International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Privacy Preserving Closed Frequent Pattern Mining

Anju Vijayan

Computer Science and Engineering, ICET, Mahatma Gandhi University, Muvattupuzha, Kerala, India

Abstract: ***Mining closed frequent item sets is one of the important problems in data mining. There exists a possibility of designing differentially private Frequent Itemset Mining (FIM) algorithm which can achieve high data utility, efficiency and high degree of privacy. Private Frequent Pattern mining algorithms have a preprocessing phase and mining phase. In the preprocessing phase a novel smart splitting algorithm is used for transforming the database. In the mining phase transaction splitting is done. Certain amount of noise is added to the output for enhancing privacy. The amount of noise added is considerably reduced.***

Keywords: Frequent Itemset Mining, transaction splitting

1. Introduction

Finding frequent itemsets is the most costly task in data mining. However, frequent sequential pattern mining is a central task in many fields. Also, release of these patterns is raising increasing concerns on individual privacy. A solution to this is provided by differential privacy. Differential privacy framework provides formal and provable guarantees of privacy. The differential privacy mechanism encrypts the frequency results with noise. In this work, a novel two-phase algorithm having preprocessing and mining phase is used.

A frequent itemset mining algorithm takes as input a dataset of transactions by a group of individuals, and produces frequent itemsets as the output. This immediately creates a privacy concern. No one can be confident that publishing the frequent itemsets in the dataset does not reveal private information about the individuals whose data is being studied. This problem concerns with some other fact that it is really difficult to know what data the individuals would like to protect. A possible answer to that challenge is presented by differential privacy, which guarantees that the presence of an individual's data in a dataset does not reveal much about that individual. The possibility of developing differentially private frequent itemsets mining algorithms is explored with the goal to guarantee differential privacy while still finding useful frequent itemsets.

Many organizations are publishing data of individuals that contain unaggregated information about individuals such as medical, voter registration, census, and customer data. This kind of data is a valuable source of information for the allocation of public funds, medical research, and trend analysis. If individuals can be uniquely identified in this kind of data then their private information would be disclosed, and this is unacceptable [1]. With the increasing ability to collect personal data, privacy has become a major concern.

Discovering frequent patterns from data is a popular exploratory technique in data mining. However, if the data are sensitive releasing information about significant patterns or trends carries significant risk to privacy. The FIM algorithm provides a secure way to 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.

Frequent item set finds item set that occur in transaction more frequently than a particular threshold set by user. Differential privacy offer strong privacy of released data. This task is very challenging due to the possibility of long transactions. A solution is to limit the length of transactions by truncating long transactions. This approach might cause too much information loss and result in inefficiency. To limit the length of transactions and to reduce the information loss, long transactions should be split rather than truncated. For that a transaction splitting based differentially private FIM algorithm is proposed. The splitting technique divides long transactions into sub-transactions whose length is within a particular limit.

In addition, a support estimation technique is used to estimate the actual support of itemsets in the original database. This can help to reduce the information loss caused by transaction splitting. It constructs a FP-tree, which is usually smaller than the original database, and thus saves the costly database scans in the subsequent mining processes. It is possible to promote the utility of differentially private frequent itemset mining algorithm by limiting the length of transactions or by splitting the transaction.

2. Related Work

Identifying patterns and trends from large quantities of data is one of the main challenges of data mining. There are various algorithms such as clustering, classification, association rule mining and sequence detection, which are developed within a centralized model, with all data being gathered into a central site, and algorithms being run against that data[3]. To mine association rules, the data is vertically partitioned, means each site contains some elements of a transaction.

While extracting important knowledge from large data collections, some of these collections are split among various parties. Directly sharing the data is not secure due to privacy concerns. To mine association rules over horizontally partitioned data, some cryptographic techniques are used to minimize the information shared, while adding little overhead to the mining task. Privacy concerns can prevent building a centralized warehouse[4][5]. Data may be distributed among several entities, none of which are allowed to transfer their data to another site.

Another way of providing privacy is to substitute cipher techniques in the encryption of transactional data for outsourcing association rule mining. Rather than the one-to-one item mapping substitution cipher, a more secure encryption scheme based on a one-to-n item mapping that transforms transactions non-deterministically is used. An effective and efficient encryption algorithm based on this method which performs a single pass over the database and is suitable for applications in which data owners send streams of transactions to the service provider This technique is highly secure with a low data transformation cost.

Anonymity of individual data of the source code is preserved during each stage of mining. In data mining, models and patterns represent a large number of individuals and thus there is a greater chance of revealing individual identities: this is the case of the minimum support threshold in frequent pattern mining. But really this belief is ill-founded. Firstly the concept of k-anonymity is shifted from source data to extracted patterns. The notion of a threat to anonymity in the context of pattern discovery is characterized and a methodology to efficiently identify all such possible threats that arise from the disclosure of the set of extracted patterns. Thus privacy protection is enabled that allow the disclosure of extracted knowledge while protecting the privacy of individuals in the source database. In some cases the threats to anonymity cannot be avoided. In such cases those threats are eliminated by means of pattern distortion performed in a controlled way.

Among the large number of algorithms have been proposed for mining frequent itemsets, the Apriori and FP-growth are the important algorithms. The difference between Apriori and FP-growth is that, Apriori is a breadth first search, candidate set generation-and-test algorithm, while FP - growth is a depth-first search algorithm, which requires no candidate generation[6][7][9]. Apriori requires the number of database scans which is equal to the maximal length of frequent itemsets while FP- growth performs only two database scans. Also FP-growth works faster than Apriori. A differentially private FIM based FP-growth algorithm provides appealing features. This algorithm also provides high data utility, high degree of privacy and high time efficiency.

Differentially private Frequent Itemset Mining is actually an enhancement of the FP-growth algorithm. The features of FP-growth algorithm are used to increase the efficiency of FIM. This results in high data utility and high degree of privacy. Existing studies show that there is no such algorithm that satisfies all these requirements simultaneously. Compared to the other itemset mining techniques, which adopt Apriori-like candidate set generation-and-test approach, using FP-growth is really economical. Candidate generation is costly, especially when the pattern is long and complicated[2]. FIM algorithm based on FP-growth outperforms Apriori in terms of time efficiency, data utility, privacy and cost.

3. Methodologies

3.1 Frequent Itemset Mining

Discovering frequent patterns from data is a popular and exploratory technique in data mining. If the data are sensitive particularly when it deals with 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 guarantees of privacy for the individuals whose information is stored in the database under consideration.

To find the most frequent patterns in a data set of sensitive records, FP-growth algorithm is seemed to be the best. In order to provide differential privacy the output is made uncertain. A degree of uncertainty is provided in the output to preserve privacy in the presence of arbitrary external information. To achieve this, a noisy list of patterns that are close to the most frequent patterns in the given data, are added with the actual output. The FIM algorithm gives importance to frequent pattern mining as well as privacy. 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.

3.2 FP-growth Algorithm

The FP-Growth Algorithm 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 this method an FP tree construction is used. FP-Growth has better performance than other methods of finding frequent patterns. The popularity and efficiency of FP-Growth Algorithm contributes with many studies that propose variations to improve the performance[8]. The frequent- pattern tree (FP-tree) is a compact structure that stores quantitative information about frequent patterns in a database. The FP-Growth Algorithm is an alternative way to find frequent itemsets without using candidate generations, thus improving performance. For that 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 itemset association information.

In simple words, FP growth algorithm works as follows: first it compresses the input database creating an FP-tree instance to represent frequent items. After the first step, it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately. Using this strategy, the FP-growth reduces the search costs looking for short patterns recursively and then concatenating them in the long frequent patterns, offering good selectivity. In large databases, it's not possible to hold the FP-tree in the main memory. A strategy to cope with this problem is to firstly partition the database into a set of smaller databases, called projected databases,

and then construct an FP-tree from each of these smaller databases.

For generating conditional pattern bases, FPgrowth forms two data structures, namely, header table and FP-tree. For the header table, item and their support are stored. For the FP-tree, each branch represents an itemset and each node is having a counter. In the header table, each item contains the head of a list which is linked to same items in the FP-tree.

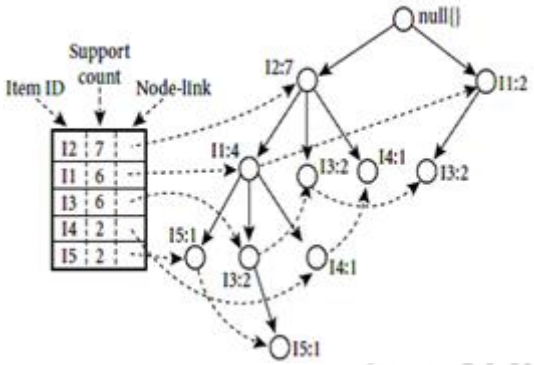

Figure: FPtree with database

The database is splitted into transactions of limited size. Each transaction represents an individual's record. Figure shows a simple transaction database and the corresponding FP tree. Each non empty set is it's transaction. The length of an itemset is the number of items in it. Support of itemset is the number of elements. To limit the length of transaction without information loss a novel smart splitting method is introduced. Also the database is spitted, not truncated which increases the efficiency of the entire process.

The FP-tree is short, and is constructed, to mine frequent patterns. Only frequent length-1 items will have nodes in the tree, and the tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than less frequently occurring nodes. Second, an FP-tree-based pattern fragment growth mining method, is developed, which starts from a frequent length-1 pattern, examines only its conditional pattern base, constructs its conditional FP-tree, and performs mining re cursively with such a tree.

The pattern growth is achieved via concatenation of the suffix pattern with the new ones generated from a conditional FP-tree. Since the frequent itemset in any transaction is always encoded in the corresponding path of the frequent pattern trees, pattern growth ensures the completeness of the result. In this context, the method is not Apriori- like restricted generation-and-test but restricted test only. The major operations of mining are count ac-cumulation and preffix path count adjustment, which are usually much less costly than candidate generation and pattern matching operations performed in most Apriori-like algorithms.

Third, the search technique employed in mining is a partitioning-based, divide-and-conquer method rather than Apriori-like bottom-up generation of frequent itemsets combinations. This dramatically reduces the size of conditional pattern base generated at the subsequent level of search as well as the size of its corresponding conditional FP-tree. Moreover, it transforms the problem of finding long frequent patterns to looking for shorter ones and then concatenating the suffix. It employs the least frequent items as suffix, which offers good selectivity. All these techniques contribute to substantial reduction of search.

4. Conclusion

In this paper, the problem of designing differentially private frequent itemset and closed frequent itemset is done. This provides privacy as well as efficiency. Frequent itemset mining incurs huge risk when the database requires privacy. The database is done with two phases. In the preprocessing phase, the database is split into transactions of limited size by using the smart splitting method. A runtime estimation method is done to eliminate information loss during splitting. The transformed database is then gone through mining phase. In the mining phase, an FP-tree is constructed to identify the frequent patterns. The patterns with its frequency are identified and from this closed frequent patterns are also identified. To enhance privacy all the output are appended with noise. This algorithm provides data utility, privacy and time efficiency.

5. Acknowledgement

The author would like to thank Meenu Varghese, Assistant Professor, Department of Information technology, Ilahia College of Engineering and Technology,Muvattupuzha for her moral and technical support.

References

[1] N. Li, W. Qardaji, D. Su, and J. Cao, "Privbasis: frequent itemset mining with differential privacy," in *VLDB*, 2012.
[2] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong, "Publishing set-valued data via differential privacy," in *VLDB*, 2011.
[3] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong, "Publishing set-valued data via differential privacy," in *VLDB*, 2011.
[4] 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.
[5] Maurizio Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "Anonymity preserving pattern discovery," *VLDB Journal*, 2008.
[6] Maurizio Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "Anonymity preserving pattern discovery," *VLDB Journal*, 2008.
[7] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *TKDE*, 2004
[8] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Base Syst*, 2002.

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

Author Profile

Anju Vijayan received the Bachelor of Technology degree in Computer Science and Engineering from Mahatma Gandhi University, Kerala. She is currently doing Master of technology degree in Computer Science and Engineering with Specialization in Information Systems from Mahatma Gandhi University, Kerala.