International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 6, June 2016

Differentially Private Frequent Item Set Mining

[1] A.Kamatchi, [2] S. Sheik Faritha Begum, [3] A. Rajesh
[1]Post Graduate Scholar, Computer Science and Engineering, C.Abdul Hakeem College of Engineering and Technology,Vellore,
[2] Research Scholar, Bharath University, Chennai,
[3] Professor, C.Abdul Hakeem College of Engineering and Technology,Vellore.

***Abstract* - Data analysis and machine learning have given the ability to improve customer service, update business processes, allocate limited resources more efficiently, and more. At the same time, there are significant (and growing) concerns about individual privacy. The solution is data anonymization but it produces information loss. In search of better privacy and accuracy, we combined the concept of differential privacy with FP –growth algorithm which is known for its effective frequent utility item set mining algorithm and construct a PFP-Growth algorithm. Our proposed work reduce the information loss and computation overhead in the mining process making it better compare with other mining process.**

***Index Terms* — Data Privacy, Differential Privacy, Utility Item sets**

I. INTRODUCTION

An emerging topic in the field of data mining is Utility Mining. The main objective of Utility Mining is to identify the item sets with highest utilities, by considering profit, quantity, cost or other user preferences. Mining High Utility item sets from a transaction database is to find item sets that have utility above a user-specified threshold. Item set Utility Mining is an extension of Frequent Item set mining, which discovers item sets that occur frequently. In many real-life applications, high-utility item sets consist of rare items. Rare item sets provide useful information in different decision-making domains such as business transactions, medical, security, fraudulent transactions and retail communities.

Despite valuable insights the discovery of frequent item sets can potentially provide, if the data is sensitive (e.g., web browsing history and medical records), releasing the discovered frequent item sets might pose considerable threats to individual privacy.

Data anonymization is a type of information sanitization whose intent is privacy protection. It is the process of either encrypting or removing personally identifiable information from data sets, so that the people whom the data describe remain anonymous. Data anonymization enables the transfer of information across a boundary, such as between two departments within an agency or between two agencies, while reducing the risk of unintended disclosure, and in certain environments in a manner that enables evaluation and analytics post-anonymization. In the context of medical data, anonymized data refers to data from which the patient cannot be identified by the recipient of the information. The name, address, and full post code must be removed together with any other information which, in conjunction with other data held by or disclosed to the recipient, could identify the patient. Fig. 1 explain how K-Anonymity is applied over patient record for handling data privacy, first tabular column hold unpreserved patient record and Fig.2 holds the preserved data which doesn't reveals the users identification

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

Inpatient Microdata

Fig.1 Un-Preserved Microdata

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. Apart from privacy issue the main focus in this paper is on mining item sets. A variety of algorithms have been proposed for mining frequent item sets. The Apriori and FP-growth are the two most prominent ones.

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

anonymous Inpatient Microdata

Fig.2 Privacy Preserved Micro data

II. RELATED WORKS

Mining Association rules is one of the research problems in data mining. Given a set of transactions where each transaction is a set of items, an association rule is an expression of the form X => Y, where X and Y are sets of items.

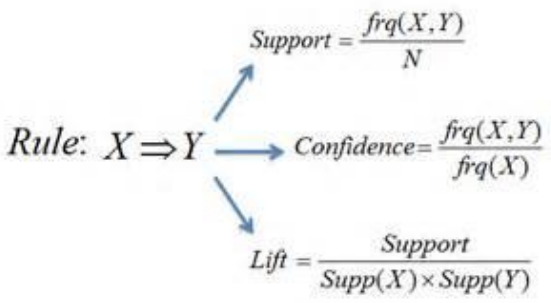

Fig.3 Association Rule Mining

Association rule mining (ARM) is a popular technique for finding co-occurrences, correlations, and frequent patterns, associations among items in a set of transactions or a database. The basic Bottleneck of association rule mining is Rare Item Problem. Most approaches to mining association rules implicitly consider the utilities of the item sets to be equal. The utilities of item sets may differ.

The traditional ARM approaches consider the utility of the items by its presence in the transaction set. The frequency of item set is not sufficient to reflect the actual utility of an item set. For example, the sales manager may not be interested in frequent item sets that do not generate significant profit.

Recently, one of the most challenging data mining tasks is the mining of high utility item sets efficiently. Identification of the item sets with high utilities is called as Utility Mining. The utility can be measured in terms of cost, profit or other expressions of user preferences. For example, a computer system may be more profitable than a telephone in terms of profit. R. Agrawal et al in introduced the concept of frequent item set mining. Frequent item sets are the item sets that occur frequently in the transaction data set. The goal of Frequent Item set Mining is to identify all the frequent item sets in a transaction dataset.

A. Apriori Algorithm

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis. Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The FP-Growth Algorithm, proposed by Han, is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree). In his study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g. the Apriori Algorithm and the Tree Projection. In some later works it was proved that FP-Growth has better performance than other methods, including Eclat and Relim. The popularity and

efficiency of FP-Growth Algorithm contributes with many studies that propose variations to improve his performance. The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance. For so much it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

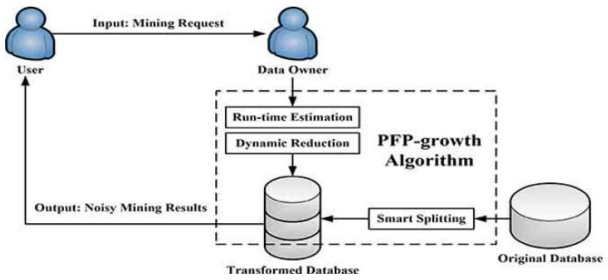

Fig.4 Architecture Diagram

III. PROPOSED WORK

Our proposed system uses the FP-Growth a partitioning-based, depth-first search algorithm for mining frequent utility item sets. FP-growth depends on two data structures, namely header table and FP-tree. For the header table, it is used to store items and their supports. For the FP-tree, each branch represents an item set and each node has a counter. In the header table, each item also contains the head of a list which links all the same items in the FP-tree. In addition we present our private FP -Growth (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 also propose three key methods to address the challenges in designing a differentially private FIM algorithm based on the FP-growth algorithm. 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 item sets 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. In the rest of this section, we discuss the details of the methods.

A. FP-Growth

The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance. For so much it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information which is shown in fig.5. The FP-Growth reduces the search costs looking for short patterns recursively and then concatenating them in the long frequent patterns, offering good selectivity.

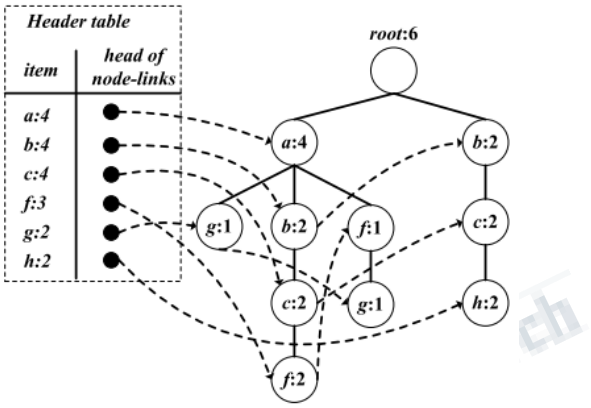

Fig.5 Header Table and FP-Tree

The frequent-pattern tree (FP-tree) is a compact structure that stores quantitative information about frequent patterns in a database. Han defines the FP-tree as the tree structure defined follows. Initially one root labeled as "null" with a set of item-prefix subtrees as children, and a frequent-item-header table (presented in the left side of Fig. 5), then each node in the item-prefix subtree consists of three fields: first Item-name: registers which item is represented by the node; second is Count: the number of transactions represented by the portion of the path reaching the node; third is Node-link: links to the next node in the FP-tree carrying the same item-name, or null if there is none. Each entry in the frequent-item-header table consists of two fields: first is Item-name: as the same to the node and second is Head of node-link: a pointer to the first node in the FP-tree carrying the item-name. Additionally the frequent-item-header table can have the count support for an item. the Fig. 4 show an example of a FP-tree. The simple working process is show in the Fig. 6 below.

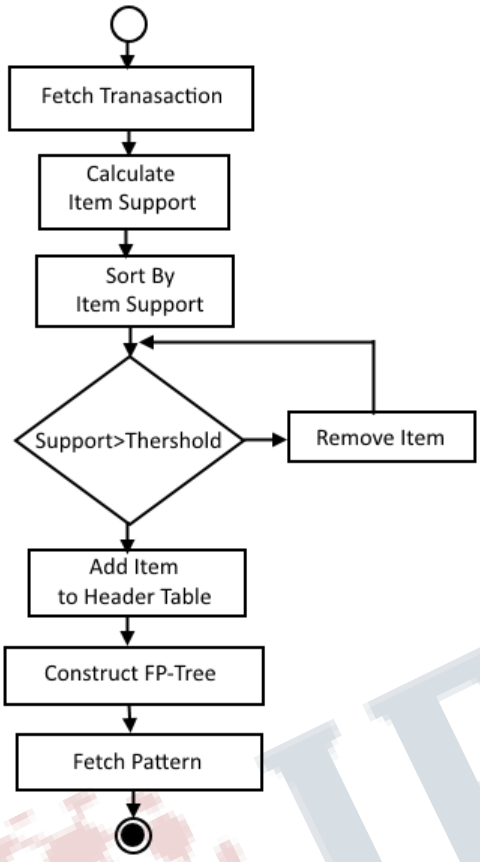

Fig.6 FP-Growth

B. 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 each of which meets the maximal length constraint. 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 item set when adding (removing) this subset into (from) the database. It can be considered as a multiplier. Due to the weighted splitting operation, each subset of the divided transactions only preserves incomplete frequency information. To offset such information loss, we propose a run-time estimation method

C. Run-Time Estimation

Despite the potential advantages, transaction splitting might cause information loss. To offset the information loss caused by transaction splitting we use run-time estimation. The method consists of two steps: based on the noisy support of an item set 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 each item set, we estimate its "average" support to determine whether it is frequent. We also estimate its "maximal" support to decide whether to use it to generate candidate frequent item sets.

D. Dynamic Reduction

For FP-growth, it is a depth-first search algorithm. We cannot obtain the exact number of support computations in Qi until the mining process is finished. A potential approach is to modify FP-growth to enforce frequent item sets of the same length to be generated simultaneously, such that we can use frequent (i-1)-item sets to obtain the exact number of support computations in Qi. This approach, however, will lead to exponential memory use. Instead, we propose our lightweight dynamic reduction method. 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 item set are infrequent), and dynamically reduce the sensitivity of support computations by decreasing the upper bound on the number of support computations.

A simple data structure up-array is used to register the upper bounds on the number of support computations. We initialize up-array to the number of all possible i-item sets. we construct the conditional pattern base of item set Based on the header table, we can see the set of items which might appear in the conditional pattern base. For each item in set, we first compute its noisy support in conditional pattern base. Then, based on the obtained noisy support, by using our run-time estimation method, we estimate the "maximal" support of item set. If the estimated "maximal" support is smaller than the threshold, we regard item as infrequent items in conditional pattern base. Next, we decrease the upper bounds based on the infrequent items found in conditional pattern base.

IV. CONCLUSION

In this paper we propose our private FP-growth algorithm, for mining frequent utility item sets in privacy preserved manner. Our process 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. Moreover, 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.

REFERENCE

[1] L. Sweeney, "k-anonymity: A model for protecting privacy,"Int. J. Uncertainity Fuzziness Knowl.-Base Syst., vol. 10, no. 5, pp. 557–570, 2002.

[2] C. Zeng, J. F. Naughton, and J.-Y. Cai, "On differentially private frequent item set mining," Proc. VLDB Endowment, vol. 6, no. 1, pp. 25–36, 2012.

[3] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 639–644, 2002.

[4] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," IEEE Trans. Knowl. Data Eng., vol. 16, no. 9, pp. 1026–1037, Sep. 2004.

[5] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 503–512, 2010.

[6] N. Li, W. Qardaji, D. Su, and J. Cao, "Privbasis: Frequent item set mining with differential privacy,"Proc. VLDB Endowment, vol. 5, no. 11, pp. 1340–1351, 2012.

[7] F. McSherry and K. Talwar, "Mechanism design via differential privacy," inProc. 48th Annu. IEEE Symp. Found. Comput. Sci., pp. 94–103, 2007.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," inProc. 3rd Conf. Theory Cryptography, pp. 265–284, 2006.

[9] X. Zhang, X. Meng, and R. Chen, "Differentially private setvalued data release against incremental updates," inProc. 18th Int. Conf Database Syst. Adv. Appl., pp. 392–406, 2013.