Discovering frequent patterns in sensitive data

Raghav Bhaskar
Microsoft Research India

Srivatsan Laxman
Microsoft Research India

Adam Smith
Pennsylvania State University

Abhradeep Thakurta
Pennsylvania State University

ABSTRACT

Discovering frequent patterns from data is a popular exploratory technique in data mining. However, if the data are sensitive (e.g. patient health records, user behavior records) releasing information about significant patterns or trends carries significant risk to privacy. This paper shows how one can accurately discover and release the most significant patterns along with their frequencies in a data set containing sensitive information, while providing rigorous guarantees of privacy for the individuals whose information is stored there.

We present two efficient algorithms for discovering the K most frequent patterns in a data set of sensitive records. Our algorithms satisfy *differential privacy*, a recently introduced definition that provides meaningful privacy guarantees in the presence of arbitrary external information. Differentially private algorithms require a degree of uncertainty in their output to preserve privacy. Our algorithms handle this by returning 'noisy' lists of patterns that are close to the actual list of K most frequent patterns in the data. We define a new notion of *utility* that quantifies the output accuracy of private top-K pattern mining algorithms. In typical data sets, our utility criterion implies low false positive and false negative rates in the reported lists. We prove that our methods meet the new utility criterion; we also demonstrate the performance of our algorithms through extensive experiments on the transaction data sets from the FIMI repository. While the paper focuses on frequent pattern mining, the techniques developed here are relevant whenever the data mining output is a list of elements ordered according to an appropriately 'robust' measure of interest.

1. INTRODUCTION

Frequent Itemsets Mining (FIM) is a fundamental problem in data mining [2, 16, 15]. In this problem, there is a universe M of *items* (or symbols) and each data record, called a *transaction*, is an unordered collection of items from M. For example, a transaction could represent the items purchased by a customer in one visit to a grocery store. An *itemset*[1] is a (typically small) subset of items out of M. A transaction *supports* a pattern if it contains the pattern. The *frequency* of a pattern is the proportion of transactions in the data that support it. The goal in FIM is to discover and report the patterns that occur most frequently in the data. There are typically two ways to control the size of output: (i) user specifies an explicit frequency threshold and the algorithm outputs all patterns whose frequencies in data exceed that threshold, or (ii) user specifies a positive number K and the algorithm outputs the K most frequent (*top* K) patterns. The *Apriori* algorithm [2, 15] for FIM is regarded as one of the most successful of techniques in data mining [25]. It forms the basis of several data mining tasks such as mining association rules [2], detecting correlations, discovering emerging patterns [7], etc. Beginning with [2], there is an extensive body of work (e.g. see [16, 13, 26, 23]) that deals with FIM and its variants in transactional data sets. In this paper we are concerned with mining top K itemsets from transaction data.

Many compelling applications of frequent pattern mining deal with sensitive data. For example, discovering strong correlations, trends and rules from electronic medical records of hospital patients can be a valuable source of information for society [12, 17]; understanding common user behavior can provide useful information for pricing advertising. However, releasing information about sensitive data carries serious risks to privacy. Simply removing obvious identifiers, such as names and exact addresses, does not protect privacy since the remaining information may identify a person uniquely [24, 4]. Even relatively sophisticated anonymization techniques (*e.g.*, those based on *k-anonymity* [24]) can fail to hide the exact values of sensitive attributes when combined with background knowledge [19] or easily available external information [11]. Recent theoretical and experimental results demonstrate that reasoning about the privacy of high-dimensional data is particularly difficult. For example, Dinur and Nissim [6] showed that even highly noisy answers to a large number of counting queries ("how many records in the database satisfy the following predicate?") allow an adversary to reconstruct large parts of a data set exactly. External information is difficult to reason about in high-dimensional settings. For example, Narayanan and Shmatikov [21] showed how even a few pieces of a long record are enough to link it to outside sources; Ganta *et al.* [11] show that independently anonymized releases of large data sets could be combined to reveal sensitive information. There is a basic tension, then, between *utility* and *privacy*. The fundamental problem is to understand where exactly the trade off lies between these two.

1.1 Contributions

We present two efficient algorithms for discovering frequent itemsets in sensitive data sets. Our algorithms satisfy *differential privacy* [9, 8], a recently introduced definition which provides meaningful privacy guarantees in the presence of *arbitrary* external in-

[1] We use the terms pattern and itemset interchangeably.

KDD '2010, Washington, DC.

formation. Differential privacy imposes a condition on the algorithm that releases some statistics about a data set x. Roughly, it states that small changes to x should not be noticeable to the users (or adversaries) who view the released statistics. This implies that *no matter what the adversary knows ahead of time, he learns the same thing about Alice whether or not her data is actually included in the data set* x [8, 11]. Our algorithms thus provide a picture of a data set's most significant structures while preserving privacy under the sole assumption that the internal random coins of the algorithm are secret.

We quantify the notion of utility (accuracy) needed for the algorithms' analysis and give rigorous accuracy bounds. Our experiments show that the algorithms perform well on a standard suite of data sets. Our algorithms are based on different techniques, but have similar performance guarantees. Nevertheless, they are incomparable: one is more accurate, the other simpler.

Quantifying "Utility" for FPM. Because differentially private algorithms must treat nearby inputs indistinguishably, they can at best return approximate answers. Thus, our algorithms produce a noisy list of itemsets which is "close" to the list of the top K itemsets with high probability. (Our algorithms also release the approximate frequency of each of the itemsets in the output.)

To quantify the algorithms' utility, we introduce a natural notion of approximation for frequent itemset mining. Roughly, we require that the itemsets in the output have frequencies within a small additive error of those of the K most frequent itemsets. Specifically, let f_K be the frequency of the K^{th} most frequent itemset in the input. Given an accuracy parameter $\gamma \in [0, 1]$, we require that with high probability *(a)* every itemset with frequency greater than $f_K + \gamma$ is output and *(b)* no itemset with frequency below $f_K - \gamma$ is output. Equivalently, the algorithm must output the top-K itemsets of an input in which all frequencies have been changed by up to γ.

In typical data sets, there is little concentration of patterns at any particular frequency. In such cases, our utility guarantee implies low false positive and false negative rates. For example, if there at most $0.02 \cdot K$ itemsets with frequencies in the range $[f_K, f_K + \gamma]$, then with high probability the FNR is at most 2%.

Evaluating Utility. We present a rigorous analysis of the privacy and accuracy of both algorithms. For a given level of differential privacy, quantified by the parameter ϵ, we prove high-probability bounds on how far the reported itemsets can be from the true top-K itemsets. The error parameter γ of both algorithms is $O(K \log(U)/n\epsilon)$, where K is the number of itemsets reported, n is the total number of records in the transaction data set and U is the total number of itemsets under consideration (*e.g.,* for sets of ℓ items among m possibilities, U is $\binom{m}{\ell}$ and $\log(U)$ is $O(\ell \log m)$).

We also provide a extensive experimental evaluation of both algorithms on all the data sets from the FIMI repository [1]. First, we calculate the concrete bounds implied by our utility theorems, and find that the bounds correspond to meaningful error rates on the FIMI data sets. The empirical error we observe in experiments is even lower than the theoretical bounds. Our results indicate that for all but one FIMI data set, we can release over 100 frequent itemsets while keeping the false negative rate below 20%. We present the results in detail in Section 4.

Evaluating Efficiency. In both our algorithms, there is a preprocessing phase which extracts the top $K' > K$ itemsets using an existing non-private algorithm (*Apriori*, [2, 15]). The preprocessing phase takes time roughly proportional to $K'n$, where n is the number of records in the database. Here K' is the number of itemsets with frequency greater than $f_K - \gamma$, where γ is the utility parameter. After preprocessing, both of our algorithms require time only $O(K' + K \log K' + nK)$ to produce the final output. Since K and K' are typically much smaller than n, the non-private itemset mining is the efficiency bottleneck. This observation was borne out by our experiments.

Techniques. The main difference between our two algorithms is technique. Our first algorithm is based on the *exponential mechanism* of McSherry and Talwar [20]. Our main contribution is to give an efficient algorithm for this case of the exponential mechanism (a priori, the mechanism requires exponential time). The second algorithm is based on a new analysis for the established technique of adding Laplace noise to released functions [9, 18, 14]; we show that in some settings one can add much less noise than was possible with previous analyses. A more detailed discussion of our techniques relative to previous work can be found in Sections 5 and 6.

The paper is organized as follows. In Sec. 2 we review the definition of Differential Privacy. Our new privacy preserving frequent itemset mining algorithms are presented in Sec. 3. The experimental evaluation of these methods is presented in Sec. 4. We extend our ideas to a more general problem called *private ranking* in Sec. 5. We review related work in Sec. 6 and conclude in Sec. 7.

2. DIFFERENTIAL PRIVACY

Our algorithms satisfy *differential privacy* [9], which bounds the effect that any single record has on the distribution of the released information. Let $\mathcal{D}^n$ be the space of transaction data sets containing n transactions.

DEFINITION 1 (ϵ-DIFFERENTIAL PRIVACY [9]). *A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if for all transaction data sets $T, T' \in \mathcal{D}^n$ differing in at most one transaction and all events $\mathcal{O} \subseteq Range(\mathcal{A})$:*

$$\Pr[\mathcal{A}(T) \in \mathcal{O}] \leq e^{\epsilon} \Pr[\mathcal{A}(T') \in \mathcal{O}] .$$

Both algorithms presented in this paper satisfy ϵ-differential privacy. In Sec. 6 we also discuss some algorithms that satisfy a weaker notion called (ϵ, δ)-differential privacy [22].

DEFINITION 2 ((ϵ, δ)-DIFFERENTIAL PRIVACY [22]). *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all transaction data sets $T, T' \in \mathcal{D}^n$ differing in at most one transaction and all events $\mathcal{O} \subseteq Range(\mathcal{A})$:*

$$\Pr[\mathcal{A}(T) \in \mathcal{O}] \leq e^{\epsilon} \Pr[\mathcal{A}(T') \in \mathcal{O}] + \delta .$$

Both these definitions capture the notion that the probability of seeing a particular output does not depend too much on any particular transaction. However, definition 2 additionally allows a small additive error factor of δ.

Example: Laplace noise. Differentially private algorithms must be randomized, since they must blur the distinction between two neighboring inputs T, T' even when T and T' are known to the adversary. A common technique to introduce randomness is the addition of Laplace noise to outputs. Suppose that we would like to release (an approximation to) a vector of real-valued statistics. That is, for some function $f : \mathcal{D}^n \to \mathbb{R}^d$, we would like to release an approximation to close to $f(T)$. Dwork *et al.* [9] showed that it suffices to add noise proportional to the *sensitivity* of the function f. Sensitivity measures the maximum possible change in the value of f when transaction from the data set is changed.

DEFINITION 3 (SENSITIVITY [9]). *The sensitivity of a function $f : \mathcal{D}^n \to \mathbb{R}^d$ is the smallest number Δf such that for all inputs $T, T' \in \mathcal{D}^n$ which differ in a single entry (transaction), $||f(T) - f(T')||_1 \leq \Delta f$.*

Consider the randomized algorithm $\mathcal{A}_f$ that computes $f(T)$ and releases $\tilde{f}(T) = f(T) + Lap\left(\frac{\Delta f}{\epsilon}\right)^d$, where $Lap(\lambda)^d$ denotes a vector of d i.i.d. samples from the Laplace distribution $Lap(\lambda)$. Recall that $Lap(\lambda)$ is the distribution on $\mathbb{R}$ with density at y given by $\frac{1}{\lambda}\exp(-|y|/\lambda)$. Dwork *et al.* [9] showed that A_f is ϵ-differentially private. The standard deviation of $Lap(\lambda)$ is $\lambda\sqrt{2}$, so this algorithm adds noise proportional to $\Delta f/\epsilon$.

Noise addition is not directly relevant to FIM because the output cannot be described by a single low-sensitivity real-valued function. However, we will use this technique for reporting the frequencies of the itemsets we output.

Recently McSherry et al. [20] proposed a technique, the *exponential mechanism*, for designing differentially private algorithms for non-real valued outputs. In the next section we discuss this mechanism in detail; we adapt their technique to FIM in our first algorithm (Section 3.1).

We provide two differentially private algorithms for top-K FIM with provable privacy and utility guarantees. Since the algorithms are randomized in nature, we cannot provide the exact solution to the FIM problem. Hence, with high probability, we want to output a list of itemsets that is close to the list of K most frequent itemsets in the transaction data set. "Close" here means roughly that the itemsets in the output have frequencies within a small additive error of those of the K most frequent itemsets. We formalize this notion in the following section.

3. PRIVATE FIM ALGORITHMS

The output of frequent itemset mining algorithms is typically a list of itemsets together with their supports or frequencies. Modifying such an algorithm to satisfy differential privacy requires introducing uncertainty into the output. There are two natural approaches to doing this: we can first construct a noisy list of itemsets (*i.e.* by including some 'infrequent' sets in the list, while leaving out some 'frequent' ones) and then perturb the frequencies of those itemsets, or we can first add noise to the frequencies of all itemsets and then select the itemsets with the highest noisy frequencies. In this paper, we present algorithms which illustrate each of these approaches. Our first algorithm is based on the exponential mechanism of [20]; the second, on the Laplace noise model of [9].

To quantify our algorithms' utility, we introduce a natural notion of approximation for frequent itemset mining. Given an input data set T, the *true frequency* of an itemset refers to the proportion of records in T in which the itemset actually occurs; in contrast, the *reported*, or *noisy*, frequency refers to the estimate reported by the algorithm.

DEFINITION 4 (APPROXIMATE TOP-K FIM). *Let T be a set of n transactions over an alphabet M of m items. Let K denote the number of frequent itemsets to be reported in the output and let ℓ denote the size of itemsets under consideration. Let f_K denote the frequency of the K^{th} most frequent itemset of size ℓ. For positive real parameters ρ, γ, η, we say an algorithm is (ρ, γ, eta)-*useful *if, with probability at least $(1-\rho)$, the output is a list of K itemsets of size ℓ along with estimated frequencies and satisfies:*

1. *(Soundness) No itemset in the output has true frequency less than $(f_K - \gamma)$.*
2. *(Completeness) Every itemset with true frequency greater than $(f_K + \gamma)$ is in the output.*
3. *(Accuracy) For every pattern in the output list, the noisy frequency reported differs by no more than η from the corresponding true frequency.*

3.1 Exponential Mechanism-based Algorithm

In this section we describe the *exponential mechanism* due to McSherry et al. [20] and show how it can be adapted, with some work, to FIM. The exponential mechanism is in fact a family of algorithms, parametrized by a finite set $\mathcal{R}$ of possible outputs (called the *range*) and a real-valued function $q : \mathcal{D}^n \times \mathcal{R} \times \mathbb{R}$ that assigns each possible output r a *score* $q(T, r)$ based on the input T. Given $\mathcal{R}$, q, T and ϵ, the goal is to produce an output with as high a score as possible, while satisfying ϵ-differential privacy. To this end, the algorithm draws a single sample from the distribution on $\mathcal{R}$ which assigns each element $r \in \mathcal{R}$ mass proportional to $\exp(\epsilon q(T, r)/2\Delta q)$. Here Δq is the maximum of the sensitivities (Def. 3) of the functions $q(\cdot, r)$. That is, Δq is the maximum over r and neighboring data sets T, T' of $|q(r, T) - q(r, T')|$. Intuitively, the mechanism is useful since high mass to elements r with high scores. McSherry and Talwar showed that this algorithm is ϵ-differentially private.

At a high level, our first algorithm consists of K applications of the exponential mechanism. In each round, we sample from the set of itemsets of size ℓ. Given a dataset T, the score of an itemset is a truncated version of its frequency, denoted $\hat{f}$. The analysis of privacy relies on bounding the sensitivity of the truncated frequency.

Algorithm 1 Exponential Mechanism based FIM

Input: Transaction data set T, privacy parameter ϵ, itemset length ℓ, K, f_K, and error parameter γ.

1: **Preprocessing:** Using *FIM algorithm*, find all length ℓ itemsets with frequencies $> \psi = f_K - \gamma$. Assume all unknown frequencies to be ψ. Call these frequencies as truncated frequencies.
2: **Sampling:** Sample K itemsets without replacement such that $Pr[\text{Selecting itemset } \mathcal{I}] \propto \exp(\frac{\epsilon n}{4K}\widehat{f}(\mathcal{I}))$, where $\widehat{f}(\mathcal{I})$ is the truncated frequency of $\mathcal{I}$.
3: **Perturbation:** Perturb the true frequencies of the itemsets sampled in the previous step by adding $Lap\left(\frac{2K}{\epsilon n}\right)$ noise.
4: **return** The sampled K itemsets and their noisy frequencies.

In Algorithm 1, we describe our exponential mechanism based FIM algorithm. The algorithm takes the transaction data set T, the data set size n, the alphabet size m, the itemset length ℓ, the number of desired patterns K, the privacy parameter ϵ and the confidence parameter ρ as input. In the **Preprocessing** step, γ is computed as $\frac{4K}{\epsilon n}\left(\ln\frac{K}{\rho} + \ln\binom{m}{\ell}\right)$ (see Lemma 5). A FIM algorithm is run with a sufficiently low threshold so as to get at least K itemsets in the output and all itemsets with frequency $\geq f_K - \gamma$. This may require two runs of the FIM algorithm (first to get f_K and the other to get all itemsets with frequency $\geq f_K - \gamma$).

In our algorithm, along with the notions of true frequency and noisy frequency, we have a notion of a truncated frequency. For an itemset with true frequency f, if $f \geq f_K - \gamma$, then its truncated frequency is f, otherwise its truncated frequency is $f_K - \gamma$ *i.e* truncated frequency $\widehat{f} = \max(f, f_K - \gamma)$. In the **Sampling** step, the truncated frequencies are used to sample K itemsets such that the probability of selecting an itemset is proportional to $\exp(\frac{\epsilon n}{4K}\widehat{f}(\mathcal{I}))$. We give details of the sampling in the next section. In the **Perturbation** step, the true frequencies of the K sampled itemsets are perturbed by a zero mean Laplace noise with parameter $\frac{2K}{\epsilon n}$ before being output. In order to compute the true frequencies of all the K itemsets, in the worst case, $O(K \cdot n)$ of additional work may be required. The noise addition step itself has complexity $O(K)$.

3.1.1 Implementation details and runtime analysis

Let $K'(> K)$ denote the number of itemsets mined by the FIM algorithm in the **Preprocessing** step. A trivial lower bound on the runtime of the FIM algorithm is $\Omega(K'n)$. This is because for every itemset it mines, it has to go through the entire data set once to compute its frequency. We show that FIM runtime is the dominant term in the overall runtime of the algorithm. The **Perturbation** step has a worst-case runtime of $O(K \cdot n)$. Next, we analyze the complexity of the **Sampling** step.

In any particular round of sampling, let $\mathcal{S}_1$ be the collection of itemsets with true frequencies $> f_K - \gamma$ and $\mathcal{S}_2$ be the collection of itemsets with true frequencies $\leq f_K - \gamma$. Note that, we sample without replacement, hence, the sets change with each round of sampling. For any itemset $\mathcal{I} \in \mathcal{S}_1$, the associated probability mass is $\frac{1}{C}\exp(\frac{\epsilon n \widehat{f}(\mathcal{I})}{4K})$, where the normalization constant $C = \sum_{\mathcal{I}\in\mathcal{S}_1}\exp(\frac{\epsilon n \widehat{f}(\mathcal{I})}{4K}) + |\mathcal{S}_2|\exp(\frac{\epsilon n(f_K-\gamma)}{4K})$. The total probability mass associated with the itemsets in $\mathcal{S}_2$ is $\frac{|\mathcal{S}_2|}{C}\exp(\frac{\epsilon n(f_K-\gamma)}{4K})$.

A simple implementation of the **Sampling** step is to partition the real number line $[0, 1]$ into $|S_1| + 1$ segments (one each for an itemset in S_1 and the last one for all itemsets in S_2) according to the probability masses defined above. We then sample a number uniformly at random within the interval $[0, 1]$. The partition in which the random number falls decides the itemset that we pick. If it falls in the partition corresponding to $\mathcal{S}_2$, we pick up an itemset from $\mathcal{S}_2$ uniformly at random. This technique is inefficient because every time an itemset is picked, one has to recompute the probability masses for all the itemsets. In fact the time complexity is $O(K \cdot K')$. One can, in fact, significantly improve the running time.

LEMMA 1. *The **Sampling** step of algorithm 1 can be implemented to run in time $O(K' + K\ln(K'))$.*

PROOF. (Sketch) The idea is to create a static, balanced binary tree with $|S_1| + 1 = K' + 1$ leaves. Each leaf is labeled by a set in $|S_1|$ except for the last leaf which represents all the itemsets in S_2. The weight of a leaf is initially set to be its sampling weight. At each internal node, we store the sum of the weights in the subtree rooted at the node. This data structure can be built in linear time $O(K')$. It allows one to sample from the exponential mechanism's distribution in time $O(\log(K'))$, since one can descend the tree from the root, at each step choosing a child with mass proportional to its subtree weight. Once a leaf has been sampled, its weight can be set to 0; updating the subtree wights on the path to the root also takes time $O(\log(K')$. Since we take K samples, the overall run time is $O(K' + K\log(K'))$. □

In our experiments, we used a simpler linked-list variant of the data structure (figure: 1) from Lemma 1, which performed well on our data sets though it has poor worst-case performance.

Let $\{1, \cdots, U\}$ denote a set of elements, where the probability of picking the i-th element is proportional to A_i. We sort the elements by weight so that $A_1 \geq A_2 \geq \cdots A_U$. We want to sample K elements from this set without replacement. We create a linked list, where the i-th node stores $P_i = \frac{A_i}{\sum_{i\leq j\leq U} A_j}$. To pick an element, we traverse the list starting at node 1. When at the i-th node, we pick element i with probability P_i and stop or we move to node $i + 1$ with probability $1 - P_i$. Thus, the probability of picking an element i in a traversal is equal to $(1-P_1)\cdots(1-P_{i-1})P_i$, which equals $\frac{A_i}{\sum_{1\leq j\leq U} A_j}$. After we have picked an element i, we remove that node from the linked list. We also recompute the P_i's for nodes $1, \cdots, i$ by removing A_i from their expressions. We start the next round of sampling in an exactly same manner as the previous round of sampling, but this time with the new linked list. We repeat this process K times. If the A_i's are highly skewed (i.e. the difference between the consecutive A_i's are quite large) then, effectively in each round of sampling one has to go a small depth in the linked list before an element is picked.

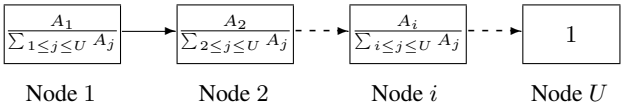


Figure 1: Link list for sampling without replacement

In our setting, $U = |S_1| + 1$. We set each of the A_i's ($i \in [1, |S_1|]$) to $\exp(\frac{\epsilon n \widehat{f}(\mathcal{I})}{4K})$ sorted in descending order, where $\mathcal{I} \in \mathcal{S}_1$. $A_{|S_1|+1}$ is set to $|\mathcal{S}_2| \cdot \exp(\frac{\epsilon n(f_K-\gamma)}{4K})$. In our experiments, the frequencies of the itemsets were highly skewed it was not necessary to go far down the linked list (on average) before an itemset was picked.

From Theorem 1, we know that the **Sampling** step can be implemented in time $O(K' + K\ln(K'))$. Further, the **Perturbation** step takes $O(Kn)$ running time. Therefore, in total steps 2 and 3 of the algorithm runs in $O(K' + K\ln(K') + Kn)$. Earlier in the analysis we saw that the **Preprocessing** of the algorithm takes time $\Omega(K'n)$. Hence, we can conclude that for data sets with reasonably large n, the **Preprocessing** step is the performance bottleneck.

3.1.2 Privacy Analysis

In this section, we prove that algorithm 1 is ϵ differentially private. First, we claim that the sensitivity of truncated frequency of any itemset is bounded

LEMMA 2. *For any itemset $\mathcal{I}$, the truncated frequency of $\mathcal{I}$ has sensitivity $\frac{1}{n}$.*

PROOF. Let T and T' be two transaction data sets with n transactions differing in only one transaction. Let $f_T(\mathcal{I})$ and $\widehat{f}_T(\mathcal{I})$ represent the true frequency and the truncated frequency of an itemset $\mathcal{I}$ in T respectively. We will represent the K-th highest frequency in T as f_K^T and the K-th highest frequency in T' as $f_K^{T'}$. Note, $f_K^T = \theta$ implies that no more than $K - 1$ itemsets have frequency $> \theta$ in T, as well as, that atleast K itemsets (including the itemset which has frequency exactly θ) have frequency $\geq \theta$.

We first prove that f_K^T and $f_K^{T'}$ differ by at most $\frac{1}{n}$. Let $f_K^T = \theta$ and $f_K^{T'} = \theta - \frac{2}{n}$. If $f_K^T = \theta$, then there are altleast K itemsets in T with frequency $\geq \theta$. These K itemsets have a frequency $\geq \theta - \frac{1}{n}$ in T'. This violates the fact that no more than $K - 1$ itemsets have a frequency $> \theta - \frac{2}{n}$ in T'. A similar contradiction arises for any $f_K^{T'} < \theta - \frac{2}{n}$. Thus $f_K^{T'} \geq f_K^T - \frac{1}{n}$. Let $f_K^T = \theta$ and $f_K^{T'} = \theta + \frac{2}{n}$. If $f_K^T = \theta$, then there are no more than $K - 1$ itemsets in T with frequency $> \theta$. Thus, there are no more than $K - 1$ itemsets with frequency $\geq \theta + \frac{1}{n}$ in T'. This violates the fact that atleast K itemsets have a frequency $\geq \theta + \frac{2}{n}$ in T'. A similar contradiction arises for any $f_K^{T'} > \theta + \frac{2}{n}$. Thus $f_K^{T'} \leq f_K^T + \frac{1}{n}$.

Next we prove that $\widehat{f}_T(\mathcal{I})$ and $\widehat{f}_{T'}(\mathcal{I})$ differ by at most $\frac{1}{n}$. For an itemset $\mathcal{I}$ in T whose true frequency is $\geq f_K^T - \gamma + \frac{2}{n}$, its truncated frequency (in both T and T') is same as its true frequency. As true frequencies differ by at most $\frac{1}{n}$ between T and T', $\widehat{f}_T(\mathcal{I})$ and $\widehat{f}_{T'}(\mathcal{I})$ can differ by at most $\frac{1}{n}$. For an itemset $\mathcal{I}$ in T whose true frequency is $\leq f_K^T - \gamma - \frac{2}{n}$, its truncated frequency is $f_K^T - \gamma$ in

$f_K^{T'} - \gamma \rightarrow$ $f_{T'}(\mathcal{I}) \downarrow$	$f_K^T - \gamma - \frac{1}{n}$	$f_K^T - \gamma$	$f_K^T - \gamma + \frac{1}{n}$
$f_K^T - \gamma$ $f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{2}{n}$	$f_K^T - \gamma$ $f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{2}{n}$	$f_K^T - \gamma$ $f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{2}{n}$	$f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{2}{n}$

Table 1: Value of $\widehat{f}_{T'}(\mathcal{I})$ as a function of $f_{T'}(\mathcal{I})$ and $f_K^{T'}$

$f_K^{T'} - \gamma \rightarrow$ $f_{T'}(\mathcal{I}) \downarrow$	$f_K^T - \gamma - \frac{1}{n}$	$f_K^T - \gamma$	$f_K^T - \gamma + \frac{1}{n}$
$f_K^T - \gamma - \frac{2}{n}$ $f_K^T - \gamma - \frac{1}{n}$ $f_K^T - \gamma$	$f_K^T - \gamma - \frac{1}{n}$ $f_K^T - \gamma - \frac{1}{n}$ $f_K^T - \gamma$	$f_K^T - \gamma$ $f_K^T - \gamma$ $f_K^T - \gamma$	$f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{1}{n}$ $f_K^T - \gamma + \frac{1}{n}$

Table 2: Value of $\widehat{f}_{T'}(\mathcal{I})$ as a function of $f_{T'}(\mathcal{I})$ and $f_K^{T'}$

T and $f_K^{T'} - \gamma$ in T'. Note that γ is identical in T and T'. Thus, $\widehat{f}_T(\mathcal{I})$ and $\widehat{f}_{T'}(\mathcal{I})$ can differ by at most $f_K^T - f_K^{T'}$ which is $\leq \frac{1}{n}$.

For an itemset $\mathcal{I}$, whose true frequency in T is exactly $f_K^T - \gamma + \frac{1}{n}$, the truncated frequency in T is $f_K^T - \gamma + \frac{1}{n}$. The truncated frequency of $\mathcal{I}$ in T' depends on both the true frequency of $\mathcal{I}$ and $f_K^{T'} - \gamma$ in T'. Table 1 shows the possible values of the truncated frequency of $\mathcal{I}$ in T' as a function of $f_{T'}(\mathcal{I})$ (along y-coordinate) and $f_K^{T'} - \gamma$ (along x-coordinate). As can be seen from the table, $|\widehat{f}_T(\mathcal{I}) - \widehat{f}_{T'}(\mathcal{I})| \leq \frac{1}{n}$. Similarly, for an itemset $\mathcal{I}$ with true frequency in T exactly $f_K^T - \gamma - \frac{1}{n}$, the truncated frequency in T is $f_K^T - \gamma$. Table 2 shows the possible values of the truncated frequency of $\mathcal{I}$ in T' as a function of $f_{T'}(\mathcal{I})$ (along y-coordinate) and $f_K^{T'} - \gamma$ (along x-coordinate). Again, $|\widehat{f}_T(\mathcal{I}) - \widehat{f}_{T'}(\mathcal{I})| \leq \frac{1}{n}$. A similar exercise for an itemset $\mathcal{I}$ with true frequency $f_K^T - \gamma$ in T shows that $|\widehat{f}_T(\mathcal{I}) - \widehat{f}_{T'}(\mathcal{I})| \leq \frac{1}{n}$. Therefore, always $\widehat{f}_T(\mathcal{I})$ and $\widehat{f}_{T'}(\mathcal{I})$ can differ by at most $\frac{1}{n}$. □

The **Sampling** step is essentially K successive applications of the exponential mechanism. In each round of exponential mechanism an itemset is sampled without replacement. The score function for an itemset $\mathcal{I}$ is $n \times$ truncated frequency of $\mathcal{I}$. Hence, by lemma 2 the sensitivity of the score function is one. From the analysis of the exponential mechanism (explained in the begining of section 3.1), each round of the **Sampling** step guarantees $\frac{\epsilon}{2K}$-differential privacy. We use the composition lemma (defined below) to guarantee $\frac{\epsilon}{2}$ differential privacy for the **Sampling** step as a whole.

LEMMA 3 (COMPOSITION LEMMA [9]). *If a randomized algorithm $\mathcal{A}$ runs K algorithms $\mathcal{A}_1, \cdots, \mathcal{A}_K$, where each algorithm is ϵ_i-differentially private, and outputs $(\mathcal{A}_1(T), \cdots, \mathcal{A}_K(T))$, then $\mathcal{A}(T)$ is $\sum_{i=1}^K \epsilon_i$-differentially private. Here T is any transaction data set.*

In the **Perturbation** step, we use the Laplace noise addtion technique (described in section 2) independently on the true frequencies of the K itemsets chosen in the **Sampling** step. The scaling parameter for the Laplace distribution used is $\frac{2K}{\epsilon n}$. Each of the noise addition step guarantees $\frac{\epsilon}{2K}$-differential privacy. By the use of composition lemma, the **Perturbation** step as a whole is $\frac{\epsilon}{2}$-differentially private.

We guarantee ϵ-differential privacy for algorithm 1 by applying composition lemma on the the **Sampling** and the **Perturbation** step together.

THEOREM 1. *Algorithm 1 is ϵ-differentially private.*

3.1.3 Utility Analysis

In this section, we provide theoretical guarantees for the utility of our algorithm. Intuitively, Theorem 2 guarantees that with high probability, the K itemsets output by our algorithm are close to the actual top K itemsets. Theorem 3 guarantees that with high probability, the reported frequencies of the itemsets output are close to their true frequencies. The main steps of the proof of Theorem 2 are stated here as Lemmas 4, 5, 6.

LEMMA 4. *At each round of sampling during the **Sampling** step, if there exists an unsampled itemset with true frequency $\geq f$, then the probability of picking any itemset with true frequency $\leq f - \gamma$ is at most $\binom{m}{\ell} exp\left(-\frac{\epsilon n \gamma}{4K}\right)$.*

PROOF. Conditioned on the fact that an itemset with true frequency f is still present, the probability of picking an itemset with true frequency $\leq f - \gamma$ is $\leq \frac{e^{\frac{\epsilon n (f-\gamma)}{4K}}}{e^{\frac{\epsilon n f}{4K}}} = \exp(-\frac{\epsilon n \gamma}{4K})$.

Since, there are at most $\binom{m}{\ell}$ itemsets with true frequency $\leq f - \gamma$ therefore, by union bound the probability of picking an itemset with true frequency $\leq f - \gamma$ is at most $\binom{m}{\ell} \exp(-\frac{\epsilon n \gamma}{4K})$. □

LEMMA 5. *Let $\mathcal{S}$ be the collection of itemsets sampled in the **Sampling** step. For all $\rho > 0$, with probability at least $1 - \rho$, the true frequencies of all the itemsets in $\mathcal{S}$ are $> f_K - \gamma$, where $\gamma = \frac{4K}{\epsilon n}\left(\ln \frac{K}{\rho} + \ln \binom{m}{\ell}\right)$. When ρ is constant, $\gamma = O\left(\frac{K \cdot \ln K + \ell \cdot \ln m}{\epsilon n}\right)$.*

PROOF. By lemma 4, in any round of sampling the probability of choosing a particular itemset with true frequency $\leq f_K - \gamma$ is at most $\exp(-\frac{\epsilon n \gamma}{4K})$. This is because in each round of sampling we are guaranteed to have at least one itemset with true frequency $\geq f_K$ which has not been sampled yet. Since, there are at most $\binom{m}{\ell}$ itemsets, therefore by union bound, in any round of sampling the probability of choosing any itemset with true frequency $\leq f_K - \gamma$ is at most $\binom{m}{\ell} \exp(-\frac{\epsilon n \gamma}{4K})$.

Further by union bound, in the **Sampling** step the probability of choosing any itemset with true frequency $\leq f_K - \gamma$ is at most $K \cdot \binom{m}{\ell} e^{-\frac{\epsilon n \gamma}{4K}}$.

Let $\rho \geq K \cdot \binom{m}{\ell} e^{-\frac{\epsilon n \gamma}{4K}}$. Then,

$$\begin{aligned} -\frac{\gamma \epsilon n}{4K} &\leq \ln\left(\frac{\rho}{K\binom{m}{\ell}}\right) \\ \Leftrightarrow \frac{\gamma \epsilon n}{4K} &\geq \ln\left(\frac{K\binom{m}{\ell}}{\rho}\right) \\ \Leftrightarrow \gamma &\geq \frac{4K}{\epsilon n}\left(\ln \frac{K}{\rho} + \ln \binom{m}{\ell}\right) \end{aligned}$$

For constant ρ, $\gamma = O\left(\frac{K(\ln K + \ell \cdot \ln m)}{\epsilon n}\right)$ will suffice. □

LEMMA 6. *For all $\rho > 0$, with probability at least $1 - \rho$, all length ℓ itemsets with true frequency $> f_K + \gamma$ are present in $\mathcal{S}$, where $\gamma = \frac{4K}{\epsilon n}\left(\ln \frac{K}{\rho} + \ln \binom{m}{\ell}\right)$. When ρ is constant, $\gamma = O\left(\frac{K(\ln K + \ell \cdot \ln m)}{\epsilon n}\right)$.*

PROOF. If any one of the itemsets with true frequency $> f_K + \gamma$ is not present in $\mathcal{S}$ then, by lemma 4, probability of picking any itemset with true frequency $\leq f_K$ is at most $\binom{m}{\ell} \exp(-\frac{\epsilon n \gamma}{4K})$.

Therefore, the probability of not picking any itemset with true frequency $\leq f_K$ in any of the K rounds of sampling is at least $\left(1-\binom{m}{\ell}e^{-\frac{\epsilon\gamma n}{4K}}\right)^K \geq \left(1-K\cdot\binom{m}{\ell}\exp(-\frac{\epsilon\gamma n}{4K})\right)$.

From the analysis of lemma 5, $\gamma \geq \frac{4K}{\epsilon n}\left(\ln\frac{K}{\rho}+\ln\binom{m}{\ell}\right)$. When ρ is constant, $\gamma = O\left(\frac{K(\ln K+\ell\cdot\ln m)}{\epsilon n}\right)$ will suffice. □

THEOREM 2. *For all $\rho > 0$, with probability at least $1-\rho$, all output itemsets have their true frequencies $> f_K-\gamma$, and all itemsets with true frequency $> f_K+\gamma$ are output, where $\gamma = \frac{4K}{\epsilon n}\left(\ln\frac{2K}{\rho}+\ln\binom{m}{\ell}\right)$. When ρ is constant, $\gamma = O\left(\frac{K(\ln K+\ell\cdot\ln m)}{\epsilon n}\right)$.*

PROOF. From the proof of lemma 5, we know that w.p. at least $1-K\cdot\binom{m}{\ell}e^{-\frac{\epsilon\gamma n}{4K}}$ all itemsets in $\mathcal{S}$ have true frequencies $> f_K-\gamma$.

From the proof of lemma 6, we know that w.p. at least $1-K\cdot\binom{m}{\ell}e^{-\frac{\epsilon\gamma n}{4K}}$ all the length ℓ itemsets with true frequency $> f_K+\gamma$ are present in $\mathcal{S}$.

By union bound, w.p at least $1-2K\cdot\binom{m}{\ell}e^{-\frac{\epsilon\gamma n}{4K}}$ all itemsets output have their true frequencies $> f_K-\gamma$ and all itemsets with true frequency $> f_K+\gamma$ are output.

Using analysis analogous to Lemma 5, we get $\gamma \geq \frac{4K}{\epsilon n}\left(\ln\frac{2K}{\rho}+\ln\binom{m}{\ell}\right)$. For constant ρ, $\gamma = O\left(\frac{K(\ln K+\ell\cdot\ln m)}{\epsilon n}\right)$ will suffice. □

THEOREM 3. *For all $\rho > 0$, with probability at least $1-\rho$, all noisy frequencies differ by at most η from their corresponding true frequencies, where $\eta = \frac{2K}{n\epsilon}\ln\left(\frac{K}{\rho}\right)$.*

PROOF. Let the true frequency of an itemset be f. In the perturbation stage we add $Lap\left(\frac{2K}{\epsilon n}\right)$ noise to f. Therefore, the probability of the noisy frequency deviating by $\geq \eta$ from f is $\geq 2\cdot\left(\frac{n\epsilon}{4K}\int_{f+\eta}^{\infty} exp\left(-\frac{(x-f)n\epsilon}{2K}\right)dx\right). = exp\left(-\frac{\eta n\epsilon}{2K}\right)$

Since we add Laplace noise to K true frequencies, therefore, by union bound the probability of any of the noisy frequencies differing by more than η from their corresponding true frequencies is at most $K\cdot exp\left(-\frac{\eta n\epsilon}{2K}\right)$. Setting, $\rho = K\cdot exp\left(-\frac{\eta n\epsilon}{2K}\right)$, we get $\eta = \frac{2K}{n\epsilon}\ln\frac{K}{\rho}$. □

3.2 Laplace Mechanism based algorithm

The second algorithm we present is easier to implement and understand than the first. The accuracy (utility) bound γ we obtain for the second algorithm slightly worse (by a factor of roughly 2) than the guarantee for the first algorithm. Nevertheless, the second algorithms' simplicity may make it preferable in some settings. Moreover, the anlysis of privacy requires a new proof technique which may be of independent interest.

The basic idea of the algorithm is to add independent Laplace noise to the frequencies of all itemsets and select the K itemsets with the highest perturbed frequencies. A naive sensitivity analysis suggests that we must add noise proportional to $\binom{m}{\ell}/\epsilon$ for this to be ϵ-differentially private. However, we show that it suffices to add noise only $O(K/\epsilon)$ to the frequencies. Additional work is required to get an efficient implementation; in particular, we use the idea of truncated frequencies from the previous algorithm.

3.2.1 Implementation details and runtime analysis

Steps 1 and 3 of the algorithm are straight forward. The **Noise addition and sampling** step requires some thought in order to perform it in a computationally efficient manner. Clearly, it is not computationally feasible to add noise to the truncated frequencies of all $\binom{m}{\ell}$ itemsets as the number of itemsets to be dealt with is large. However, the number of itemsets with true frequencies $> f_K-\gamma$ is within computable limit. Hence, we can add noise to the truncated frequencies of these itemsets. Using the same notation of the previous subsection, $\mathcal{S}_1$ represents itemsets with true frequencies $> f_K-\gamma$ and $\mathcal{S}_2$ represents itemsets with true frequencies $\leq f_K-\gamma$. We only need a special strategy for $\mathcal{S}_2$.

Algorithm 2 Laplace Mechanism based FIM

Input: Transaction data set T, privacy parameter ϵ, itemset length ℓ, K, f_K, and error parameter γ.

1: **Preprocessing:** Using *FIM algorithm*, find all length ℓ itemsets with frequencies $> \psi = f_K-\gamma$. Assume all unknown frequencies to be ψ. Call these frequencies as truncated frequencies.
2: **Noise addition and sampling:** Add $Lap\left(\frac{4K}{\epsilon n}\right)$ to the truncated frequencies of all $\binom{m}{\ell}$ itemsets to obtain the noisy frequencies. Pick the top K itemsets in terms of the noisy frequencies. Let this set be denoted as $\mathcal{S}$. {We will discuss later how to perform this step in a computationally efficient manner.}
3: **Perturbation:** Perturb the true frequencies of the itemsets in $\mathcal{S}$ with fresh $Lap\left(\frac{2K}{\epsilon n}\right)$ noise to obtain the noisy frequencies for the itemsets in $\mathcal{S}$.
4: **return** The set $\mathcal{S}$ and the corresponding noisy frequencies.

Let $\widehat{lfreq}_{\mathcal{S}_1}$ be the Kth largest noisy frequency in the set $\mathcal{S}_1$. Let $\psi = f_K-\gamma$. Now, an itemset whose true frequency is $\leq \psi$, if it has to make it to the final output then its noisy frequency should be greater than $\widehat{lfreq}_{\mathcal{S}_1}$. Therefore, the probability of it making to the final output is $< \frac{1}{2}e^{-\frac{\left|\psi-\widehat{lfreq}_{\mathcal{S}_1}\right|n\epsilon}{4K}}$ if $\widehat{lfreq}_{\mathcal{S}_1} \geq \psi$ and $< 1-\frac{1}{2}e^{-\frac{\left|\psi-\widehat{lfreq}_{\mathcal{S}_1}\right|n\epsilon}{4K}}$ if $\widehat{lfreq}_{\mathcal{S}_1} < \psi$. Let us denote this probability as p. Thus, the number of itemsets with true frequency $< f_K-\gamma$ which has noisy frequency $> \widehat{lfreq}_{\mathcal{S}_1}$ follows a Binomial distribution with parameters $\binom{m}{\ell}-|\mathcal{S}_1|$ and p.

We now pick a random number X according to the Binomial distribution mentioned above and pick X itemsets uniformly at random from the set $\mathcal{S}_2$. For now let us assume that $\widehat{lfreq}_{\mathcal{S}_1} \geq \psi$. In fact almost all the time this will be true. Conditioned on the fact that there are X itemsets with true frequencies $\leq \psi$, whose noisy frequencies are greater than $\widehat{lfreq}_{\mathcal{S}_1}$, the distribution of these X itemsets follow an exponential distribution with mean $\widehat{lfreq}_{\mathcal{S}_1}+\frac{4K}{\epsilon n}$ and standard deviation $\frac{4K}{\epsilon n}$. This follows from the *memorylessness property* of exponential distribution. Thus, the noisy frequencies of these X itemsets are picked i.i.d. from the mentioned exponential distribution. We call the set of these noisy frequencies and the corresponding itemsets $\mathcal{V}$. In the unlikely event of $\widehat{lfreq}_{\mathcal{S}_1} \leq \psi$, we can get a similar distribution using Bayes' Theorem.

Now, we pick the top K itemsets in terms of the noisy frequencies from the set $\mathcal{S}_1 \cup \mathcal{V}$ and pass them on to the the **Perturbation** step. We next discuss about the running time of the algorithm. Let ρ be the confidence parameter (defined earlier). We set the error parameter $\gamma = \frac{8K}{\epsilon n}\left(\ln\left(\frac{\binom{m}{\ell}}{\rho}\right)\right)$. We will see in section 3.3.2, the utility guarantee requires γ to be set to this value. For this value of γ, the following theorem holds true.

THEOREM 4. *With probability at least $1-\rho$, steps 2 and 3 of algorithm 2 runs in time $O(K'+Kn)$, where K' is the number of itemsets mined by the FIM algorithm.*

PROOF. To prove this claim, we will use the result from theorem 6. Theorem 6 is stated and proved in section 3.3.2. From the statement of theorem 6 we know that with probability at least $1-\rho$, no itemset from $\mathcal{S}_2$ is present in the final output. This implies, with probability at least $1-\rho$ the value of the random number X (which denotes the number of itemsets from $\mathcal{S}_2$ whose noisy frequencies are greater than $\widehat{lfreq_{\mathcal{S}_1}}$) is zero. Therefore, in such a situation the **Noise addition and sampling** step will take time $O(K')$. Clearly, the **Perturbation** step takes time $O(Kn)$. Hence, with probability at least $1-\rho$, steps 2 and 3 of algorithm 2 runs in time $O(K'+Kn)$. □

In the runtime analysis of algorithm 1, we have seen that the step that involves the *Apriori algorithm* is usually the performance bottleneck. From the theorem above, we know that with high probablity steps 2 and 3 of algorithm 2 runs in time $O(K'+Kn)$. And earlier we saw that runtime of FIM algorithm is $\Omega(K'n)$. Hence, with high probability even for the present algorithm, *Apriori algorithm* is the performance bottleneck.

3.3 Privacy and Utility Analysis

3.3.1 Privacy guarantee

THEOREM 5. *The algorithm is ϵ-differentially private.*

PROOF. Let $\mathcal{D}^n$ be the domain of data sets of n transactions where each transaction is a subset of M. Let $S_T = \{\langle \mathcal{I}_1, \overline{f}_T(\mathcal{I}_1)\rangle, \cdots, \langle \mathcal{I}_K, \overline{f}_T(\mathcal{I}_K)\rangle\}$ represent the output of the algorithm $\mathcal{A}$ running on data set $T \in \mathcal{D}^n$. Similar to that in the proof of theorem 1, $\mathcal{I}_i \subseteq U$ represents the itemsets and $\overline{f}_T(\mathcal{I}_i)$ represent the corresponding noisy frequencies. We prove the privacy guarantee in two parts. First, we prove that the collection of K itemsets (i.e. $\{\mathcal{I}_1, \cdots, \mathcal{I}_K\}$) sampled after step two of algorithm 2 preserves $\frac{\epsilon}{2}$-differential privacy. Then we prove the $\frac{\epsilon}{2}$ differential privacy for the noisy frequencies output for these particular K itemsets after step three of the algorithm 2. We then argue that by the composability property from lemma 3, the algorithm as a whole is ϵ-differentially private.

Let W denote the collection of the K itemsets output by the algorithm $\mathcal{A}$. Let $T, T' \in \mathcal{D}^n$ be any two data sets differing in exactly one transaction. We want to first show that $\Pr[\mathcal{A}(T) = W] \leq e^{\frac{\epsilon}{2}} \Pr[\mathcal{A}(T) = W]$. This is an abuse of notation as the output of $\mathcal{A}$ is actually the collection of itemsets and their frequencies. For now we will consider just the collection of itemsets it outputs. To denote the intermediate noisy frequency for an itemset $\mathcal{I}$ in step two of the algorithm, we use $\widetilde{f}_T(\mathcal{I})$.

Now,

$$\Pr[\mathcal{A}(T) = W] = \int_{v_1 \in \mathbb{R}} \cdots \int_{v_K \in \mathbb{R}} pdf_T[\widetilde{f}_{\mathcal{I}_1} = v_1] \cdot pdf_T[\widetilde{f}_{\mathcal{I}_K} = v_K] \prod_{\mathcal{I} \in 2^U - W, |\mathcal{I}| = \ell} \Pr_T[\widetilde{f}_{\mathcal{I}} < min\{v_1, v_2, \cdots, v_k\}]$$

We use the notation $pdf_T[\cdot]$, $\Pr_T[]$ to parameterize the probability density function and the probability mass function on data set T.

We want to upper bound the ratio $\frac{\Pr[\mathcal{A}(T)=W]}{\Pr[\mathcal{A}(T')=W]}$ by $e^{\frac{\epsilon}{2}}$. In order to upper bound the ratio by $e^{\frac{\epsilon}{2}}$, we will minimize the denominator. To minimize $\Pr[\mathcal{A}(T') = W]$, $\forall \mathcal{I} \in 2^U, |\mathcal{I}| = \ell$, we can either increase or decrease $\widehat{f}_{\mathcal{I}}(T)$ by $\frac{1}{n}$ to obtain $\widehat{f}_{\mathcal{I}}(T')$, since $\left|\widehat{f}_{\mathcal{I}}(T) - \widehat{f}_{\mathcal{I}}(T')\right|$ can be at most $\frac{1}{n}$ (as discussed in the proof of theorem 1). $\widehat{f}_{\mathcal{I}}(T)$ represent the truncated frequency of an itemset $\mathcal{I}$ in data set T. Thus, to minimize $\Pr[\mathcal{A}(T') = W]$, one has to have $\widehat{f}_{\mathcal{I}}(T') - \widehat{f}_{\mathcal{I}}(T) = \frac{1}{n}$ for all $\mathcal{I} \in 2^U - W, |\mathcal{I}| = \ell$. For all the itemsets $\mathcal{I} \in W$, depending on the value of $\widehat{f}_{\mathcal{I}}(T)$ one has either $\widehat{f}_{\mathcal{I}}(T') - \widehat{f}_{\mathcal{I}}(T) = \frac{1}{n}$ or $\widehat{f}_{\mathcal{I}}(T) - \widehat{f}_{\mathcal{I}}(T') = \frac{1}{n}$ in order to minimize $\Pr[\mathcal{A}(T') = W]$. This is because for any $\mathcal{I} \in 2^U$ and for any $v \in \mathbb{R}$, $pdf_T[\widetilde{f}_{\mathcal{I}} = v] = \frac{1}{2\lambda} e^{-\frac{|v - \widehat{f}_{\mathcal{I}}(T)|}{\lambda}}$. Similarly, $\Pr_T[\widetilde{f}_{\mathcal{I}} < v] = \frac{1}{2} e^{-\frac{|v - \widehat{f}_{\mathcal{I}}(T)|}{\lambda}}$, when $v < \widehat{f}_{\mathcal{I}}(T)$, and $\Pr[\widetilde{f}_{\mathcal{I}} < v] = 1 - \frac{1}{2} e^{-\frac{|v - \widehat{f}_{\mathcal{I}}(T)|}{\lambda}}$, when $v \geq \widehat{f}_{\mathcal{I}}(T)$. Note that $\Pr[\widetilde{f}_{\mathcal{I}} < v]$ decreases when $\widetilde{f}_{\mathcal{I}}$ increases.

One critical observation is that algorithm $\mathcal{A}$ behaves the same (in terms of outputting the itemsets) on a data set $T'' \in D^n$ as it behaves on $T \in D^n$ if all the truncated frequencies of itemsets in T are shifted by $\frac{1}{n}$ in the same direction (i.e. either increase all or decrease all) to form T'', since all that matters are the differences in the truncated frequencies. This property is also known as translation invariance.

Therefore, instead of following the previous procedure, if one increases $\widehat{f}_{\mathcal{I}}$ for all $\mathcal{I} \in W$ which were increased in the previous procedure but this time with $\frac{2}{n}$, increase $\widehat{f}_{\mathcal{I}}$ for all $\mathcal{I} \in W$ which were kept constant in the previous procedure by $\frac{1}{n}$ and for all $\mathcal{I} \in W, |\mathcal{I}| = \ell$ whose truncated frequencies were decreased in the previous procedure, keep the same. Also keep the truncated frequencies of $\mathcal{I} \in 2^U - W, |\mathcal{I}| = \ell$ same. In this way the two procedures of obtaining $\Pr[\mathcal{A}(T') = W]$ are exactly identical. Thus, when we obtain $\Pr[\mathcal{A}(T') = W]$, we need to only know about the change of $\widehat{f}_{\mathcal{I}}$ for all $\mathcal{I} \in W, |\mathcal{I}| = \ell$. As we saw in the previous step, this change can be at max $\frac{2}{n}$.

For an itemset $\mathcal{I} \in W$, $\frac{pdf_T[\widetilde{f}_{\mathcal{I}} = v]}{pdf_{T'}[\widetilde{f}_{\mathcal{I}} = v]}$ is at most $e^{\frac{2}{n\lambda}}$ (since we are changing $\widehat{f}_{\mathcal{I}}$ by at most $\frac{2}{n}$). Since, in each term in the integration of the expression for $\Pr[\mathcal{A}(T') = W]$ there are exactly K terms which has $\mathcal{I} \in W$, therefore, when we change from T to T' each term in the integration changes by at most $\frac{2K}{n\lambda}$. Therefore, $\frac{\Pr[\mathcal{A}(T)=W]}{\Pr[\mathcal{A}(T')=W]}$ is upper bounded by $\frac{2K}{n\lambda}$.

Hence, setting $\lambda = \frac{4K}{n\epsilon}$ guarantees $\frac{\epsilon}{2}$-differential privacy for the **Noise addition and sampling** step of the algorithm.

Since, the **Perturbation** step of both the algorithms 1 and 2 are same hence, the privacy guarantees for this step in both are also same. The **Perturbation** step assures that the set of the noisy frequencies output for the itemsets sampled in the **Noise addition and sampling** step is $\frac{\epsilon}{2}$-differentially private.

Hence by the composition lemma 3 , the algorithm as a whole is ϵ-differentially private. □

3.3.2 Utility guarantee

In this subsection we provide utility guarantees which are analogous to the ones presented in the exponential mechanism based approach. The utility guarantee in theorem 6 is at most two times worse than the utility guarantee in theorem 2.

THEOREM 6. *For all $\rho > 0$: with probability at least $1-\rho$, all itemsets output have their true frequencies $> f_K - \gamma$, and all itemsets with true frequency $> f_K + \gamma$ are output, where $\gamma = \frac{8K}{\epsilon n}\left(\ln\left(\frac{\binom{m}{\ell}}{\rho}\right)\right)$. When ρ is constant, $\gamma = O\left(\frac{K\ell \ln(m)}{\epsilon n}\right)$.*

PROOF. Since, we are adding $Lap\left(\frac{4K}{n\epsilon}\right)$ noise to all the truncated frequencies, it can be shown that with probability at least $1 - \binom{m}{\ell} \cdot e^{-\frac{\epsilon\Gamma n}{4K}}$ all itemsets of length ℓ have their noisy frequencies within γ margin of their truncated frequencies.

Data set	n	m	$\|t\|$
accidents	340183	469	34.81
chess	3196	76	38
connect	67557	130	44
kosarak	990002	41270	8.09
mushroom	8124	120	24
pumsb	49046	2114	75
pumsb-star	49046	2089	51.48
retail	88162	16471	11.31
T10I4D100K	100000	871	11.1
T40I10D100K	100000	943	40.61

Table 3: Data sets characteristics: Number of transactions N, size of alphabet m, average size of transaction, $|t|$.

Therefore, if we set $\gamma = 2\Gamma$, then with probability at least $1 - \binom{m}{\ell} \cdot e^{-\frac{\epsilon\gamma n}{8K}}$, all itemsets output have their true frequencies $> f_K - \gamma$ and all itemsets with true frequencies $> f_K + \gamma$ are output. Thus if we set $\rho = \binom{m}{\ell} \cdot e^{-\frac{\epsilon\gamma n}{8K}}$, then $\gamma = \frac{8K}{\epsilon n}\left(\ln\left(\frac{\binom{m}{\ell}}{\rho}\right)\right)$ suffices. For constant ρ, $\gamma = O\left(\frac{K\ell\ln(m)}{\epsilon n}\right)$. □

THEOREM 7. *For all $\rho > 0$, with probability at least $1 - \rho$, all noisy frequencies differ by at most η from their corresponding true frequencies, where $\eta = \frac{2K}{n\epsilon}\ln\left(\frac{K}{\rho}\right)$.*

PROOF. The proof is exactly the same as that for theorem 3. □

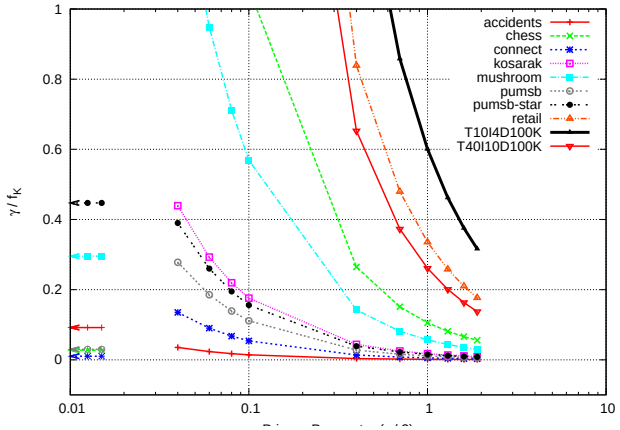
4. EXPERIMENTS

In this section, we present the results of several experiments we performed to evaluate the performance of the above proposed algorithms. We first describe the data sets on which we ran our algorithms. Then we present the relationships between the different parameters (eg. ϵ, γ, ρ, η) that we obtain by applying the theoretical results to these data sets. We also study extensively the utility of our algorithms for these data sets under a wide range of parameters.

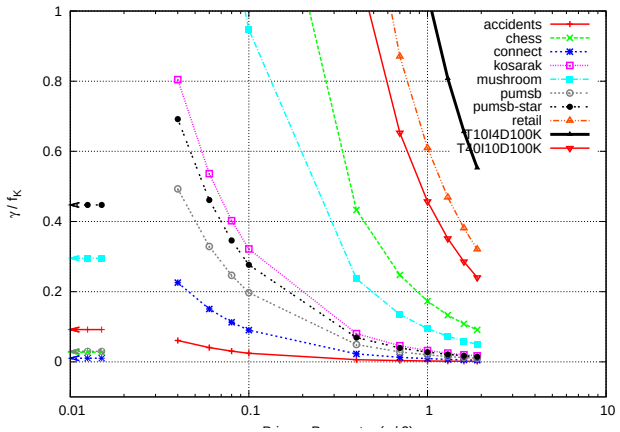
For the evaluation of our experiments we use data sets publically available at the FIMI repository `http://fimi.helsinki.fi`. These data sets are listed in Table 4. This collection of data sets contain both real-world and synthetic data sets, and also have widely varying characteristics like number of transactions n, number of items m and average transaction length $|t|$.

Summary of the results:

a) *Theoretical guarantees result in useful parameter ranges on these data sets* - We show that our theorems about privacy and utility, when applied to these data sets yield a useful range for all the parameters of the system. In particular, the efficiency of our algorithms greatly depends on the threshold at which the underlying frequent itemset mining (FIM) algorithm runs. The threshold we provide to the FIM algorithm is $f_K - \gamma$. A small γ implies that the privacy overhead in terms of the running time of the FIM algorithm is not too high. We plot $\frac{\gamma}{f_K}$ as a function of $\frac{\epsilon}{2}$, ℓ and K. These plots tell us how low a threshold we have to provide to the FIM algorithm for various choices of other parameters. Our plots show that for most data sets, at typical values of the parameters ($\frac{\epsilon}{2} = 0.7$, $\rho = 0.1$, $l = 3$, $k = 10$), γ is a small fraction of f_K. The other theoretical guarantee that we provide is about η, which is the difference between the reported frequencies of the output itemsets and their true frequencies. For these data sets, we show that the actual value of η obtained is a small fraction of f_K. Note that we plot variation of γ and η against $\frac{\epsilon}{2}$ to emphasize that the final privacy parameter is ϵ when both the patterns and their frequency are output.



(a) Algorithm 1



(b) Algorithm 2

Figure 2: Variation of $\frac{\gamma}{f_K}$ with the privacy parameter $\frac{\epsilon}{2}$

b) *For a wide range of parameters, the algorithms give good utility on these data sets* - For the same set of parameter ranges as in (a), we run our algorithm on these data sets and plot the False Negative Rate (FNR) for the output. Note that False Positive Rate (FPR) is essentially small for this output as the number of infrequent patterns are typically high compared to the total number of frequent patterns (since $K << \binom{m}{l}$). In these data sets, the highest possible FPR that can be achieved is 0.03 (this is assuming that all the top K itemsets are false positives). Our plots show that, again for typical values of the parameters, FNR is under 0.2 for eight data sets (while for 6 of them it is close to 0.02).

In our first set of experiments, we study the behaviour of $\frac{\gamma}{f_K}$ and $\frac{\eta}{f_K}$ as other parameters ϵ, K and ℓ vary. For these experiments, $\frac{\epsilon}{2}$ varies from 0.04-2, K varies from 10-100 and ℓ varies from 2-6. In an experiment, while one parameter varies, the other three remain fixed. These fixed values are $\frac{\epsilon}{2} = 0.7$, $\rho = 0.1$, $K = 10$ and $l = 3$. Figure 2(a) shows the plot of $\frac{\gamma}{f_K}$ as $\frac{\epsilon}{2}$ varies from 0.04 - 2 (note x axis is in log scale) for *Algorithm 1*. We clamp the y-axis at 1, as $\frac{\gamma}{f_K}$ greater than 1 implies a negative FIM threshold, that is, $f_K - \gamma < 0$. Whenever the theoretical requirement causes $f_K - \gamma$ to

become negative, one of the utility guarantees (namely, soundness) becomes trivial. Also note, that when $\frac{\gamma}{f_K}$ becomes greater than $\frac{1}{f_K}-1$, $f_K+\gamma$ becomes greater than 1. In this case, the other utility guarantee (namely, completeness) becomes trivial. Thus whenever $\frac{\gamma}{f_K}$ is less than $\min(1, \frac{1}{f_K}-1)$, both the utility guarantees are non-trivial. In the figures, the arrowhead on the y-axis indicate $\frac{1}{f_K}-1$ for each data set. For some data sets, $\frac{1}{f_K}-1$ is greater than 1, thus it does not show up in the plots. For algorithm 1, at $\frac{\epsilon}{2}=0.7$, for all data sets except chess and T10I4D100K, both the utility guarantees (soundness and completeness) are non-trivial as the obtained $\frac{\gamma}{f_K}$ is less than $\min(1, \frac{1}{f_K}-1)$. As expected, the ratio $\frac{\gamma}{f_K}$ decreases as the privacy requirement (ϵ) is relaxed. Figure 3(b) shows the variation of $\frac{\gamma}{f_K}$ as K varies from 10-100. It can be observed that $\frac{\gamma}{f_K}$ rises rapidly for data sets which have either a large alphabet size m (eg. retail) or a low f_K (eg. T10I4D100K and T40I10D100K) or a small n (eg. chess and mushroom). Note that for kosarak the rise is not that rapid despite having a big m as n is also quite large for it. Figure 3(c) shows the variation of $\frac{\gamma}{f_K}$ as ℓ varies from 2-6. The trend in this plot is quite similar to the one in 3(b).

In the same set of experiments we also study the noise added to the frequencies of the output items. We show the variation of $\frac{\eta}{f_K}$ with $\frac{\epsilon}{2}$. In figure 4(a), we see that at $\frac{\epsilon}{2}=0.7$, the ratio $\frac{\eta}{f_K}$ goes below 0.1 for all data sets. We skip the plots of $\frac{\eta}{f_K}$ v/s ρ and $\frac{\eta}{f_K}$ v/s K due to lack of space.

In our next set of experiments we study the False Negative Rates produced in the output as we vary the parameters over the same ranges as in the earlier set of experiments. The underlying FIM algorithm employed was the "fp-growth" version of Ferenc Bodon's implementation (http://www.cs.bme.hu/ bodon/en/apriori/). It was run on a machine with an Intel(R) Xeon(R) CPU E5345 @2.33 GHz with 16 GB of RAM. In our experiments, we found the running time of the underlying FIM algorithm as the dominant term in the overall running time. Thus, to have a reasonable check on the running time of the complete experiment, we decided to discard all experiment runs in which the underlying FIM algorithm ran for more than 5 minutes or produced a pattern output file of size greater than 5GB. Thus, if for a certain choice of parameters, the $f_K-\gamma$ value was such that the FIM algorithm run violated the above constraints, we don't report the FNR. This does not mean that our algorithms fail to terminate for that choice of parameters. Infact, under such stringent computational constraints, the algorithms continue to provide good results for a wide range of parameters. Each FNR reading reported in this set of experiments is averaged over 10 runs of the experiment. The standard deviation in the FNR was always under 0.15 for all data sets.except for the T10I4D100K data set in the FNR v/s ρ plot, where it was 0.2. We don't show the standard deviations to make the plots more readable.

Figure 5(a) shows the plot of FNR against $\frac{\epsilon}{2}$. At $\frac{\epsilon}{2}=0.7$, except data sets chess and T10I4D100K, all others have a FNR of under 0.2. In fact for most data sets (6 of them) the FNR is close to 0.02. We skip FNR v/s ρ plot due to lack of space. In Figs. 6(a)-6(b) the FNR seems to rise with increasing K or ℓ. Note, for some data sets including T10I4D100K, T40I10D100K, chess and retail, there are a lot of missing points as the underlying FIM algorithm run violated our computational constraints often. For all other data sets, the FNR continues to remain low.

5. GENERAL RANKING

The algorithms 1 and 2 proposed for FIM naturally extends to any generic problem on ranked list. Following [20], a ranked list is a list of elements ordered according to some measure of interest. Instead of considering a universe of itemsets of length ℓ drawn from an item base of size m, we can consider an universe of U elements where each element has a score associated with it. Let the universe of elements be represented as $\mathcal{S}=\{\mathcal{E}_1,\cdots,\mathcal{E}_U\}$. Let $T\in\mathcal{D}^n$ be a transaction database of n transactions, where each row is a subset of $\mathcal{S}$. Let $q:\mathcal{S}\times\mathcal{D}^n.\rightarrow\mathbb{R}$ be a function which assigns score to each element. The score function is analogous to the frequency of an itemset in FIM. The goal in this abstract setting is to output the top K elements in terms of the scores assigned by the function q. As in the case of differentially private FIM, here also we have the error parameters γ and η, and the confidence parameter ρ. Let Δq be the sensitivity of the function q, i.e. the amount by which the function changes if one row of the database T is changed. Recall that the sensitivity of the frequency function in the case of FIM is $\frac{1}{n}$. In the algorithms and the associtated privacy and utility guarantees in section 3, if we replace the size of the universe of itemsets (i.e. $\binom{m}{\ell}$) by the size of the universe of elements (i.e. $|\mathcal{S}|=U$), replace the frequency function by q and replace the sensitivity of the frequency function (i.e. $\frac{1}{n}$) by Δq, we obtain algorithms and their associated privacy and utility guarantees for the problem on ranked lists. Note that, the privacy guarantees will remain exactly the same as that of FIM.

6. RELATED WORK

Randomized response.

One approach to Privacy Preserving Data Mining is randomized response. In this approach each entry in the data set is independently randomized before allowing the data mining algorithm to access it. Evfimievski et al. [10] and Agrawal et al. [3] considered randomized response in the context of FIM. They consider the threshold variant where the goal is to return all the itemsets of length ℓ whose frequencies are greater than a predefined threshold θ. They define the term *amplification factor* which quantifies the privacy guarantee of the mining algorithm. The amplification factor directly corresponds to e^ϵ, where ϵ is the differential privacy parameter. The work of [3] is an improvement over the work of [10].

We compare our algorithms 1 and 2 to the algorithms of [3] on the same CENSUS data set used by [3] from the UCI repository `http://archive.ics.uci.edu/ml/`. To enable comparison, we set the parameters of our algorithms as follows: First, we set K so that f_K equals the threshold θ used by [3]. Second, they use amplification factor $e^\epsilon=19$, where as we set it to $e^\epsilon=e^2$ (that is, we impose an even stronger privacy guarantee). Third, we set the confidence parameter ρ for our algorithms to 0.05; there is no analogous parameter in [3].

To measure utility, [3] used the false negative rate (FNR). We compared the FNR of our algorithms to those of the two best-performing algorithms from [3] (RAN-GD and DET-GD) for various itemset lengths; the results are plotted in Figure 7. We find that both of our algorithms have consistently lower FNR. The FNR for RAN-GD and DET-GD were taken from Agrawal et al. [3, Figures 1(a) and 2(a)].

Privacy preserving search log release.

Götz et al. [14] and Korolova et al. [18] independently presented algorithms for releasing search log statistics in a differentially private manner. Both the algorithms are very similar to each other. We can adapt the algorithms to provide differentially private algorithms for FIM.

It is difficult to compare the performance of these two algorithms

against our algorithms because they were optimized for the search log setting.

Specifically, the algorithms add $Lap(\lambda)$ noise to frequencies of the itemsets present in the data set and outputs all the noisy frequencies and their corresponding itemsets which exceed a specified threshold τ'. (In contrast, we output the top-K itemsets and add noise independently.)

If we consider the FIM setting, a single transaction can potentially change the frequencies of $\binom{m}{\ell}$ length-ℓ itemsets. In the experimental settings we consider, the value of $\binom{m}{\ell}$ is far higher than the maximum value ω (i.e., the number of elements whose scores change by changing one users data) used by [14] and [18]. In order to make their assumption reasonable for FIM, we impose a bound t on the length of any transaction in the data set. (The length of a transaction is the number of items present in it.) A single transaction can potentially change the frequencies of $\binom{t}{\ell}$ length-ℓ itemsets. We can map the parameter ω from the search log setting to $\binom{t}{\ell}$ in our setting.

Götz et al. [14, Theorem 1] state the value of λ (i.e. the scaling parameter of Laplace noise) and τ' sufficient to guarantee (ϵ, δ)-differential privacy for algorithms by [14] and [18] respectively. ((ϵ, δ)-differential privacy is a relaxation to the definition of ϵ-differential privacy allowing a small additive error of δ.) Adapting this theorem to our setting we get, $\lambda = \frac{2\binom{t}{\ell}}{n\epsilon}$ and $\tau' \geq \frac{1}{n}\binom{t}{\ell}\left(1 - \frac{1}{\epsilon}\ln\left(2\delta/\binom{t}{\ell}\right)\right)$. Table 4 shows the requirement on τ' for the different data sets we have considered in our experiments. We have set $\epsilon = 1, \delta = 0.05$ and $\ell = 3$.

Data set	n	ℓ'	$\tau' \geq$
Accident	340183	52	0.8644
Chess	3196	38	32.5796
Connect	67557	44	2.5081
Kosarak	990002	2498	6.55E+04
Mushroom	8124	24	2.7194
Pumsb-star	49046	64	11.8418
Pumsb	49046	75	19.8569
Retail	88162	77	12.0333

Table 4: Required values for τ'

We find that in all but for the accident data set, τ' is greater than one. In order to output the K most frequent itemsets, we would like to have τ' be at most f_K. This makes the algorithms by Götz et al. and Korolova et al. unreasonable for our experimental setup. Note that in cases where t and ℓ are small, their approach might indeed work well. However, in terms of privacy guarantee they provide (ϵ, δ)-differential privacy guarantee which is strictly worse than the guarantee we provide.

Synthetic data sets.

Blum et al. [5] provided a method to output a synthetic data set $\widetilde{T}$, which provides near accurate answers for frequency queries (i.e. close to the frequencies in the original data set T). This data set can be output in a ϵ-differentially private manner. For $\gamma \geq \tilde{O}\left(\frac{(m\ell)^{1/3}}{(\epsilon n)^{1/3}}\right)$, the utility guarantees for the algorithm due to Blum et al. and our algorithms 1 and 2 are similar. Recall that that in our algorithms, we need $\gamma \geq \tilde{O}\left(\frac{K\ell}{\epsilon n}\right)$. In the experimental settings we consider, $n\epsilon$ is far larger than K, hence the lower bound on γ in our case is better. However, in settings where K is larger than $m^{\frac{1}{3}}(n\epsilon)^{\frac{2}{3}}$, the [5] algorithm gives a better bound on γ. Even in these settings, our approach may be preferable for efficiency. The only known implementation of [5] runs in time $2^{\tilde{O}\left(\frac{m^2}{\epsilon^2}\right)}$, which is impractical with the current computational resources available.

7. CONCLUSIONS

In this paper we presented two efficient differentially private algorithms for top-K frequent pattern mining. In our algorithms we adapted the Exponential Mechanism and the Laplace noise-addtion mechanism by introducing techniques that are efficient in the context of frequent pattern mining. We introduced a new notion of utility for top-K pattern mining and provided theoretical analysis of our methods under this criterion. We also presented extensive experimental results that demonstrate the effectiveness of our methods on the FIMI benchmark data sets. Though we present our algorithms for the problem of frequent pattern mining, our techniques are applicable in the general problem of *private ranking* as well. For example, our algorithms can be used in the settings of [14] and [18], where they analyze the problem of releasing search log statistics privately.

The utility guarantees we provide in theorems 2 and 6 are dependent on the size of the universe of items. In some cases, the universe of items can be large, resulting in large run-times as well as *loose* utility guarantees. A possible future direction is to devise techniques that remove this dependency on the size of the universe of items, thereby extending the applicability of the algorithms to bigger and more complex data sets.

Acknowledgements. A.S. and A.T. are partly supported by NSF grants #0747294, 0729171. We thank Daniel Kifer for helpful comments.

8. REFERENCES

[1] Frequent itemset mining implementations repository. `http://fimi.helsinki.fi`.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–216, May 1993.

[3] S. Agrawal and J. R. Haritsa. A framework for high-accuracy privacy-preserving mining. In *ICDE*, pages 193–204, 2005.

[4] M. Barbaro and T. Zeller. A face is exposed for aol searcher no. 4417749. *The New York Times*, Aug. 2006.

[5] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.

[6] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.

[7] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, pages 43–52, 1999.

[8] C. Dwork. Differential privacy. In *ICALP*, LNCS, pages 1–12, 2006.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[10] A. V. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003.

[11] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *KDD*, pages 265–273, 2008.

[12] N. G.N., B. A., H. J., S. K., and E. I.R. Temporal pattern discovery for trends and transient effects: Its application to patient records. In *Proceedings of the Fourteenth International Conference on Knowledge Discovery and Data Mining SIGKDD 2008*, pages 963–971, 2008.

[13] B. Goethals. Survey on frequent pattern mining. Manuscript, 2003.

[14] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Privacy in search logs. *CoRR*, abs/0904.0682, 2009.

[15] J. Han and M. Kamber. *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers, San Fransisco, CA, USA, 2001.

[16] D. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. MIT Press, Cambridge, MA, USA, 2001.

[17] V. Hristidis, editor. *Information Discovery on Electronic Health Records*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Boca Raton, FL, USA, 2009.

[18] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180, 2009.

[19] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.

[20] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[21] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[22] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84, 2007.

[23] P. K. Novak, N. Lavrac, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging patterns and subgroup mining. *Journal of Machine Learning Research*, (10):377–403, 2009.

[24] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[25] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2008.

[26] M. J. Zaki and C.-J. Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17, 2005.

APPENDIX

A. PSEUDOCODES OF SAMPLING STEP FOR ALGORITHMS 1 AND 2

Let $S_{>f_K-\gamma}$ be the set of itemsets and their corresponding true frequencies output by the Apriori algorithm. Let $S_{>f_K-\gamma}(i)$ represent the i-th itemset in the set $S_{>f_K-\gamma}$ and let $freq\left(S_{>f_K-\gamma}(i)\right)$ represent its frequency. We follow this notation for both the algorithms. First we present the pseudocode of the *Sampling* step of Exponential Mechanism based FIM in algorithm 3.

Next we present the pseudocode for *Noise addition and sampling* step in algorithm 4

Algorithm 3 Sampling step of Exponential Mechanism based FIM

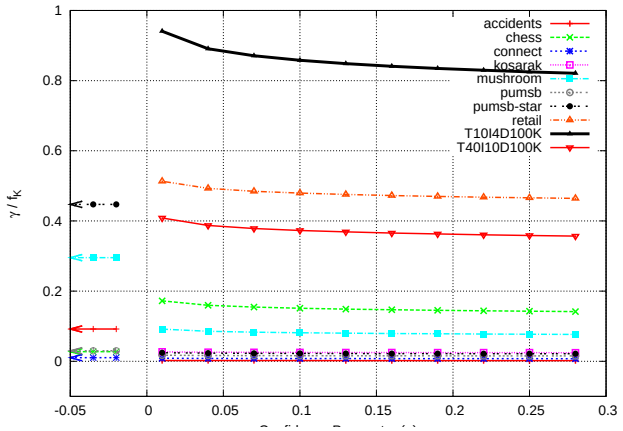
Input: Set $S_{>f_K-\gamma}$, database size n, privacy parameter ϵ, itemset length ℓ, K, f_K, and error parameter γ.

$N \leftarrow |S_{>f_K-\gamma}| + 1$
for $i = 1$ to $N-1$ **do**
  $A_i.itemset \leftarrow S_{>f_K-\gamma}(i)$
  $A_i.freq \leftarrow freq(S_{>f_K-\gamma}(i))$
  $A_i.expData \leftarrow exp\left(\frac{\epsilon n \cdot freq(S_{>f_K-\gamma}(i))}{4K}\right)$
end for
$A_N.itemset \leftarrow lowFreqItems$
$A_N.expData \leftarrow \left(\binom{m}{\ell} - |S_{>f_K-\gamma}|\right) exp\left(\frac{\epsilon n(f_K-\gamma)}{4K}\right)$
Sort the array $A[1, \cdots, N-1]$ in descending order on the member variable $expData$
Create a doubly linked list L with N nodes such that any node L_i stores A_i and $X_i = \sum_{i \le j \le N} A_j.expData$
$FORBIDDEN \leftarrow \emptyset$
$OUTPUT \leftarrow \emptyset$ {Initialize the Output set}
for $i = 1$ to K **do**
  $flag \leftarrow FALSE$
  $j \leftarrow 1$
  while $flag == FALSE$ **do**
    Generate $Y \sim Bernoulli(\frac{A_j.expData}{X_j})$
    if $N == j$ **then**
      $flag \leftarrow TRUE$
      Sample uniformly at random an itemset $\mathcal{I}$ from $Universe - (S_{>f_K-\gamma} \cup FORBIDDEN)$, where $Universe$ is the collection of all length ℓ itemsets
      $FORBIDDEN \leftarrow FORBIDDEN \cup \mathcal{I}$
      $OUTPUT.itemset \leftarrow \mathcal{I}$
      $OUTPUT.freq \leftarrow f_K - \gamma$
      Update $A_N \leftarrow A_N - exp\left(\frac{\epsilon n(f_K-\gamma)}{4K}\right)$
      Update $\forall 1 \le q \le N, X_q \leftarrow X_q - exp\left(\frac{\epsilon n(f_K-\gamma)}{4K}\right)$
    else if $1 == Y$ **then**
      $OUTPUT.itemset \leftarrow A_j.itemset$
      $OUTPUT.freq \leftarrow A_j.freq$
      Update $\forall 1 \le q < j, X_q \leftarrow X_q - A_j.expData$
      Remove Node L_j and decrease N by 1
      $flag \leftarrow TRUE$
    end if
    $j \leftarrow j + 1$
  end while
end for
return The set $OUTPUT$

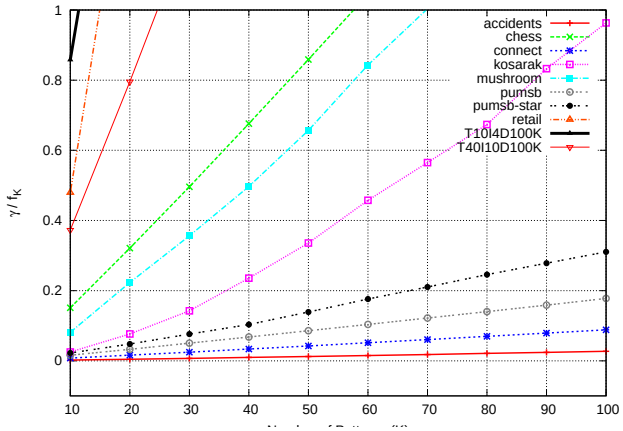
Algorithm 4 Noise addition and sampling step of Laplace Mechanism based FIM

Input: Set $S_{>f_K-\gamma}$, database size n, privacy parameter ϵ, itemset length ℓ, K, f_K, and error parameter γ.

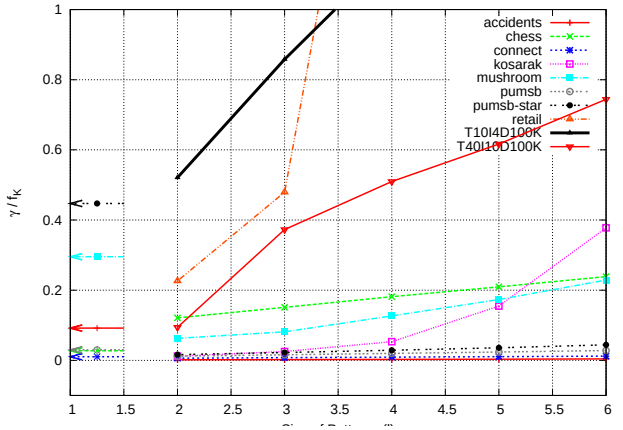
$N \leftarrow |S_{>f_K-\gamma}|$
$X \leftarrow \emptyset$
$\psi \leftarrow f_K - \gamma$
for $i = 1$ to N **do**
  $X_i.itemset \leftarrow S_{>f_K-\gamma}(i)$
  $X_i.freq \leftarrow freq(S_{>f_K-\gamma}(i))$
  $X_i.noisyFreq \leftarrow X_i.freq + Lap\left(\frac{4K}{\epsilon n}\right)$
end for
$lFreq \leftarrow K$-th highest noisy frequency in X
if $lFreq \geq \psi$ **then**
  $p \leftarrow \frac{1}{2}e^{-\frac{|\psi - lFreq|n\epsilon}{4K}}$
else
  $p \leftarrow 1 - \frac{1}{2}e^{-\frac{|\psi - lFreq|n\epsilon}{4K}}$
end if
$Y \sim Binom\left(\binom{m}{\ell} - N, p\right)$
$FORBIDDEN \leftarrow \emptyset$
for $i = N + 1$ to $N + 1 + Y$ **do**
  Sample uniformly at random an itemset $\mathcal{I}$ from $Universe - (S_{>f_K-\gamma} \cup FORBIDDEN)$, where $Universe$ is the collection of all length ℓ itemsets
  $FORBIDDEN \leftarrow FORBIDDEN \cup \mathcal{I}$
  $X_i.itemset \leftarrow \mathcal{I}$
  $X_i.freq \leftarrow \psi$
  $X_i.noisyFreq \sim$ Exponential distribution with mean $lFreq + \frac{4K}{\epsilon n}$ and standard deviation $\frac{4K}{\epsilon n}$
end for
Set $OUTPUT$ to top$-K$ of the elements from X in terms of the noisy frequency
return The set $OUTPUT$



(a) $\frac{\gamma}{f_K}$ v/s ρ

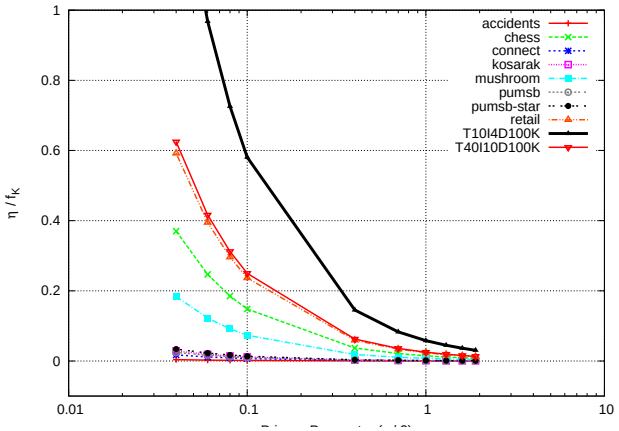


(b) $\frac{\gamma}{f_K}$ v/s K



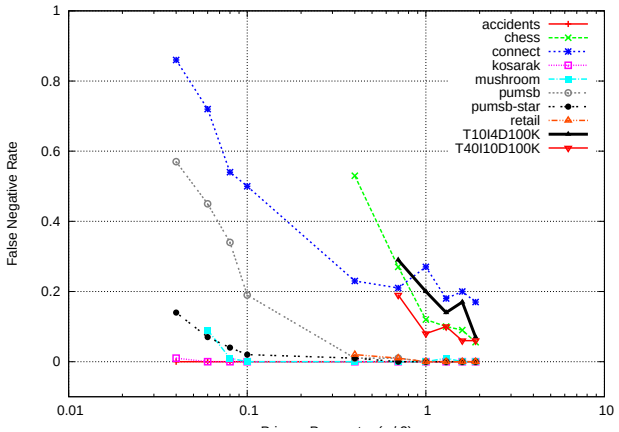
(c) $\frac{\gamma}{f_K}$ v/s ℓ

Figure 3: Variation of $\frac{\gamma}{f_K}$, for algorithm 1,as confidence ρ,number of patterns K and size of patterns ℓ vary.

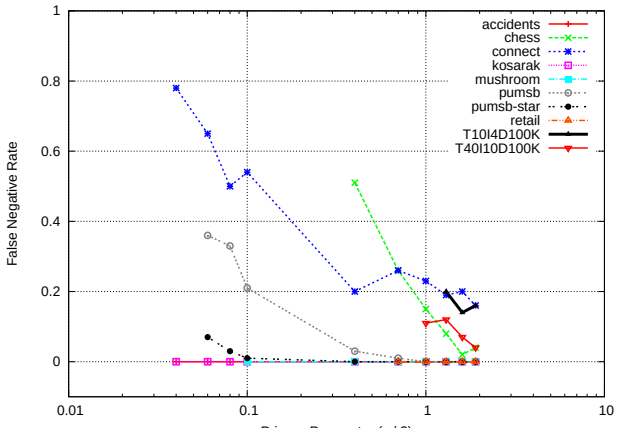


(a) $\frac{\eta}{f_K}$ v/s $\frac{\epsilon}{2}$

Figure 4: Variation of ($\frac{\eta}{f_K}$) under algorithms 1 and 2as DP parameter $\frac{\epsilon}{2}$, confidence parameter ρ and number of patterns Kvary.

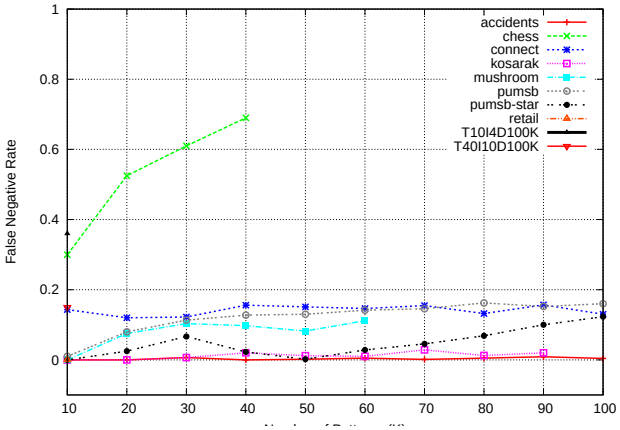


(a) Algorithm 1

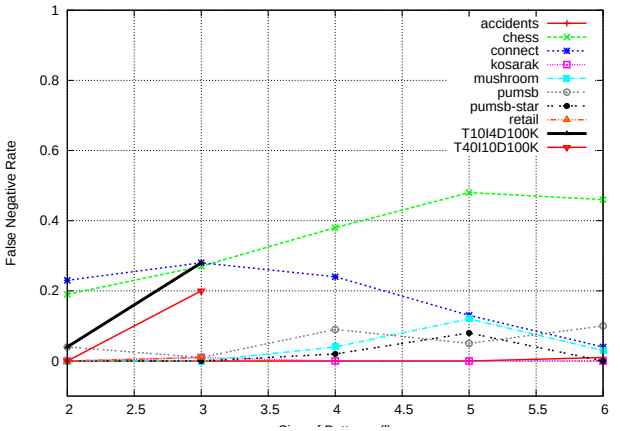


(b) Algorithm 2

Figure 5: Variation of FNR as epsilon varies: FNR v/s $\frac{\epsilon}{2}$



(a) FNR v/s K



(b) FNR v/s ℓ

Figure 6: Variation of FNR under algorithm 1as K and ℓ are varied.

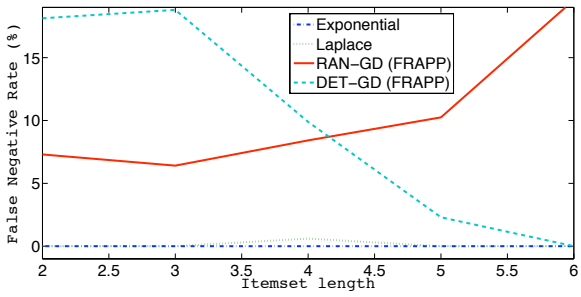


Figure 7: FNR obtained while comparing our algorithms 1 and 2 with the FRAPP framework