ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Frequent Itemset Mining Using PFP-Growth via Transaction Splitting

Anusuya M[1],Sudharani K[2],Ganthimathi M**[3],**Sumathi G [4]

PG Student, Dept. of C.S.E, Muthayammal Engineering College, Rasipuram, Tamilnadu, India[1]

PG Student, Dept. of C.S.E Muthayammal Engineering College, Rasipuram, Tamilnadu, India[2]

Associate Professor, Dept. of C.S.E, Muthayammal Engineering College, Rasipuram, Tamilnadu, India[3]

Associate Professor, Dept. of C.S.E, Muthayammal Engineering College, Rasipuram, Tamilnadu, India[4]

ABSTRACT: Frequent itemset mining (FIM) is one of the part in data mining. It has practical importance in a wide range of application areas such as decision support, Web usage mining, bioinformatics, etc. Frequent Itemset not only achieve high time efficiency and a high degree of privacy, but also offer high data utility. Aprioriand FP-growth are the two most important algorithm, to find Frequent itemset. 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*.It have two steps: Find all itemsets that have minimum support frequent itemsets(candidate list) and generate frequent itemsets list . FP-growth is a depth-first search algorithm, which requires no candidate generation. Compared with Apriori, FP-growth less time consuming algorithm, but it enforces the limit by truncating transactions, if a transaction has more items than the limit, deleting items until its length is under the limit. Thus, the PFP (private FP-growth) approach was proposed to find frequent itemset for all data items in the database without truncating.

KEYWORDS: Frequent itemset mining, Transaction splitting, PFP-growth algorithm,Run- time estimation, Dynamic reduction.

I. INTRODUCTION

Data mining also called as data or knowledge discovery, it is the process of evaluating data from different perspectives and summarizing it into useful information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for evaluating data and summarizes the relationships identified. To show the figure1.1Technically, data mining allows users to analyze data from many different dimensions or angles, categorize process of finding correlations or patterns among dozens of fields in large relational databases.Although data mining is a relatively new term, the technology is not. Companies have used powerful computers to sift through volumes of supermarket scanner data and evaluate market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost. Data mining is the practice of automatically probing large stores of data to discover patterns and trends that go beyond simple analysis. Data mining uses cultured mathematical algorithms to section the data and evaluate the probability of future events. Data mining is also known as Knowledge Discovery in Data (KDD) refer to Figure 1.2.

1.1 Association Rule

Association rule is an important data mining model studied extensively by the database and data mining community. Assume all data are categorical. Association rule is the method for discovering interesting relations between variables in large database. This is intended to identify strong rules discovered in databases using different measures of interestingness. Introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems.

Association rules are created by analyzing data for frequent then patterns and using the criteria support, confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

DOI: 10.15680/IJIRCCE.2016. 0402147

In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, catalog design and store layout. Programmers use association rules to build programs capable of machine learning. Machine learning is a type of artificial intelligence (AI) that seeks to build programs with the ability to become more efficient without being explicitly programmed.Items that occur often together can be associated to each other.These together occuring items form a frequent itemset. Conclusions based on the frequent itemsets form association rules.
For ex. {milk, cocoa powder} can bring a rule *cocoa powder* ➔ *milk*

1.2 FrequentItemset Mining (FIM)

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. A variety of algorithms have been proposed for mining frequent itemsets. Apriori and FP-growth are the two most important algorithm. To find Frequent itemset. Apriori is a breadth first search, candidate set generation-and-test algorithm.FP-growth is a depth-first search algorithm, which requires no candidate generation. Compared with Apriori, FP-growth less time consuming algorithm, but 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). Thus, the PFP (private FP-growth) approach was proposed.

II.LITRATURE REVIEW

Mining frequent patterns in transaction databases, time-series databases, and many other kinds of databases has been studied popularly in data mining research. Most of the previous studies adopt an *Apriori*-like candidate set generation-and-test approach. However, candidate set generation is still costly, especially when there exist a large number of patterns and/or long patterns. In this study, propose a novel frequent-pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method, *FP-growth*, for mining *the complete set of frequent patterns* by pattern fragment growth. Efficiency of mining is achieved with three techniques: (1) a large database is compressed into a condensed, smaller data structure, FP-tree which avoids costly, repeated database scans, (2) our FP-tree-based mining adopts a pattern-fragment growth method to avoid the costly generation of a large number of candidate sets, and (3) a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined patterns in conditional databases, which dramatically reduces the search space. Our performance study shows that the *FP-growth* method is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the *Apriori*algorithm and also faster than some recently reported new frequent-pattern mining methods.**[6]**Frequent sequential pattern mining is a central task in many fields such as biology and finance. However, release of these patterns is raising increasing concerns on individual privacy. In this paper, study the sequential pattern mining problem under the differential privacy framework which provides formal and provable guarantees of privacy. Due to the nature of the differential privacy mechanism which perturbs the frequency results with noise, and the high dimensionality of the pattern space, this mining problem is particularly challenging. In this work, propose a novel two-phase algorithm for mining both prefixes and substring patterns. In the first phase, our approach takes advantage of the statistical properties of the data to construct a model-based prefix tree which is used to mine prefixes and a candidate set of substring patterns. The frequency of the substring patterns is further refined in the successive phase where employ a novel transformation of the original data to reduce the perturbation noise. Extensive experiment results using real datasets showed that our approach is effective for mining both substring and prefix patterns in comparison to the state-of-the art solutions.[1] The possible number of patterns grows exponentially with the length of the patterns, which makes the mining process inefficient if done naively. The count of occurrences of patterns has high sensitivity, which means that a large amount of perturbation noise is needed to guarantee differential privacy. The sequential pattern mining problem under the differential privacy framework which provides formal and provable guarantees of privacy. Due to the nature of the differential privacy mechanism which perturbs the frequency results with noise, and the high dimensionality of the pattern space, this mining problem is particularly challenging.**[2]**Here present a framework for mining association rules from transactions consisting of categorical items where the data has been randomized to preserve privacy of individual transactions. While it is feasible to recover association rules and preserve privacy using a straightforward “uniform" randomization, the discovered rules can unfortunately be exploited to and privacy breaches. analyze the nature of privacybreaches and propose a class of randomization operators that are much more effective than uniform randomization in limiting the breaches. Derive formulae for an unbiased support estimator and its variance, which allow us to recover itemset supports from randomized datasets, and show how to incorporate these formulae into mining algorithms. Finally, present experimental results that validate the algorithm by applying it on real datasets. Data mining can extract important knowledge from large data collections – but sometimes these collections are split among various parties. Privacy concerns may prevent the parties from directly sharing the data, and some types of information about the data. This paper addresses secure mining of association rules over horizontally partitioned data. The methods incorporate cryptographic techniques to minimize the information shared, while adding little overhead to the mining task.**[3].** Outsourcing association rule mining to an outside service provider brings several important benefits to the data owner. These include (i) relief from the high mining cost, (ii) minimization of demands in resources, and (iii) effective centralized mining for multiple distributed owners. On the other hand, security is an issue; the service provider should be prevented from accessing the actual data since (i) the data may be associated with private information, (ii) the frequency analysis is meant to be used solely by the owner. This paper proposes substitution cipher techniques in the encryption of transactional data for outsourcing association rule mining. Our algorithm performs a single pass over the database and thus is suitable for applications in which data owners send streams of transactions to the service provider. A comprehensive cryptanalysis study is carried out. The results shows that our technique is highly secure with a low data transformation cost.[4].

III.EXISTING SYSTEM

Apriori

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*.Support counting is expensive, Multiple database scans (I/O), breath first search algorithm to be used.Probably the best known algorithm. It have two steps: Find all itemsets that have minimum support frequent itemsets. Use frequent itemsets to generate rules. The Apriori property follows a two step process:

- Join step: Ck is generated by joining Lk-1 with itself
- Prune step: Any(k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

The Apriori Algorithm

Ck: Candidate itemset of size k
Lk: frequent itemset of size k

```
L1= {frequent items};
for(k= 1; Lk!=∅; k++) do begin
Ck+1= candidates generated from Lk;
for each transaction tin database do
increment the count of all candidates in Ck+1that are contained in t
Lk+1= candidates in Ck+1with min_support
end
return∪kLk;
```

Limitation

- Needs several iterations of the data.Uses a uniform minimum support threshold.Difficulties to find rarely occuring events
- Alternative methods (other than appriori) can address this by using a non-uniform minimum support thresold,Some competing alternative approaches focus on partition and sampling

Apriori is Candidate generation generates large numbers of subsets ,the algorithm attempts to load up the candidate set with as many as possible before each scan. Bottom-up subset exploration essentially a breadth-first traversal of the subset lattice finds any maximal subset S only after all of its proper subset.

FP-growth

FP-growth is a depth-first search algorithm, which requires no candidate generation. FP-growth only performs two database scans, which makes FP-growth an order of magnitude faster than Apriori. 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, FP growth is a depth-first search algorithm. It is hard to obtain the exact number of support computations of *i*-itemsets during the mining process. Divide and conquer method to use extract prefix path sub-trees ending in an item.

Two data structures, namely *header table* and *FP-tree*: FP-tree : Branch represents an itemset and each node has a counter ,Header table: store items and their supports.

- FP-Tree Construction.
- Extracts frequent itemsets directly from the FP-Tree.

Limitations

- It is hard to obtain the exact number of support computations of *i*-itemsets during the mining process.
- 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).

IV.PROPOSED SYSTEM

Address these challenges, present our Private FP growth (PFP-growth) algorithm, which consists of a preprocessing phase and a mining phase. In the preprocessing phase, 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, 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, privately discover frequent itemsets. During the mining process, dynamically estimate the number of support computations, so that can gradually reduce the amount of noise required by differential privacy.In the mining phase, to offset the information loss caused by transaction splitting, devise a run-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 Dynamic reduction method to dynamically reduce the amount of noise added to guarantee privacy during the mining process. Explore the possibility of designing a differentially private FIM algorithm which can not only achieve high data utility and a high degree of privacy, but also offer high time efficiency.

Advantages

- PFP-growth algorithm is time-efficient and can achieve both good utility and good privacy.
- The preprocessing phase does not consume too much time.
- The performance is significantly improved by adopting our transaction Splitting techniques

V.SYSTEM ARCHITECTURE

Through formal privacy analysis, show that our PFP growth algorithm is ϵ-differentially private. Extensive experimental results on real datasets show that our algorithm outperforms existing differentially private FIM algorithms. To use the folling figure 2.4. Moreover, to demonstrate the generality of our transaction splitting techniques and further enrich the application spectrum, apply our transaction splitting techniques, including the smart splitting and run-time estimation methods, to Apriori by modifying the algorithm. Preliminary experimental results show that the performance of the Apriori-based algorithm is significantly improved by adopting our transaction splitting techniques.

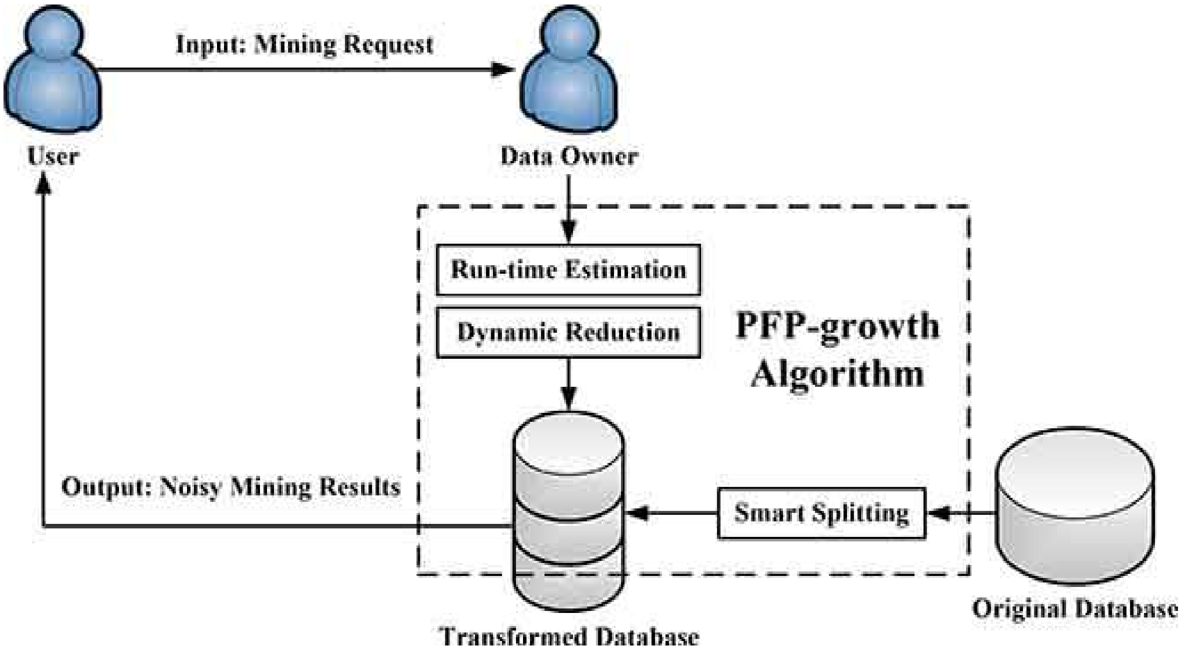
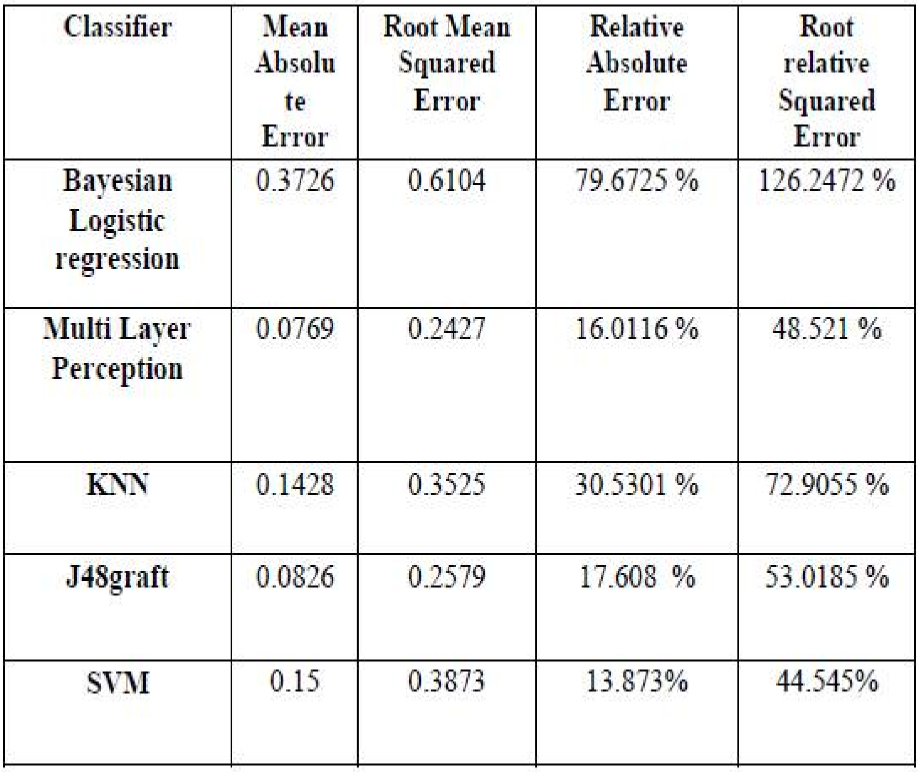


Fig 2.4 System Architecture

To summarize, our key contributions are: 1). Revisit the tradeoff between utility and privacy in designing a differentially private FIM algorithm. demonstrate that the tradeoff can be improved by our novel transaction splitting techniques. Such techniques are not only suitable for FP-growth, but also can be utilized to design other differentially private FIM algorithms. 2). Develop a time-efficient differentially private FIM algorithm based on the FP-growth algorithm, which is referred to as PFP-growth. In particular, by leveraging the downward closure property, a dynamic reduction method is proposed to dynamically reduce the amount of noise added to guarantee privacy during the mining process. 3). Throughformal privacy analysis, show that our PFP-growth algorithm is ϵ-differentially private. Extensive experiments on real datasets illustrate our algorithm substantially outperforms the state-of-the-art techniques.

VI.RESULTS

The sequential pattern mining problem under the differential privacy framework which provides formal and provable guarantees of privacy. Due to the nature of the differential privacy mechanism which perturbs the frequency results with noise, and the high dimensionality of the pattern space, this mining problem is particularly challenging. Dynamic reduction method to dynamically reduce the amount of noise added to guarantee privacy during the mining process.



Classifier	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root relative Squared Error
Bayesian Logistic regression	0.3726	0.6104	79.6725 %	126.2472 %
Multi Layer Perception	0.0769	0.2427	16.0116 %	48.521 %
KNN	0.1428	0.3525	30.5301 %	72.9055 %
J48graft	0.0826	0.2579	17.608 %	53.0185 %
SVM	0.15	0.3873	13.873%	44.545%

Explore the possibility of designing a differentially private FIM algorithm which can not only achieve high data utility and a high degree of privacy, but also offer high time efficiency.System design is the process of planning a new system to complement or altogether replace the old system. The purpose of the design phase is the first step in moving from the problem domain to the solution domain. The design of the system is the critical aspect that affects the quality of the software. System design is also called top-level design. The design phase translates the logical aspects of the system into physical aspects of the system.

VII.CONCLUSIONS AND FUTURE WORK

In this project, investigate the problem of designing a differentially private FIM algorithm. All product details are maintained by admin and these details stored in database using insert, update and delete operation. Customer enters our site using username and password. After login admin generate random password and send to customer mode for verification. If customer gave the wrong random password means ignore customer login. This is an extra security for login. 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, devise a smart splitting method to transform the database.

REFERENCES

[1] Sen Su, ShengzhiXu, Xiang Cheng, Zhengyi Li, and FangchunYang,“Differentially Private Frequent Itemset Mining via Transaction Splitting,” DOI 10.1109/TKDE.2015.2399310, IEEE Transactions on Knowledge and Data Engineering.

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

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

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

[5] C. Zeng, J. F. Naughton, and J.-Y. Cai, “On differentially private frequent itemset mining,” in *VLDB*, 2012.

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

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

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

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

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