Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 489-495, 2016
ISSN 1990-9233;

DOI: 10.5829/idosi.mejsr.2016.24.TAET23629

An Efficient Private Frequent Itemset and Association Rule Mining

A. Jayakumari, R. Rohini and B. Anitha

PG Scholar, Department of CSE,
Vivekanandha College of Engineering for Women, Tamilnadu, India

Abstract: Privacy thought has a lot of significance within the application of information mining. It's important that the privacy of individual parties won't be exposed once data processing techniques square measure applied to an outsized assortment of information concerning the parties. This paper addresses the problem of privacy-preserving frequent pattern and association rule mining in such a cache schema across ASC to DB. In this paper we present an efficient private FP-Growth and canopy rule approach for mining private frequent itemsets and private rule generation in cache dataset. Our experimental results obtained using both synthetic and real data sets.

Key words: Cache Data · FP-Growth · Cover rule · Private frequent itemset · Private rule generation

INTRODUCTION

Frequent Itemset Mining (FIM) is individual in the majority critical problems in data mining. It has practical importance in a broad collection of function areas such as result maintain, Web usage mining, bioinformatics, etc. Given a database, everywhere all operation contains a set of items, FIM tries to find itemsets that occur in transactions more frequently than a given entrance. Inspite of expensive insight the innovation of frequent itemsets can potentially provide, if the data is susceptible (e.g., web browsing olden times and medicinal records), releasing the discovered frequent itemsets might pose considerable intimidation to personality confidentiality. Differential privacy has been proposed as a way to deal with such difficulty. Nothing like the anonymization-base confidentiality model (e.g., k-anonymity and l-diversity), degree of difference isolation offer physically powerful imaginary assurance on the privacy of free data with no making assumption about an attacker's background knowledge. In exacting by adding up a watchfully chosen quantity of noise, differential privacy assure that the production of a calculation is insensitive to changes in any individual's proof and thus restrict privacy leak through the outcome. A mixture of algorithms have been proposed for mining frequent itemsets.

Vertical partitioning often results in a star schema, where a site (the center of the star) will contain the joining information as a relation for the relations in the remaining sites. The center table in this site is also called the fact table and the residual tables are called the dimension tables.

The number of association rules generated by data mining algorithms can easily overcome a human analyst. To address this problem several methods were proposed. Our paper continues the line of research from by introducing a new rule of private for association rules and by defining the idea of a cover of the relationship rules as a minimum set of rules that are non-redundant with respect to this new private rule.

Let us consider a real-life motivating example where performance, accuracy and privacy are important issue. Credit card dealings are winning a large share of the payment systems worldwide and have led to a higher price of stolen account numbers and consequent wounded by banks. Each bank has a database which stores both legitimate and fake credit card dealings, so let us think distributed data mining for credit card fraud detection. The purpose is to construct a data mining replica which will be used by automated fraud detector systems to prevent fraudulent dealings. This data mining model must be very universal and exact, because a

Corresponding Author: A. Jayakumari, PG Scholar, Department of CSE,
Vivekanandha College of Engineering for Women, Tamilnadu, India.

mistake means great loss of money. In sort to construct accurate models of credit card scheme, a data mining system must have access to information about the fake pattern of all banks (i.e., some types of fraud can have occurred only in one bank). The problem is that banks are constrained by law (and also by competitive reasons) from sharing data about their customers with other banks. However, they may split "black-box" model; that is, they can share knowledge about fraudulent transactions, but not their data. Moreover, there are millions of credit card dealings process each day and the data mining system must be scalable and able to calculate the model of credit card fraud in a timely approach.

Related Work: A frequent method for mining circulated databases is the centralized one, where all the data is moved to a single innermost spot and then extracted. Another general method is the local one, where models are built locally in each location and next motivated to a regular position anywhere they are combined. The later approach is the quickest but often the most perfect, while the past method is more perfect but generally quite expensive in terms of time required. In the seek for exact and professional solution, a number of intermediate approaches have been proposed [1, 4, 2]. In [1] three dispersed mining approach were projected. The Count Distribution algorithm is a simple parallel implementation of APRIORI [2]. The entire site generates the complete set of candidates and each site can thus independently get local support counts as of its division. At each iteration the algorithms do a sum reduction operation to obtain the global support counts by exchange local support counts by way of all other sites. Since only the support counts are exchange between the sites, the message transparency is compact

In [1] Authors Mining frequent patterns in transaction databases, time-series databases and various previous kinds of databases has been studied usually in data mining research. Most of the previous studies approve an Apriori like applicant set invention-and-experiment approach. However, candidate set age group is still expensive, particularly when there exist large number of patterns and/or long patterns. In [2] Authors In this paper, we study the chronological pattern mining problem over the degree of difference confidentiality structure which provide proper and verifiable guarantee of privacy.

Owing to the natural history of the discrepancy privacy system which perturb the occurrence results with blast and the lofty dimensionality of the outline space, this mining problem is mainly challenging. In this work, we propose a narrative two-phase algorithm for removal both prefixes and substring patterns. In [3] Author present a framework for withdrawal organization rules from dealings consisting of definite items where the information has been randomized to protect privacy of person dealings. Although it is achievable to improve association rules and protect confidentiality by means of a uncomplicated "regular" randomization, the exposed rules can unfortunately be exploited to and privacy breaches. In [4] Author outsourcing society rule taking out to an outside facility supplier brings numerous significant profit to the data holder. These contain (i) release from the high mining cost, (ii) minimization of stress in income and (iii) efficient central mining for numerous circulated owners. This paper proposes substitution cipher technique in the encryption of transactional information for outsourcing society rule mining. In [5] Authors verdict recurrent itemsets is the most expensive job in association rule mining.

Outsourcing this assignment to a facility supplier brings a number of profit to the information holder such as cost relief and a less obligation to storage space and computational income.

Mining grades, still can be spoiled if the service provider (i) is sincere but make mistakes in the mining procedure, or (ii) is lazy and reduce expensive calculation, recurring imperfect results, or (iii) is nasty and contaminate the mining grades.

Problem Definition: Consider a relational database with a cache schema. There are multiple caches, say A and B, storing their own tables, TA and TB, respectively. There is also a special cache C, which stores the relationship of the other table in a table called the cache table, or simply CT. Sites A and B are called the dimension table while C is called the cache table. The table at every aspect of location is called a dimension table, which has its own primary key and other attributes. The primary keys of transactions of tables TA and TB are denoted by ai and bi, respectively. For simplicity, we assume that such a primary key is also the corresponding transaction id and these transaction ids will be foreign keys in the cache table. Therefore, table F stores the relationship of TA and TB by storing the combinations of foreign keys ai and bi in FT.

The attributes-value pairs of a transaction or a record in the table TA and TB are denoted by xi and yi, respectively. For example, we may have season, summer and occupation, academics, as some

attribute-value pairs. An itemset is a place of attribute-value pairs, into which the attributes should be unique. An itemset appears in a record of a table if all of its attribute-value pairs appear in the record.

Proposed Work: The proposed system makes a small cache of the privacy-preserving FIM algorithms proposed above. A similar property of these algorithms is that, on each step of each iteration, all they compute a frequent itemset of size k, based on the frequent itemsets of size k-1 (the difference of these algorithms is mainly in which of k-sets are computed in parallel). Given the cache Table I = {i1;...; in}, a transaction t is a subset of I and a transaction database D is a multiset of communication. Each transaction represent an entity's record. A simple transaction database. A non-empty set X I is called an itemset. The length of an itemset is the number of items in it. An itemset is called a k-itemset if it contains k items. We say a transaction t contains an itemset X if X is a subset of t. The support of itemset X is the number of transactions containing X in the database. An itemset is frequent if its support is no less than the user-specified minimum support threshold. Given a transaction database and a user-specified minimum support threshold, the goal of PFP-Growth is to find the complete set of frequent itemsets. In the rest of this paper, we use "threshold" since hand over proposed for "client-particular smallest amount sustain threshold".

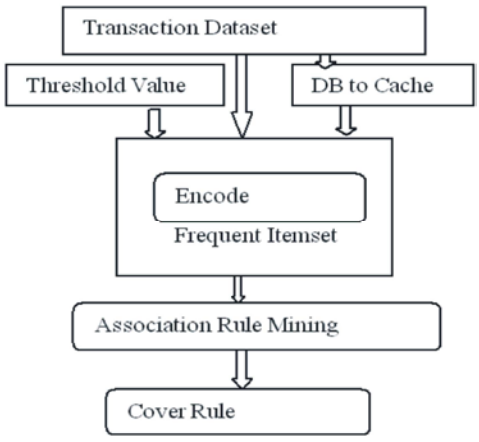

Fig. 4.1: Proposed Architecture

FP-Growth Algorithm: FP-growth is a partitioning-based, depth-first search algorithm. It adopts a divide-and-conquer manner to decompose the mining task into many smaller tasks for finding frequent itemsets in conditional pattern bases. A conditional pattern base is a "sub-database" which consists of itemsets co-occurring with the prefix itemset. In, it proves, for an item i, the support of itemset {i [X} is equal to the support of item I in the conditional pattern base of itemset X. To efficiently generate conditional pattern bases, FP growth leverages 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 itemset 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 FP-growth, the process of mining frequent itemsets can be considered as first identifying frequent items in the database, constructing the conditional pattern bases of frequent items and mining frequent items in these conditional pattern bases and so on. In particular, FP-growth first scans the database to count the support of every item. The frequent items are inserted into the header table HT and sorted in decreasing order of their supports. Then, in the second database scan, FP-growth constructs a FP-tree for the database. For the frequent items in each transaction, they are arranged according to the order of HT and inserted into FP tree as a branch. If the branch has a prefix shared with some existing branch, the counter of the corresponding nodes in the existing branch is increased by one. The remaining suffix is inserted into the tree and the counter of each node is initiated to one.

Private FP-Growth Mining: Cache privacy guarantees that the presence or absence of an individual's information has little effect on the output of an algorithm and therefore an opponent can study partial information about any individual. In our context, the information contribute by an personality is her operation.

In particular for any database the set of nearest databases of τ, each one of which differs as of τ by at most one transaction. In cache patterns compression, when estimating potentially short frequent sequences, we can use a small number of patterns to replace the original consecutive patterns. Thus, consecutive patterns compression is particularly effective in cache sequences when we estimate short frequent sequences

Candidate Generation: Given a frequent pattern of size k, this step adjoins a frequent edge (which belongs to F1) with c to obtain a candidate pattern d of size k + 1. If d contains an additional vertex then the added edge is called a forward edge, or else it is call a backside edge; the later basically connect two of the existing vertices

of c. Additional vertex of a forward border is known an numeral id, which is the major integer id following the ids of the existing vertices of c; thus the apex-id stand for the sort in which the onward edges are adjoin while building a candidate pattern. In graph mining terminology, c is called the parent of d and d is a child of c and based on this parent-child relationship we can position the rest of applicant pattern of a removal task in a candidate production tree.

Cache Conversion: Cache conversion The irrelevant item deletion and cache patterns compression can effectively shrink sequences without incurring any frequency information loss. However, after applying these two schemes, some sequences might still violate the length constraint. We have to delete some items from the sequences until they meet the constraint. Clearly, if we randomly delete items, much frequency information in the sequences will be lost, which leads to inaccurate estimations of potentially frequent sequences. as many candidate sequences as possible. Formally, suppose the maximal length constraint is l max. Given the candidate k-sequences and a sequence S (|S| > l max), we aim to construct a new sequence S (|S| = l max), such that the digit of general applicant k-sequence contained in both S and ˆ S is maximized. A naive approach is to enumerate all possible sequences whose length is l max and find the sequence which contains the maximum number of common candidate sequences with S. However, this approach suffers from exponential time complexity. If there are |I| items, for example, we have to enumerate |I| l max sequences, which is computationally infeasible in practice.

Candidate Sequence Tree: It groups candidate sequences with identical prefix into the same branch. The height of the CS-tree is equal to the length of candidate sequences. Each node in the CS-tree is labeled by an item and associated with a sequence of items from the root to it. Figure 1 illustrates the CS-tree of the candidate 3-sequence set {abc, bcd, bda, bdb}. For a CS-tree constructed based on candidate k-sequences, the nodes at level k are associated with candidate k-sequences. We call the nodes at level k c-nodes, In addition, every node at level k-1 is associated with a (k-1)-sequence which can generate a candidate k-sequence by appending an item to it send. We call the nodes at level k-1 g-nodes and call their associated cache generating sequences.

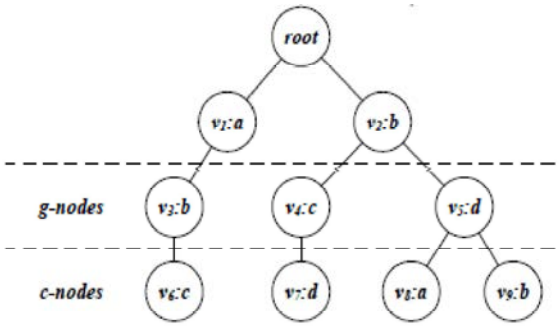

Fig. 4.2.3.1: Candidate Sequence Tree

Sequence Shrinking: The candidate k-sequences and a sequence S whose length exceeds the length constraint, we shrink S by the following steps. We first delete the irrelevant items from S. Then, we iteratively compress consecutive patterns in S in order of increasing pattern length. The irrelevant items deletion and consecutive patterns compression schemes do not cause the decrease of the support of candidate k-sequences. At last, if the length of S still violates the length constraint, we use the sequence reconstruction scheme to construct a new sequence, which might cause some frequency information loss.

Private Association Rule Mining: Let be the rest of every one of suitable regulations extracted from a table A cover of _ is a minimal set, such that any rule from is enclosed by a law in A rule belong to is called a _cover rule.

A cover summarizes the set of valid rules in a related way in which the huge itemsets review the set of private frequent itemsets. A cover can also be used to make simpler the appearance of rules to user: firstly, only cover rules could be shown to a user, then the user could select a cover rule I and retrieve a subset private of all rules covered by I and then the process could be repeated. During this technique, the client can carry out his investigate for policy without being overwhelmed by their number. A similar kind of regulation inquiry has been put forward, in the situation of the so called direction setting rules.

Algorithm Steps: Private Cover Rules) Algorithm for generating an informative cover for the valid rules. Let be a queue that will enclose repeated itemsets and let be the set of rules in which we will place the cover rules.

- Initialize by enqueuing into it all maximal private frequent itemsets, in decreasing order of their size. is <.0
- If is empty, then output and exit; else extract an itemset (I from.
- For all strict non-empty subsets > of I, with sorted primarily through their hold positive value (decrease) and second by their cardinality (increasingly), do:
- If the rule is suitable, then attach it to if it is not enclosed by a rule previously in Go to step 2.
- If I =1 and, then add to _ each subset of (that has dimension and that is not previously built-in in an itemset from Continue step 3.Algorithm private cover policy starts from the set of maximal private frequent itemsets and examines them in decreasing sort of cardinalities (steps 1-2). For each itemset (,search for a subset having maximum support, such that Such a rule is a candidate cover law and, on one occasion found, the look for stops and the law is added to the set of cover rules if it is not. Algorithm private Cover Rules starts from the set of maximal private frequent itemsets and examines them in decreasing array of their cardinalities (steps 1-2). For every such itemset (search for a subset ! having maximum support, such that EM(is a suitable association rule (step 3). Such a regulation is a candidate cover rule and, once found, the search stops and the regulation is additional to the place of envelop rules if it is not covered by one of the rules of (step 3.1). During the assessment of each separation of (we could come across some subsets such with the intention of they cannot be used as an precursor of a law based on top of the items of (in support of these subsets, we will contain to verify whether they know how to be experience of rules base on subsets)This is why, in step 3.2 of the algorithm, we attach to all the subsets. Those subsets that are already included in an itemset of though, do not require to be added further. Step 3.2 requirements to be perform only once, as a result we carry out it if the first subset examined in step 3 cannot be used as an predecessor. The collected works is a line for the reason that we would similar to to appear at the maximal ordinary itemsets in falling order of their size before we examine their subsets. We examine these itemsets in falling order of their dimension for the reason that a law whose set of substance is superior that cannot be enclosed by a law whose put of substance is slighter. This ensure that a wrap rule additional to cannot be covered by another cover rule that we can find out later. Every occasion that we propose to add a rule, however, we still need to check whether that law can be enclosed by one of the system previously.

RESULTS AND DISCUSSION

We evaluate the performance of the proposed algorithm PFP growth with association rule and compare with the algorithms Differentially Private Frequent on market dataset transaction datasets. All algorithms were written in Java programming verbal communication. The arrangement of the difficult stage is as follows: Windows 7 operating system, 4GB memory, Intel(R) Core dual CPU @ 2.60 Ghz.

Table 1: Frequent Itemsets of Supermarket dataset with different Support Value

Algorithm	0.1	0.2	0.3	0.4
Private Apriori	45	13	5	1
DPFP	71	15	8	2
PFGrowth-Rule	85	17	11	3

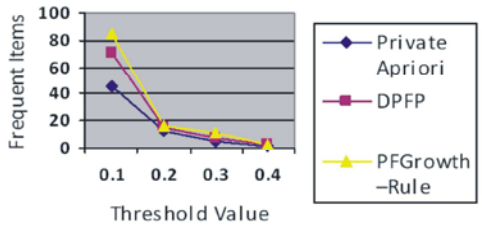

Fig. 6.1: Compare existing algorithm with proposed

Table 2: Memory of Supermarket dataset with different Support Value

Algorithm	0.1	0.2	0.3	0.4
Private Apriori	11.2	15.3	21.3	24.32
DPFP	8.43	10.4	15.32	19.43
PF Growth-Rule	6.348	8.2	5.27	6.68

Table 3: Times of Supermarket dataset with different Support Value

Private Apriori	0.1	0.2	0.3	0.4
DPFP	23	19.2	16.9	14.2
PF Growth-Rule	14	8.2	5.9	3.1
Private Apriori	5	3.8	3.4	2

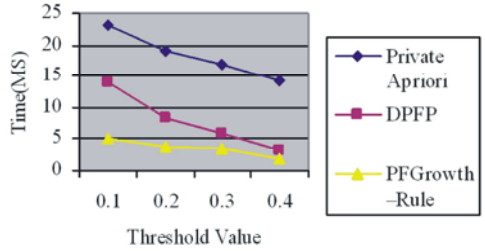

Fig. 6.2: Comparison of different algorithm time takes svstem Threshold value

CONCLUSION AND FUTURE WORK

The proposed private frequent itemset mining with rule generation the performance of Private CoverRules, as well as that of PFP-gen rules, slows down when the databases are denser and when the number of maximal frequent itemsets increase. The presentation of the algorithms varies in a different way with the change of minimum For dense databases, the measurement of the attach is individual-two commands of period lesser than the number of valid rules and shows the propensity of receiving smaller as the individual with no job in the generated system increases. There are many potential opportunities for future work. One such path would be to discover additional new complicated truncating algorithms. One more route would be to discover as substitute method to limit the information loss due to truncating. Finally, the achievement of our algorithm relies on the statement that the cache file transactions dominate the datasets. How to contract with datasets conquered by extended message is an unwrap trouble, even though with no constraints on the record our hypothetical outcome on confidentiality/ effectiveness tradeoffs propose that algorithms that concurrently attain good confidentiality and value may prove elusive.

REFERENCES

1. Agrawal, R. and J. Shafer, 1996. Parallel mining of association rules. In IEEE Transactions on Knowledge and Data Engineering, 8: 962-969, December 1996.
2. Agrawal, R. and R. Srikant, 2000. Fast algorithms for mining association rules. In Proc. of the 20 Int'l Conf. on Very Large Databases, SanTiago, Chile, June 1994.
3. Agrawal, R. and R. Srikant, 2000. Privacy-preserving data mining. In Proc. of the ACM SIGMOD Conference on Management of Data, pp: 439-450, May 2000.
4. Cheung, D., V. Ng, A. Fu and Y. Fu, 1996. A fast distributed algorithm for mining association rules. In Proc. of the 4 Int'l Conference on Parallel and Distributed Systems, pp: 31-42, Los Alamitos, USA, December 1996.
5. Evfimievski, A., R. Srikant, R. Agrawal and J. Gehrke, 2000. Privacy preserving mining of association rules. In Proc. of 8 ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, July 2002.
6. Gouda, K. and M. Zaki, 2001. Efficiently mining maximal frequent itemsets. In Proc. of the 1 IEEE Int'l Conference on Data Mining, San Jose, USA, November 2001.
7. Kantarcioglu, M. and C. Clifton, 2002. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, June 2002.
8. Lin, J. and M. Dunham, 1998. Mining association rules: Anti-skew algorithms. In Proc. of 14 IEEE Int'l Conf. on Data Engineering, October 1998.
9. Lindell, Y. and B. Pinkas, 2000. Privacy preserving data mining. Advances in Cryptology, 1880: 36-54, 2000.
10. Veloso, A., W. Meira Jr. and M. B. de Carvalho, 2002. Mining reliable models of associations in dynamic databases. In Proc. of the 17 Brazilian Symposium on Databases, pp: 263-277, Gramado, Brazil, October 2002.
11. Veloso, A., W. Meira Jr., M.B. de Carvalho, S. Parthasarathy and M. Zaki, 2003. Parallel, incremental and interactive mining for frequent itemsets in evolving databases. In Proc. of the 6 Int'l Workshop on High Performance, Pervasive and Stream Data Mining, San Francisco, USA, April 2003.
12. Veloso, A., W. Meira Jr, M.B. de Carvalho, B. Pˆossas, S. Parthasarathy and M. Zaki, 2002. ining frequent itemsets in evolving databases. In Proc. of the 2 SIAM Int'l Conf. on Data Mining, Arlington, USA, May 2002.
13. Veloso, A., W. Meira Jr., M.B. de Carvalho, B. Rocha, S. Parthasarathy and M. Zaki, 2002. Efficiently mining approximate models of associations in evolving databases. In Proc. Of the 6 Int'l Conf. on Principles and Practices of Data Mining and Knowledge Discovery in Databases, Helsinki, Finland, August 2002.
14. Zaki, M., S. Parthasarathy, M. Ogihara and W. Li, 1997. New algorithms for fast discovery of association rules. In Proc. of the 3 ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, August 1997.
15. Zaki, M., S. Parthasarathy, M. Ogihara and W. Li, 1997. New parallel algorithms for fast discovery of association rules. Data Mining and Knowledge Discovery: An International Journal, 4(1): 343-373, December 1997.

16. Bhuvaneswari, M., R. Rohini and B. Preetha, 2013. "A Survey on privacy preserving public auditing for secure data storage", International Journal of Engineering Research and Technology, Nov 2013.
17. Praveena, A., B. Anitha and R. Rohini, 2015. "Study of Iterative Mapreduce Techniques on Frequent Subgraph Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, November 2015, ISSN: 2277 128X.
18. Jayakumari, A., R. Rohini and B. Anitha, 2015. "Study of Privacy Preserving Frequent Itemset Mining Via Smart Splitting", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, November 2015, ISSN: 2277 128X.