponds to the iteration at which the total sample used is equal to S_{Oracle}. Note that the algorithm stabilizes around 110th iteration.

6.3.1 Hard Case for Iterative Sampling

Recall that the θ values for the different datasets were chosen so as to make it difficult for sampling algorithms to obtain accurate results. In evaluating VISTA, we go one

Dataset	S_{LT}	S_{MFI}	S_{VISTA}	S_{Oracle}
SD1	13.7	1.83	0.73	0.5
ACCIDENTS	30.7	3.4	1.4	0.8
CONNECT	14.7	1.7	0.6	0.3
PUMSB-STAR	45.8	4.4	0.49	0.3
MUSHROOM	44	7.7	1.2	0.6
RETAIL	49	4.3	0.7	0.6

Table 6.10: VISTA Algorithm Results

Dataset	RelaxedRandom	VISTA
Size (Mill)	1.5	1.5
Support	21.4%	21.4%
Total Output (TP+AFP)	22852	19375
True Positives (TP)	18853	18853
AFP'S (AFP)	3999	522

Table 6.11: Hard Case: Evaluation on MUSHROOM at $\theta = 0.214$

step further and identify an even harder case. We try to identify a support level $\theta = x$ such that the ratio $\frac{N_{x-\delta}}{N_x} \gg 1$ (the notations N_x and $N_{x-\delta}$ are used as in Chapter 5), i.e, there is a step-like increase in frequent itemsets at x. It turned out that not all the datasets had such a value of x but the MUSHROOM dataset did feature such a point at $\theta = 0.214$. Figure 6.2 shows the log plot of the number of frequent itemsets versus the θ support threshold. Observe the step-like increase around 21.4%. We evaluated VISTAs accuracy in this scenario as compared to the accuracy of the RelaxedRandom algorithm (Chapter 3) on a random sample of equivalent size S_{VISTA}. The results are captured in Table 6.11 and show that while RelaxedRandom returns far too many itemsets in the range $[(1-\epsilon)\theta, \theta]$, VISTA returns very few of them.

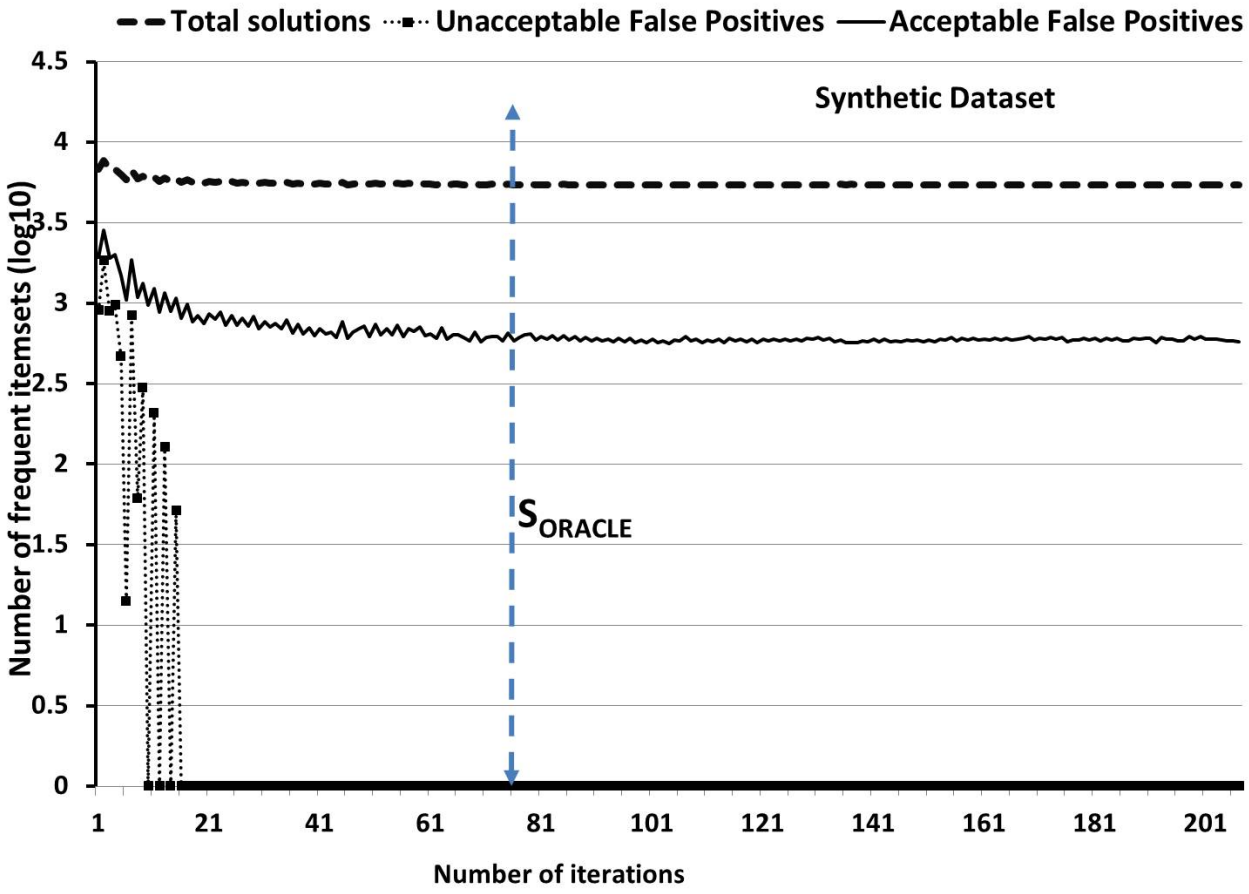

Figure 6.1: Progression towards stabilization for SYNTHETIC

6.3.2 Frequencies reported by VISTA

From previous sections, we know that VISTA algorithm is able to report all the identities of the frequent itemsets. To find the corresponding frequencies of the itemsets we executed the method given in Chapter 4.4. We observed that the frequencies reported by VISTA are with in ϵ-range. We call the frequencies reported by VISTA as ϵ-close frequencies. Table 6.12 summarizes the evaluation of ϵ-close frequencies reported by VISTA. We use =0.01 for all datasets. For each dataset, it captures the support(θ), number of frequent itemsets in database (F), number of ϵ-close frequent item sets reported by VISTA(F_ϵ), number of itemsets whose reported frequency is in ϵ-close frequency range,

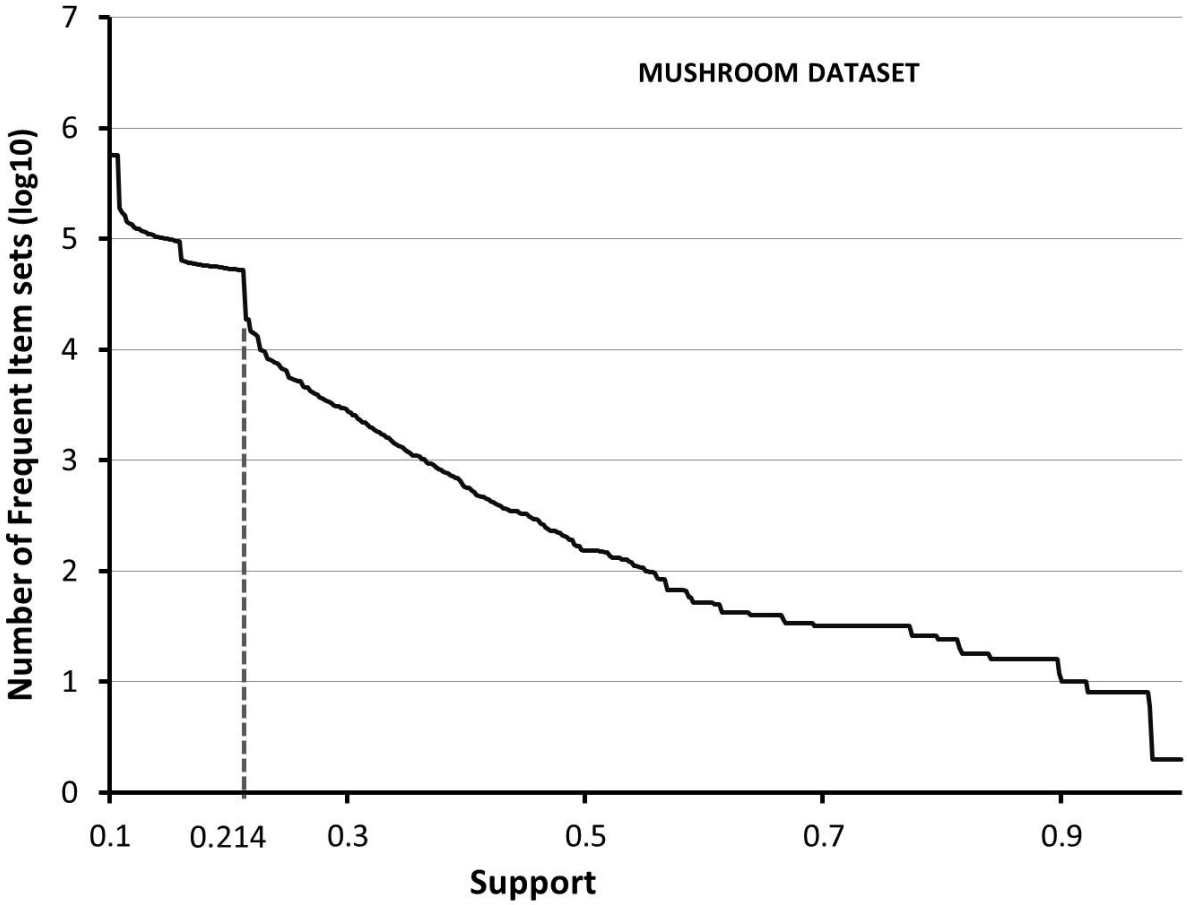

Figure 6.2: Choosing θ in the hard case

maximum deviation of ϵ-close frequency from database frequency, average deviation of ϵ-close frequency from database frequency for all itemsets. The last two columns shows the deviation in terms of ratio of al lowed deviation (ϵ) and observed deviation ($\epsilon_{avg}, \epsilon_{max}$). Across all datasets we observe that VISTA is able to report ϵ-close frequencies for all itemsets. And deviation of ϵ-close frequency from database frequency is lower than allowed deviation.

Dataset	θ	F	F_ϵ	F_ϵ^{freq}	$\frac{\epsilon_{avg}}{\epsilon}$	$\frac{\epsilon_{max}}{\epsilon}$
MUSHROOM	0.2	53846	54653	54653	0.26	0.63
ACCIDENTS	0.5	8099	11052	11052	0.34	0.73
CONNECT	0.9	35411	42404	42404	0.43	0.81
PUMSB-STAR	0.4	27348	30004	30004	0.21	0.58
RETAIL	0.0045	580	611	611	0.28	0.79

Table 6.12: Summary of Frequencies reported by VISTA

Chapter 7

Conclusions

In this thesis, we have first evaluated the looseness of various theoretical bounds on the sample size required to obtain ϵ-close solutions. Our results over a large and representative suite of massive datasets demonstrate that both the LT bound from the literature, as well as our new and tighter MFI bound which assumes apriori knowledge of the number of maximal frequent itemsets, are both rather loose as compared to the minimum size required, often running to orders of magnitude. The reasons for the looseness are inherent since they are based on the Chernoff bounds which turn out to be rather weak. As a practical solution to this issue, we have presented VISTA, a voting-based iterative algorithm that consistently provides the data mining quality of ϵ-close solutions while only incurring overheads comparable to that required by Oracle, an idealized one-shot sampling algorithm.

In a nutshell, we have shown here how sampling can be effectively used for mining massive datasets, whereas most of the prior literature had confined their attention and were applicable only to small datasets.

Appendix A

Stabilization of VISTA Algorithm for FIMI Datasets

In Chapter 6, we showed how the stabilization phase of the VISTA algorithm proceeds for synthetic dataset. Here, we show the graphs for the real datasets. Figure A.1 through A.4 shows the stabilization phase of VISTA for FIMI real datasets.

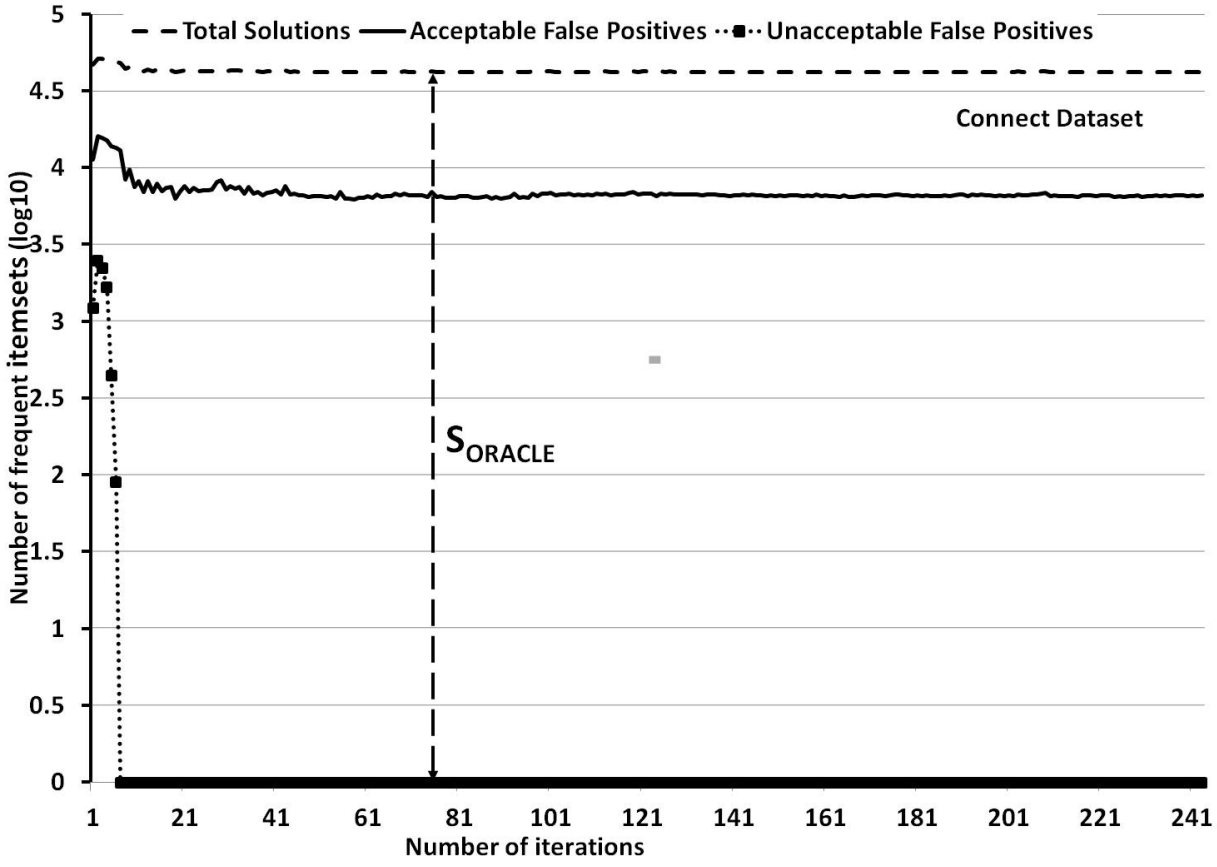

Figure A.1: Progression towards stabilization for CONNECT

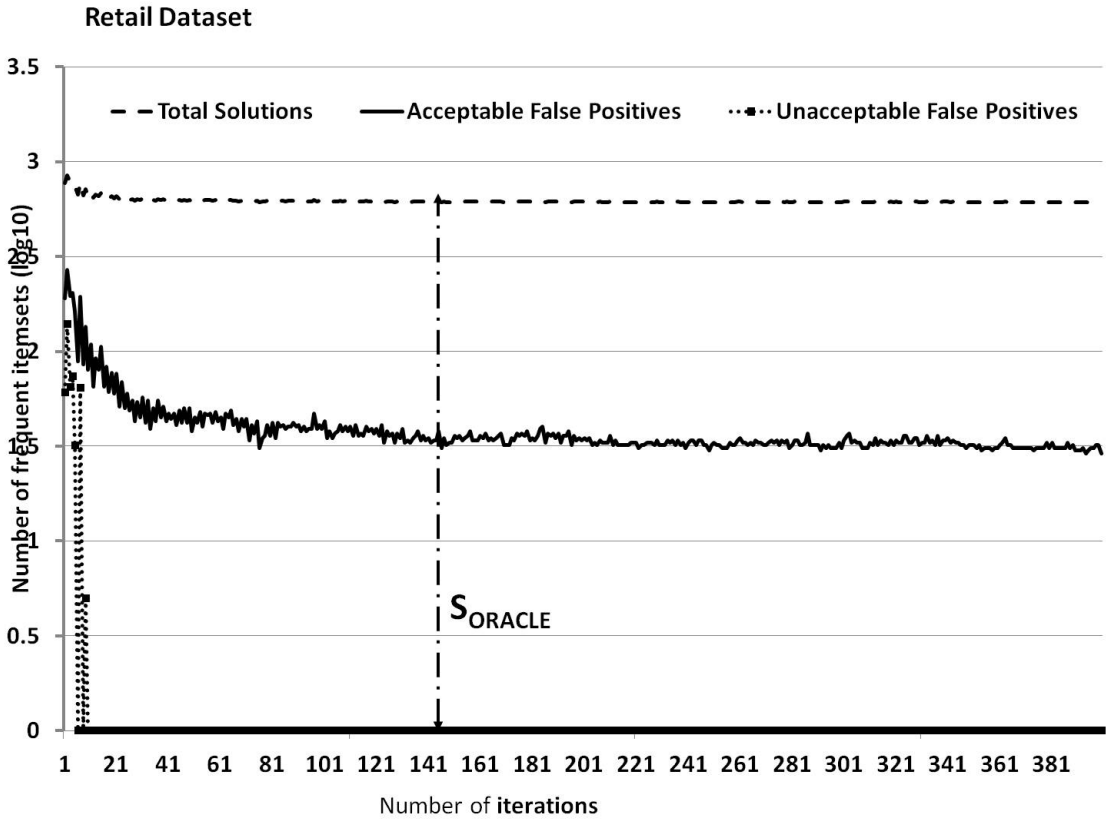

Figure A.2: Progression towards stabilization for RETAIL

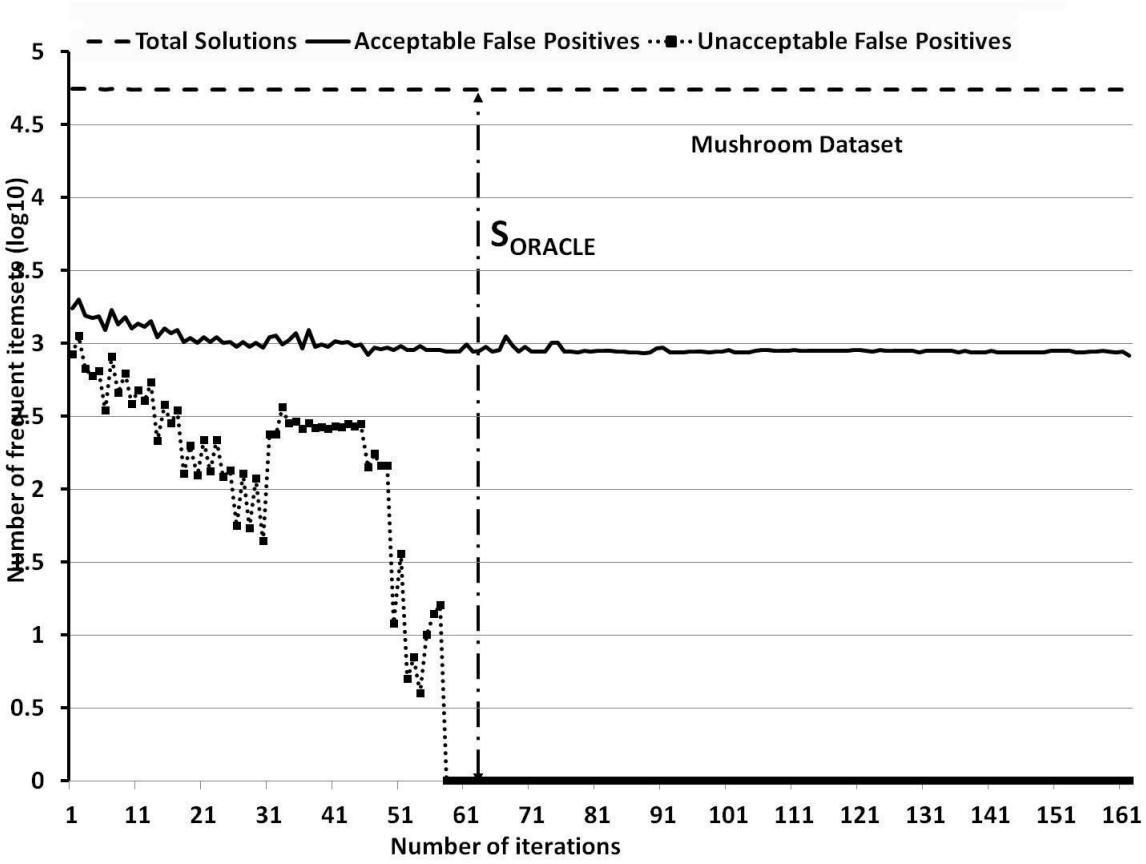

Figure A.3: Progression towards stabilization for MUSHROOM

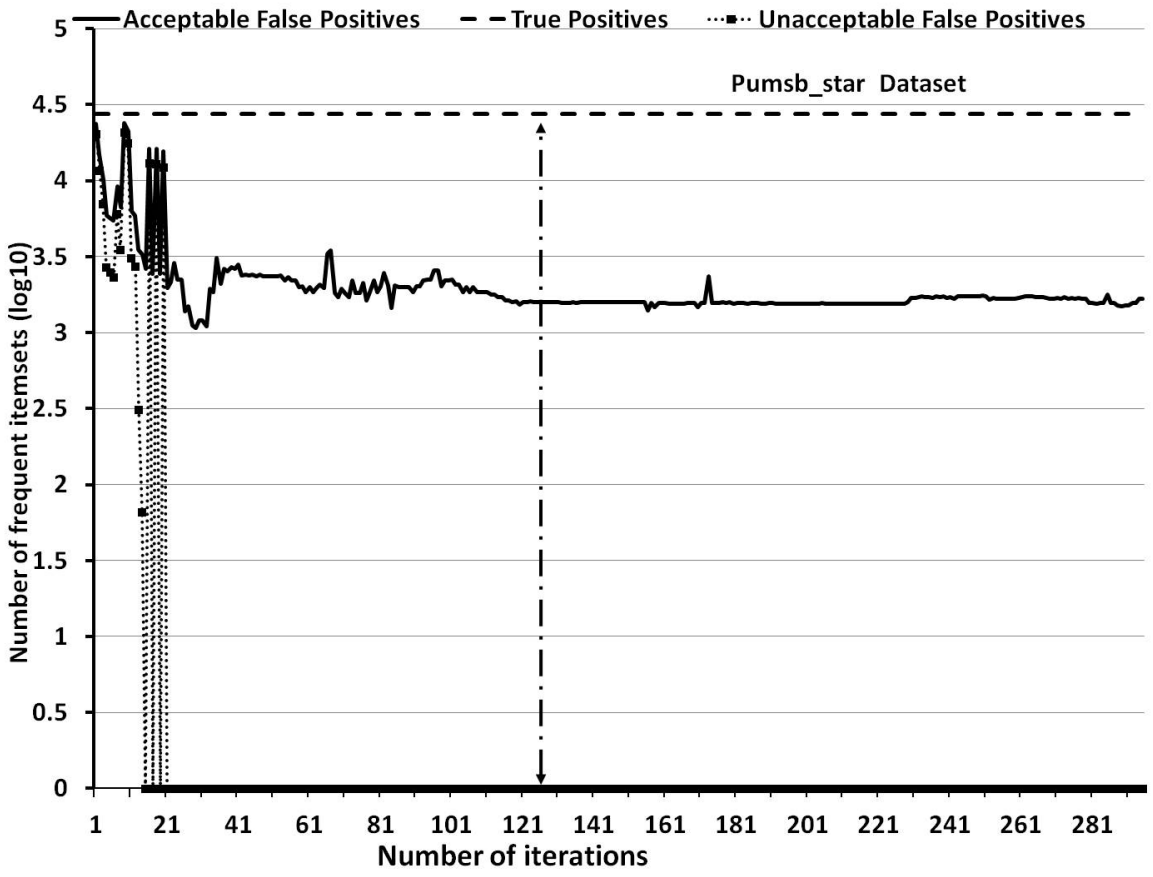

Figure A.4: Progression towards stabilization for PUMSB_STAR

References

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993, pp. 207-216.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the International Conference on Very Large DataBases (VLDB)*, 1994, pp. 487-499, The IBM QUEST synthetic data generator described in the paper is available at the following URL: http://www.cs.loyola.edu/cgiannel/assoc_gen.html.

[3] N. Alon and J. Spencer, *The Probabilistic Method.* John Wiley, 1992.

[4] R. Bayardo Jr, "Efficiently mining long patterns from databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998, pp. 85-93.

[5] A. Ceglar and J. Roddick, "Association mining," *ACM Computing Surveys*, vol. 38, no. 2, 2006.

[6] V. Chakaravarthy, V. Pandit, and Y. Sabharwal, "Analysis of sampling techniques for association rule mining," in *Proceedings of the International Conference on Database Theory (ICDT)*, 2009, pp. 276-283.

[7] B. Chen, P. Haas, and P. Scheuermann, "A new two-phase sampling based algorithm for discovering association rules," in *Proceedings of ACM-SIGKDD Conference on Knowledge Discovery and Data mining (KDD)*, 2002, pp. 462-468.

[8] B. Goethals and M. Zaki, "Advances in frequent itemset mining implementations: Report on FIMI 03," *ACM SIGKDD Newsletter*, 2004, The FIMI Repository is available online at URL: http://fimi.cs.helsinki.fi/data.

[9] D. Gunopulos, R. Khardon, H. Mannila, S. Saluja, H. Toivonen, and R. Sharma, "Discovering all most specific sentences," *ACM Transactions on Database Systems* , vol. 28, no. 2, pp. 140-174, 2003.

[10] H. Mannila, H. Toivonen, and A. Verkamo, "Efficient algorithms for discovering association rules," in *Proceedings of the AAAI workshop on Knowledge Discovery in Databases* , 1994, pp. 181-192.

[11] S. Parthasarathy, "Efficient progressive sampling for association rules," in *Proceedings of the International Conference on Data Mining (ICDM)*, 2002, pp. 354-361.

[12] A. Pietracaprina, M. Riondato, E. Upfal, and F. Vandin, "Mining top-k frequent itemsets through progressive sampling," *Data Mining and Knowledge Discovery* , vol. 21, no. 2, pp. 310-326, 2010.

[13] F. Provost, D. Jensen, and T. Oates, "Efficient progressive sampling," in *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 1999, pp. 23-32.

[14] H. Toivonen, Sampling large databases for association rules, in *Proceedings of the International Conference on Very Large DataBases (VLDB)*, 1996, pp. 134-145.

[15] "Walmart Report," Available online at http://walmartstores. com/FactsNews/FactSheets/Merchandising Fact Sheet.pdf.

[16] M. Zaki, S. Parthasarathy, W. Li, and M. Ogihara, " Evaluation of sampling for data mining of association rules," in *Proceedings of the International Workshop on Research Issues in Data Engineering (RIDE)*, 1997.