ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 22, April 2016

Privacy-Preserving Private Frequent Itemset Mining via Smart Splitting

M.Nandhini[1]
PG Scholar,
Department of Computer Science,
Tagore Institute of Engineering and Technology,
Attur,Salem -636112
Email:mvnandhu@gmail.com

P.Madhubala[2]
Assistant Professor,
Department of Computer Science,
Tagore Institute of Engineering and Technology,
Attur,Salem -636112

Dr.K.Murugesan[3]
Principal,Professor ,Depaertment of Electronics and Communication Engineering,
Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College,
Avadi,Chennai-600062

ABSTRACT: FIM (Frequent Itemset Mining) is one of the problem in data mining. Here we designed private FIM algorithm cannot be achieve high data utility, and high degree of privacy, and also offer high time of efficiency. In exist,Apriori and FP growth are used. In this paper we proposed the PFP growth algorithm to the smart splitting method. In PFP growth can be consist of two phase (i.e. pre-processing and mining phase).In the pre-processing phase,we transform the database to limit the length of transactions. The pre-processing phase is irrelevant to user specified thresholds and needs to be performed only once for a given database. We argue, to enforce such a limit, long transactions should be split rather than truncated. That is, if a transaction has more items than the limit, we divide it into multiple subsets (i.e., sub-transactions) and guarantee each subset is under the limit. In the mining phase, given the transformed database and a user-specified threshold, we privately discover frequent itemsets. During the mining process, we dynamically estimate the number of support computations, so that we can gradually reduce the amount of noise required by differential privacy. In the mining phase, to offset the information loss caused by transaction splitting, we devise a run-time estimation method to estimate the actual support of itemsets in the original database.

I. INTRODUCTION

Frequent itemset mining (FIM) is one of the most fundamental problems in data mining. It has practical importance in a wide range of application areas such as decision support, Web usage mining, bioinformatics, etc. Given a database, where each transaction contains a set of items, FIM tries to find itemsets that occur in transactions more frequently than a given threshold. Despite valuable insights the discovery of frequent itemsets can potentially provide, if the data is sensitive (e.g., web browsing history and medical records), releasing the discovered frequent itemsets might pose considerable threats to individual privacy. Differential privacy has been proposed as a way to address such problem. Unlike the anonymization-based privacy models (e.g., k-anonymity and l-diversity), differential privacy offers strong theoretical guarantees on the privacy of released data without making assumptions about an attacker's background knowledge. In particular, by adding a carefully chosen amount of noise, differential privacy assures that the output of a computation is insensitive to changes in any individual's record, and thus restricting privacy leaks through the results. A variety of algorithms have been proposed for mining frequent itemsets. The Apriori and FP-growth are the two most prominent ones. In particular, Apriori is a breadth first search, candidate set generation-and-test algorithm. It needs l database scans if the maximal length of frequent itemsets is l. In contrast, FP-growth is a depth-first search algorithm, which requires no candidate generation. Compared with Apriori, FP-growth only performs two database scans, which makes FP-growth an order of magnitude faster than Apriori. The appealing features of FP-growth motivate us to design a differentially private FIM algorithm based on the FP-growth algorithm. In this paper, we argue that a practical differentially private FIM algorithm should not only achieve high data utility and a high degree of privacy, but also offer high time efficiency. Although several differentially private FIM algorithms have been proposed, we are not aware of any existing studies that can satisfy all these requirements simultaneously. The resulting demands inevitably bring new challenges. It has been shown that the utility-privacy tradeoff can be improved by limiting the length of transactions. Existing work presents an Apriori-based differentially private FIM algorithm. It enforces the limit by truncating transactions (i.e., if a transaction has more items than the limit, deleting items until its length is under the limit). In particular, in each database scan, to preserve more frequency information, it leverages discovered frequent itemsets to re-truncate transactions. However, FP-growth only performs two database scans. There is no opportunity to re-truncate transactions during the mining process. Thus, the transaction truncating approach proposed is not suitable for FP-growth. In addition, to avoid privacy breach, we add noise to the support of itemsets.

Given an i-itemset X (i.e., X contains i items), to satisfy differential privacy, the amount of noise added to the support of i-itemset X depends on the number of support computations of i-itemsets. Unlike Apriori, FPgrowth is a depth-first search algorithm. It is hard to obtain the exact number of support computations of i-itemsets during the mining process. A naive approach for computing the noisy support of i-itemset X is to use the number of all possible i-itemsets. However, it will definitely produce invalid results.

II. RELATED WORK

Lots of studies have been proposed to solve the privacy preserving FIM problem from different aspects.Considering K- anonymity model for protecting privacy in [2], [12] propose an algorithm to publish annonymised frequent itemset. However these two studies do not satisfies differential privacy. And thus they cannot provide sufficient privacy protection against attackers with background knowledge. [3] A new novel and powerful privacy definition called l –diversity. [3] Show the weak points of k-anonymity; how it is weaken to protect information against attacker with background knowledge. Diversity framework introduced here to give strong privacy guarantee. [4] Proposed fast algorithm for mining association rule i.e. Apriori & AprioriHybrid algorithms .These compared with previous algorithms and these algorithm gives excellent performance for large database with transactions, but these generates candidate set. [5] Introduces FP growth algorithm, with is nothing but mining frequent pattern without candidate generation, as we have seen in [4] apriori algorithm performs mining fastly with candidate set generation, which is costly. In [5] FP tree is used as data structure to store large database compressed in small data structure. Algorithm introduced in [5] is scalable and efficient than apriori algorithm. [11] Present set of randomization operators to limit privacy beaches in FIM. [13] Proposed new algorithm for discovering frequent patterns in sensitive data adopted exponential mechanism & Laplace noise-addition mechanism techniques which are efficient in context of frequent item mining. Christo Ananth et al. [6] discussed about a method, Wireless sensor networks utilize large numbers of wireless sensor nodes to collect information from their sensing terrain. Wireless sensor nodes are battery-powered devices. Energy saving is always crucial to the lifetime of a wireless sensor network. Recently, many algorithms are proposed to tackle the energy saving problem in wireless sensor networks. There are strong needs to develop wireless sensor networks algorithms with optimization priorities biased to aspects besides energy saving. In this project, a delay-aware data collection network structure for wireless sensor networks is proposed based on Multi hop Cluster Network. The objective of the proposed network structure is to determine delays in the data collection processes. The path with minimized delay through which the data can be transmitted from source to destination is also determined. AODV protocol is used to route the data packets from the source to destination. [14] Proposes algorithm Privbasis with perform frequent itemset mining with differential privacy by using minimum support threshold. Any itemset that occurs in transaction often than minimum support threshold is subset of some basis with differential privacy guarantee.

But [13] [14] addresses some issues performing frequent item mining with differential privacy.

Differential Privacy:-

For 2 databases D & D', they are neighbouring databases if they differ by at most one record.

Є –Differential Privacy:

A private algorithm A satisfies Є –Differential Privacy Iff for any two neighbouring databases D & D' and any subset of outputs S is subset of Range (A),

$$Pr\ [A\ (D) \in S] \leq e\epsilon * Pr\ [A\ (D') \in S]$$

Where probability is taken over the randomness of A.

Frequent Itemset mining:

Itemset nothing but collection of one or more items & Frequent items means itemsets whose support is greater than or equal to minimum support threshold. Finding such frequent itemset is called frequent item mining.

TABLE 1

A Simple Transaction Database

TID	Items
101	f,a,b,c
102	b,c,d
103	d,a,b,c,f
104	h,d,g
105	a,f,g
106	e,f,h

III. KEY METHODS

In this section, we propose three key methods to address the challenges in designing a differentially private FIM algorithm based on the FP-growth algorithm. In particular, to limit the length of transactions without introducing much information loss, we propose our *smart splitting* method. Moreover, to offset the information loss caused by transaction splitting, a *run-time estimation* method is used to estimate the actual support of itemsets in the mining process. Furthermore, to lower the amount of added noise, we develop a *dynamic reduction* method which dynamically reduces the sensitivity of support computations by decreasing the upper bound on the number of support computations.

Smart Splitting: To improve the utility-privacy tradeoff, we argue that long transactions should be split rather than truncated. That is, we transform the database by dividing long transactions into multiple subsets (i.e., sub-transactions), each of which meets the maximal length constraint.

For example, assume itemsets *{a, b, c}* and *{d, e, f}* are frequent and the maximal length constraint is 4. Given a transaction *t = {a, b, c, d, e, f}*, if we simply truncate *t* to be *{a, b, c, d}*,the support of itemset *{d, e, f}* and its subsets will all decrease. Consequently, some iemsets which are frequentin the original database may become infrequent. Instead, if we divide *t* into *t*1 = *{a, b, c}* and *t*2 = *{d, e, f}*, the support of itemsets *{a, b, c}*, *{d, e, f}* and their subsets will not be affected.

Theorem: Let *A* be an ϵ-differentially private algorithm for the transformed database and ***f*** be a function that can divide one transaction into at most *k* subsets. Then, for any neighboring databases *D* and *D'*, and any subset of outputs $S \subsetneq Range(A)$, we have:

$\Pr(A(f(D)) = S) \le ek \cdot \epsilon \Pr(A(f(D')) = S)$.

Proof: Consider two neighboring databases *D* and *D′*. Let *t* denote the transaction in *D′* but not in *D* (i.e., *D′=D+t*). Suppose the transformed database of *D* is *˜D* and *t* is divided into *k* subsets *t*1, ..., *tk*. Since *A* is an ϵ-differentially private algorithm for the transformed database *˜D*, based the definition of differential privacy, for any subsets of outputs S ⊆ Range (A), we have:

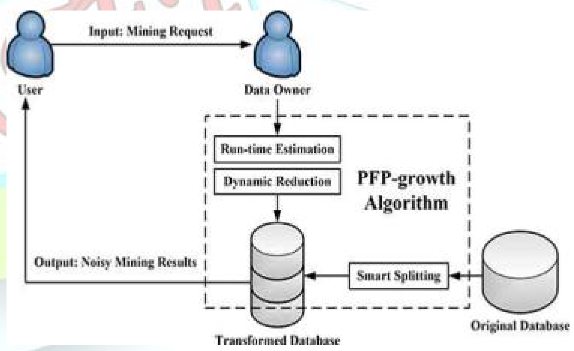
$\Pr(A(\tilde{D}) = S) \le e\epsilon \Pr(A(\langle \tilde{D}, t1 \rangle) = S)$.

Similarly, we can prove that:

$\Pr(A(\tilde{D}) = S) \le ek \cdot \epsilon \Pr(A(\langle \tilde{D}, t1, ..., tk \rangle) = S)$.

Since *˜D* is the transformed database of *D* and *t* is divided into *t*1, ..., *tk*, $\langle \tilde{D}, t1, ..., tk \rangle$ can be considered as the transformed database of *D′*. The theorem then follows.

To this end, we introduce the weighted splitting operation. When we divide a long transaction, we assign a weight to each generated subset. The weight of a subset indicates the change to the support of an itemset when adding (removing) this subset into (from) the database. It can be considered as a multiplier.



Run-time Estimation: Runtime estimation method to quantify the information loss caused by transaction splitting. Such information loss comes from two aspects. Suppose a transaction *t={a, b, c, d}* is divided into *t*1=*{a, b}* and *t*2=*{c, d}* with weight *w*1, *w*2 respectively. On the one hand, assigning weights makes the support of itemsets *{a, b}* and *{c, d}* decrease from 1 to *w*1 and *w*2. On the other hand, splitting *t* causes the support of some itemsets, such as itemset *{a, c}*, decreases from 1 to 0. To offset the information loss caused by transaction splitting, inspired by the double standards method in [7], we propose the *run-time estimation* method. The method consists of two steps: based on the noisy support of an itemset in the transformed database, 1) we first estimate its actual support in the transformed database, and 2) then we further compute its actual support in the original database.

For an itemset X, let ω denote its noisy support in the transformed database and ω' denote its actual support in the transformed database.

Based on the Bayesian rule, we have

$\Pr(\omega'|\omega) = \Pr(\omega|\omega') \cdot \Pr(\omega') / \Pr(\omega)$.

For the information loss caused by transaction splitting,it also depends on how the items in a transaction are partitioned into subsets. Our smart splitting method utilizes the CR-tree to guide the splitting process. However, due to the privacy requirement, we cannot use the CR-tree to quantify the information loss.

Our run-time estimation method only depends on differentially private information. In particular, in the first step, to get the probability distribution of ω', we only need the noisy support ω.

In our PFP-growth algorithm, we use the run-time estimation method in the following manner. Suppose the conditional pattern base of itemset *Y*, *DY*, is currently being mined. When we obtain the noisy support of an item *i* in *DY*, we first estimate the average support of *i* in *DY* (i.e., the average support of itemset $\{Y \cup i\}$). If this average support exceeds the threshold, we output itemset $\{Y \cup i\}$ as a frequent itemset. Then, we further estimate the maximal support of *i* in *DY* (i.e., the maximal support of itemset $\{Y \cup i\}$). If this maximal support exceeds the threshold, we insert item *i* into *Y's* header table and generate the conditional pattern base of itemset $\{Y \cup i\}$.

Dynamic Reduction: As the method is performed in the mining process,we should ensure the method would not incur much computational overhead. Our main idea is to leverage the downward closure property (i.e., the supersets of an infrequent itemset are infrequent), and dynamically reduce the sensitivity of support computations by decreasing the upper bound on the number of support computations.

Then,based on the obtained noisy support, by using our run-time estimation method, we estimate the "maximal" support of itemset $\{Y \cup i\}$. If the estimated "maximal" support is smaller than the threshold, we regard *i* as infrequent items in *DY* . Next, we decrease the upper bounds based on the infrequent items found in *DY*. Let *S*2 denote the infrequent items found in *DY*. For each item $j \in S2$, itemset $Z = \{Y \cup j\}$ is infrequent. Based on the downward closure property, it is unnecessary to compute the support of the itemsets which are the concatenations of *Z* with any subsets of $\{S1 - j\}$.

PFP Growth Algorithm

The PFP-growth algorithm consists of two phases. In particular,in the preprocessing phase, we extract some statistical information from the original database and leverage the smart splitting method to transform the database. Notice that, for a given database, the preprocessing phase is performed only once. In the mining phase, for a given threshold, we privately find frequent itemsets. The run-time estimation and dynamic reduction methods are used in this phase to improve the quality of the results.

We divide the total privacy budget ϵ into five portions:

$\epsilon 1$ is used to compute the maximal length constraint,

$\epsilon 2$ is used to estimate the maximal length of frequent itemsets,

$\epsilon 3$ is used to reveal the correlation of items within transactions,

$\epsilon 4$ is used to compute μ-vectors of itemsets, and

$\epsilon 5$ is used for the support computations.

Pre processing Phase:

In the preprocessing phase, we transform the database to limit the length of transactions. The preprocessing phase is irrelevant to user specified thresholds and needs to be performed only once for a given database. To enforce such a limit, long transactions should be split rather than truncated. That is, if a transaction has more items than the limit, we divide it into multiple subsets.

Mining Phase:

In the mining phase, given the transformed database and a user-specified threshold, we privately discover frequent item sets. During the mining process, we dynamically estimate the number of support computations, so that we can gradually reduce the amount of noise required by differential privacy. In the mining phase, to offset the information loss caused by transaction splitting, we devise a **r**un-time estimation method to estimate the actual support of itemsets in the original database. Runtime estimation method to quantify the information loss caused by transaction splitting.

IV. EXPERIMENTAL SETUP

Effect of Transaction Splitting

FP-growth only performs two database scans. There is no opportunity to re-truncate transactions during the mining process. Thus, the transaction truncating approach proposed in is not suitable for FP-growth. Unlike Apriori, FPgrowth is a depth-first search algorithm. It is hard to obtain the exact number of support computations of *i*-itemsets during the mining process. It will definitely produce invalid results.It is hard to obtain the exact number of support computations of *i*-itemsets during the mining process.

Effect of Smart Splitting

To address these challenges, we present our private FPgrowth (PFP-growth) algorithm, which consists of a preprocessing phase and a mining phase. In the preprocessing phase, we transform the database to limit the length of transactions. The preprocessing phase is irrelevant to user specified thresholds and needs to be performed only once for a given database. We argue, to enforce such a limit, long transactions should be split rather than truncated. That is, if a transaction has more items than the limit, we divide it into multiple subsets (i.e., sub-transactions) and guarantee each subset is under the limit. In the mining phase, given the transformed database and a user-specified threshold, we privately discover frequent itemsets.

Preprocessing and Running time

The preprocessing phase does not consume too much time. For the splitting process, we can see it is efficient and the consumed time is mainly affected by the number of divided transactions. For PFP, the running time does not include the preprocessing time, as the preprocessing phase is only performed once for each dataset and is irrelevant to the thresholds. We can see PFP gains comparable performance with FP-growth. It indicates the run-time estimation and dynamic reduction methods do not incur much overhead during the mining process.

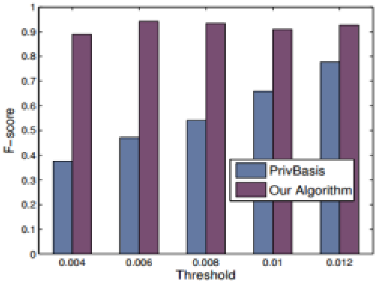
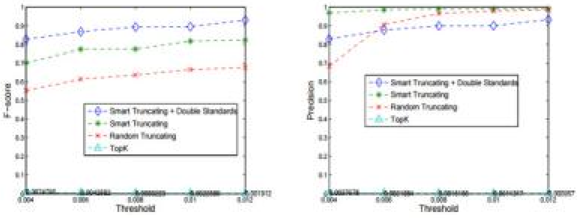


Fig 4.Effect of F-Score and Threshold



(a) F-score (b) Precision

V. CONCLUSION

In this project, we investigate the problem of designing a differentially private FIM algorithm. We propose our private FP-growth (PFP-growth) algorithm, which consists of a preprocessing phase and a mining phase. In the preprocessing phase, to better improve the utility-privacy tradeoff, we devise a smart splitting method to transform the database. In the mining phase, a run-time estimation method is proposed to offset the information loss incurred by transaction splitting. By leveraging the downward closure property, we put forward a dynamic reduction method to dynamically reduce the amount of noise added to guarantee privacy during the mining process. Formal privacy analysis and the results of extensive experiments on real datasets show that our PFP-growth algorithm is time-efficient and can achieve both good utility and good privacy.

REFERENCES

[1] C. Dwork, “Differential privacy,” in *ICALP*,2006.

[2] L. Sweeney, “k-anonymity: A model for protecting privacy,”*Int. J. Uncertain. Fuzziness Knowl.-Base Syst*, 2002.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam,“l-diversity: Privacy beyond k-anonymity,” in *ICDE*,2006.

[4] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *VLDB*, 1994.

[5] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in *SIGMOD*, 2000.

[6] Christo Ananth, T.Rashmi Anns, R.K.Shunmuga Priya, K.Mala, “Delay-Aware Data Collection Network Structure For WSN”, International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Special Issue 2 - November 2015, pp.17-21

[7] J. Vaidya and C. Clifton, “Privacy preserving association rule mining in vertically partitioned data,” in *KDD*, 2002.

[8] M. Kantarcioglu and C. Clifton, “Privacy-preserving distributed mining of association rules on horizontally partitioned data,” *TKDE*, 2004.

[9] W. K.Wong, D.W. Cheung, E. Hung, B. Kao, and N. Mamoulis, “Security in outsourcing of association rule mining,” in *VLDB*,2007.

[10] W. K.Wong, D.W. Cheung, E. Hung, B. Kao, and N. Mamoulis,“An audit environment for outsourcing of frequent itemset mining,” in *VLDB*, 2009.

[11] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy preserving mining of association rules,” in *KDD*, 2002.

[12] Maurizio Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, “Anonymity preserving pattern discovery,” *VLDB Journal*,2008.

[13] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, “Discovering frequent patterns in sensitive data,” in *KDD*, 2010.

[14] N. Li, W. Qardaji, D. Su, and J. Cao, “Privbasis: frequent itemset mining with differential privacy,” in *VLDB*, 2012.

[15] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *FOCS*, 2007.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *TCC*, 2006.

[17] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong, “Publishing set-valued data via differential privacy,”in *VLDB*, 2011.

[18] X. Zhang, X. Meng, and R. Chen, “Differentially private setvalued data release against incremental updates,” in *DASFAA*,2013.

[19] L. Bonomi and L. Xiong, “A two-phase algorithm for mining sequential patterns with differential privacy,” in *CIKM*, 2013.

[20] E. Shen and T. Yu, “Mining frequent graph patterns with differential privacy,” in *KDD*, 2013.