Literature Review on data privacy or Information hiding process

Naveen Kumari[1], Tarun Dalal[2]

[1]M.Tech. Student (CSE Dept.), CBS Group of Institutions Jhajjar, Haryana
[2]Assitant Professor (CSE Dept.), CBS Group of Institutions Jhajjar, Haryana

ABSTRACT

Association Rule Mining from a large amount of data is one of the most important issue in data mining, because the discovered knowledge is commercially valuable. Sometimes companies involved in the similar business are often willing to cooperate each other so that they can perform data mining to extract knowledge from combined datasets. Generally the main objective behind such kind of data mining is mutual gain of all involved parties. But the company dataset contains private or sensitive data.

In this paper, a literature review to protect sensitive information using various efficient algorithms has been studied.

Keywords: data mining, sensitive, information, hiding, privacy.

INTRODUCTION

The market-basket problem assumes we have some large number of items, e.g., “bread," “milk." Customers fill their market baskets with some subset of the items, and we get to know what items people buy together, even if we don't know who they are. Marketers use this information to position items, and control the way typical customer traverses the store.

In addition to the marketing application, the same sort of question has the following uses:

1. Baskets = documents; items = words. Words appearing frequently together in documents may represent phrases or linked concepts. This concept can be used for intelligence gathering.
2. Baskets = sentences, items = documents. Two documents with many of the same sentences could represent plagiarism or mirror sites on the Web.

Framework for Frequent Itemset Mining

We use the term frequent itemset for “a set S that appears in at least fraction s of the baskets," where s is some chosen constant, typically 0.01 or 1%.

We assume data is too large to fit in main memory. Either it is stored in a RDB, say as a relation Baskets (BID; item) or as a flat file of records of the form (BID; item$_1$; item$_2$; : : : ; item$_n$). When evaluating the running time of algorithms Count the number of passes through the data. Since the principal cost is often the time it takes to read data from disk, the number of times we need to read each datum is often the best measure of running time of the algorithm.

There is a key principle, called monotonicity or the a-priori trick that helps us find frequent itemsets: If a set of items S is frequent (i.e., appears in at least fraction s of the baskets), then every subset of S is also frequent.

To find frequent itemsets, we can:

1. Proceed levelwise, finding first the frequent items (sets of size 1), then the frequent pairs, the frequent triples, etc. In

our discussion, we concentrate on finding frequent pairs because:

(a) Often, pairs are enough.
(b) In many data sets, the hardest part is finding the pairs; proceeding to higher levels takes less time than finding frequent pairs. Level wise algorithms use one pass per level.

2. Find all maximal frequent itemsets (i.e., sets S such that no proper superset of S is frequent) in one pass or a few passes.

Finding large Item Sets

The problem with Apriori is that it generates too many 2-itemsets that are not frequent. A new proposal is direct hashing and pruning (DHP) algorithm that reduced the size of candidate set by filtering any k-itemset out of the hash table if the hash entry does not have minimum support. This powerful filtering capability allows DHP to complete execution when Apriori is still at its second pass.

This algorithm is presented below:

```
$L_1$={large 1-itemsets};
FOR (k=2; $L_{k-1}$ != 0; i++ ) DO BEGIN $C_k$=apriori-
  gen($L_{k-1}$);
  FORALL transactions t in D DO BEGIN
    $C_t$=subset($C_k$,t);
    FORALL candidates c in $C_t$ DO
      c.count++; END
  $L_k$={c in $C_k$ | c.count >= minsup} END
Answer = Sum $L_k$;
FUNC apriori-gen(set $L_{k-1}$) BEGIN INSERT INTO $C_k$
  SELECT p.item$_1$, p.item$_2$,...,p.item$_{k-1}$,q.item$_{k-1}$
  FROM $L_{k-1}$ p, $L_{k-1}$ q
  WHERE p.item$_1$=q.item$_1$,...,p.item$_{k-2}$=q.item$_{k-2}$, p.item$_{k-1}$<q.item$_{k-1}$;
  FORALL itemset c in $C_k$ DO FORALL (k-1)-subsets s
    of c DO
              IF (s not in $L_{k-1}$) THEN
                DELETE c from $C_k$;
END
```

Algorithm 1: Algorithm Apriori

```
$L_1$={large 1-itemsets}; $C_1$'=database
D;
FOR (k=2; $L_{k-1}$ != 0; i++ ) DO BEGIN $C_k$=apriori-
  gen($L_{k-1}$);
  $C_k$'=0;
  FORALL entries t in $C_{k-1}$' DO BEGIN
    $C_t$={c in $C_k$ | (c-c[k]) in t.set-of-itemsets ^ (c-c[k-1]) in t.set-of-itemsets};
    FORALL candidates c in $C_t$ DO
      c.count++;
    IF ($C_t$ != 0) THEN $C_k'$ +=<t.TID,$C_t$>; END
  $L_k$={c in $C_k$ | c.count >= minsup} END
Answer = Sum $L_k$;
```

***Algorithm 2: Algorithm* AprioriTid**

Some further efforts to improve Apriori algorithm utilize parallel algorithm. There exist 3 parallel algorithms in literature based on Apriori to speed up mining of frequent itemsets. The Count Distribution (CD) algorithm minimizes communication at the expense of carrying out duplicate computations. The Data Distribution (DD) algorithm uses the main memory of the system to broadcast local data to all other nodes in the system. The Candidate Distribution algorithm is a load balancing algorithm that reduces synchronization between the processors and segments the database based upon different transaction patterns. These parallel algorithms were tested among each other and CD had the best performance against the Apriori algorithm. Its overhead is less than 7.5% when compared with Apriori..

Objectives

This paper discusses privacy and security issues that are likely to affect data mining projects. It introduces solutions to problems where the question is how to obtain data mining results without violating privacy, whereas standard data mining approaches would require a level of data access that violates privacy and security constraints. This paper aims to study to the solution of two specific problems. First, the problem of sharing sensitive knowledge by sanitization. Second, developing and improving algorithms for privacy in data mining tasks in scenarios which require multi-party computation.

LITERATURE REVIEW

The work in [1] proposed a hybrid method to hide a rule by decreasing either its support or its confidence. This method uses features of both ISL & DSR algorithms. This is done by decreasing the support or the confidence n units at a time by modifying the values of transactions.

In 2008, belwal et al. Presented an algorithm. To hide any specified association rule X → Y our algorithm works on the basis of confidence (X → Y) and support (X → Y).To hide the rule X → Y (containing sensitive element X on LHS),our algorithm increases the special variable of the rule X → Y until confidence (X → Y) goes below a minimum specified threshold confidence (MCT).As the confidence (X → Y) goes below MCT (minimum specified confidence threshold), rule X → Y is hidden i.e. it will not be discovered through data mining algorithm.

Actually any given specific rules to be hidden, many approaches for hiding association, classification and clustering rules have been proposed. Some of the researchers have used data perturbation techniques to modify the confidential data values in such a way that the approximate data mining results could be obtained from the modified version of the database. Some researchers also recognize the necessity of analyzing the various data mining algorithms in order to increase the efficiency of any adopted strategy that deals with disclosure limitation of sensitive data and knowledge.

Also disclosure limitation of sensitive knowledge by data mining algorithms, based on the retrieval of association rules, has been recently investigated. Proposed algorithm is also based on the reduction of support and confidence of sensitive rules but in this method some modified terms and some new variable are used to do the job. Also this work specifies that it can hide any given association rule, as some of the previous work cannot.

Association Rule Mining

Association rule mining finds interesting associations and/or correlation relationships among large sets of data items [1]. Association rules show attributes value conditions that occur frequently together in a given dataset. A typical and widely-used example of association rule mining is Market Basket Analysis [2].

For example, data are collected using bar-code scanners in supermarkets. Such market basket databases consist of a large number of transaction records. Each record lists all items bought by a customer on a single purchase transaction. Managers would be interested to know if certain groups of items are consistently purchased together. They could use this data for adjusting store layouts (placing items optimally with respect to each other), for cross-selling, for promotions, for catalog design and to identify customer segments based on buying patterns.

Association rules provide information of this type in the form of “if-then” statements. These rules are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature.

In addition to the antecedent (the “if” part) and the consequent (the “then” part), an association rule has two numbers that express the degree of uncertainty about the rule. In association analysis the antecedent and consequent are sets of items (called item sets) that are disjoint (do not have any items in common). The first number is called the support for the rule.

The support is simply the number of transactions that include all items in the antecedent and consequent parts of the rule (the support is sometimes expressed as a percentage of the total number of records in the database).

The other number is known as the confidence of the rule. Confidence is the ratio of the number of transactions that include all items in the consequent as well as the antecedent (namely, the support) to the number of transactions that include all items in the antecedent.

Association Rule Hiding

The problem of association rule hiding was first probed in 1999. After that, many approaches were proposed. Roughly, they can fall into two groups: data sanitization data modification approaches and knowledge sanitization data reconstruction approaches.

The basic idea of data modification approaches is the so-called data sanitization. They hide sensitive association rules by directly modifying, or we say, sanitizing the original data D, and get the released database D' directly from D, Most of the existing methods belong to this data modification prosperous track. According to different modification means, it can be further classified into: Data-Distortion techniques and Data-Blocking techniques. However, data modification approaches cannot control the hiding effects intuitively as the sanitization is performed on data level. Moreover, data sanitization can produce a lot of I/O operations, especially when the original database includes a large number of transactions [3].

The other solution towards the association rule hiding problem is the data reconstruction approaches. The basic idea is knowledge sanitization and data reconstruction. Unlike data modification, they put the original data aside and start from sanitizing the so-called "knowledge base" K. The new released data D' (apostrophe) is then reconstructed from the sanitized knowledge base K. This idea is inspired by the recently emerging inverse frequent set mining problem. The production in this track is very limited, involving only 3 papers to the best of my knowledge among which two papers are about classification rule hiding.

Privacy-Preserving Distributed Data Mining

A Distributed Data Mining (DDM) model assumes that the data sources are distributed across multiple sites. The challenge here is: how can we mine the data across the distributed sources securely or without either party disclosing its data to the others? Most of the algorithms developed in this field do not take privacy into account because the focus is on efficiency. A simple approach to mining private data over multiple sources is to run existing data mining tools at each site independently and combine the results. However, this approach failed to give valid results for the following reasons:

- Values for a single entity may be split across sources. Data mining at individual sites will be unable to detect cross-site correlations.
- The same item may be duplicated at different sites, and will be overweighed in the results.
- Data at a single site is likely to be from a homogeneous population. Important geographic or demographic distinctions between that population and others cannot be seen on a single site.

Recently, research has addressed classification using Bayesian Networks in vertically partitioned data [3], and situations where the distribution is itself interesting with respect to what is learned. Shenoy et al. proposed an efficient algorithm for vertically mining association rules. Finally, data mining algorithms that partition the data into subsets have been developed. However, none of this work has directly addressed privacy issues and concerns.

METHODOLOGY USED

The second phase is to perform sanitation algorithm over *FS*, which involves selecting the hiding strategy and identifying sensitive frequent itemsets according to sensitive association rules. In best case, the sanitation algorithm ensures from the sanitized set of frequent itemsets with supports and support counts (*FS'* in short in the figure) we can get exactly the set of non-sensitive rules with no normal rules lost an no ghost rules generated.

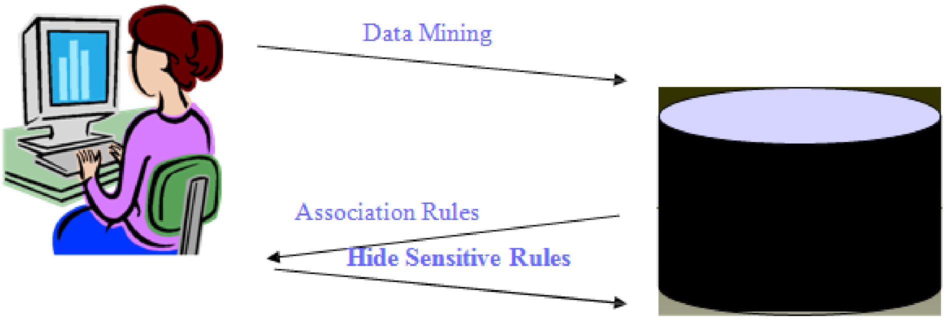
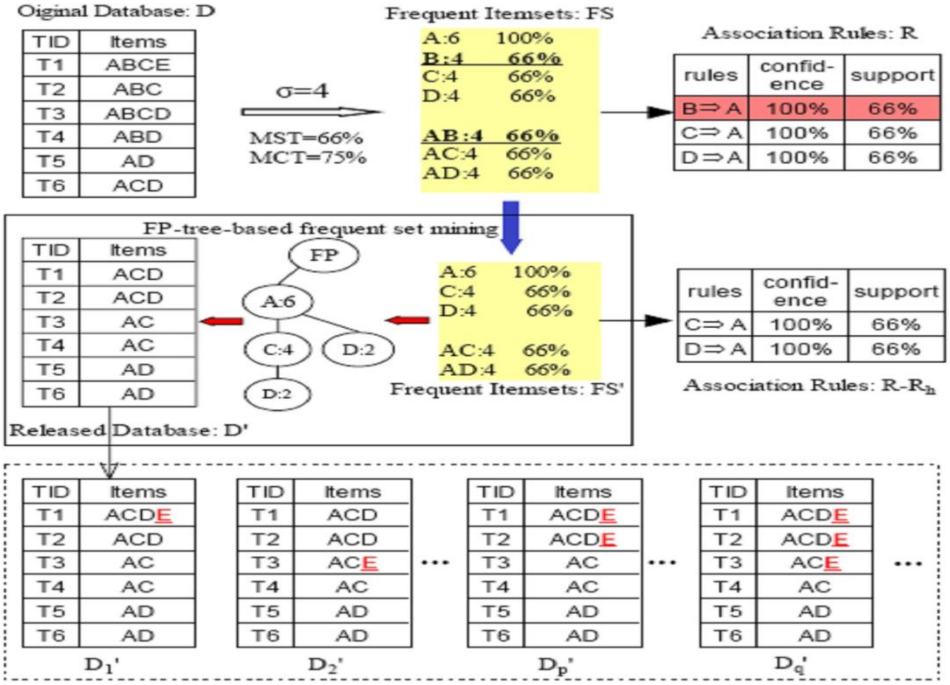


Figure 1: Hide Sensitive Association rules

To illustrate previous approach for the association rule hiding problem and validate its feasibility, let us consider an example shown in Figure 2.

First, compared with the early popular data sanitation algorithms, our sanitation algorithm is performed over the set of frequent itemsets with support counts, not on the original data. The set of frequent itemsets is much closer to the set of association rules than the data, which gives the database owner a more direct, visible and intuitive control towards the rules set. That is, by performing sanitation directly on knowledge level of data, one can control the discovered knowledge more handily. Second, compared with the recent emerging knowledge sanitation algorithm proposed in [4], our sanitation algorithm aims at hiding sensitive association rules, while theirs aims at hiding sensitive itemsets for simplicity. Usually, hiding sensitive rules is a more general, familiar and intuitive requirement than hiding sensitive itemsets. Another difference is that their sanitation algorithm performs on the whole itemsets space, while ours performs only on the small part of frequent itemsets, which can reduce much of sanitation cost. In Figure 2, given *I*= {*A*, *B*, *C*, *D*, *E*}, an original database *D*= {*T*1, *T*2, *T*3, *T*4, *T*5, *T*6}, minimum support count threshold *σ*=4, minimum support threshold MST=66%, minimum confidence threshold MCT=75%. All frequent itemsets, their support counts and their supports obtained from *D* are listed in the *FS*). All significant association rules obtained from the frequent itemsets in *FS* are shown in table *R*.



Oiginal Database: D

TID	Items
T1	ABCE
T2	ABC
T3	ABCD
T4	ABD
T5	AD
T6	ACD

Frequent Itemsets: FS

A:6	100%
B:4	66%
C:4	66%
D:4	66%
AB:4	66%
AC:4	66%
AD:4	66%

Association Rules: R

rules	confid-ence	support
B⇒A	100%	66%
C⇒A	100%	66%
D⇒A	100%	66%

Released Database: D'

TID	Items
T1	ACD
T2	ACD
T3	AC
T4	AC
T5	AD
T6	AD

Frequent Itemsets: FS'

A:6	100%
C:4	66%
D:4	66%
AC:4	66%
AD:4	66%

Association Rules: R-R_h

rules	confid-ence	support
C⇒A	100%	66%
D⇒A	100%	66%

D_1'

TID	Items
T1	ACDE
T2	ACD
T3	AC
T4	AC
T5	AD
T6	AD

D_2'

TID	Items
T1	ACD
T2	ACD
T3	ACE
T4	AC
T5	AD
T6	AD

D_p'

TID	Items
T1	ACDE
T2	ACDE
T3	AC
T4	AC
T5	AD
T6	AD

D_q'

TID	Items
T1	ACDE
T2	ACDE
T3	ACE
T4	AC
T5	AD
T6	AD

Figure 2: An example for knowledge hiding

Let us suppose *B* € *A* is a sensitive rule that needs hiding. First, instead of performing sanitation on the original database, we perform sanitation algorithm on *FS* by deleting the sensitive frequent itemsets *B*: 4 and *AB*: 4.

Here, we select hiding a sensitive rule by reducing the support of its corresponding large itemsets. Furthermore, we adopt thorough hiding strategy meaning that the large itemset the sensitive rule corresponds to needs to be completely hided and its support is reduced to zero. So after the sanitation we get the frequent itemsets set *FS'* from which we can obtain the set of association rules $R\text{-}R_h$ exactly (with no normal rules lost and no ghost rules generated).

PROPOSED TECHNIQUE

To hide any specified association rule X → Y this algorithm works on the basis of confidence (X → Y) and support (X → Y). To hide any sensitive rule X → Y, this algorithm first finds the value of support (sup) and confidence (conf) in the available set of rules and then it computes the support and confidence of the sensitive rule using following:
Confidence (X → Y) = (conf * X factor);
Support (X → Y) = (sup * X factor);

Advantage of Proposed Technique:

1. Hides sensitive items on both sides left as well as right.
2. Execution time is less in comparison to previous algorithm.

If owner of the data wants to hide a sensitive item in left side as well as in right too. Then in that case the above mentioned algorithm can be updated to hide sensitive items on both sides. The algorithm is as follows:

Input:

1. A database of transactions
2. A database of rules
3. A set of sensitive items X
4. A minimum support threshold (MST)
5. A minimum confidence threshold (MCT)

Output:

A transformed database of rules with modified support and confidence where rules containing X will be hidden.

Procedure:

```
//find value of support and confidence
Select confidence into conf from database.
Select support into supp from database.
For each X
{
    // Now check all the rules containing sensitive element x.
    For each rule R which contain X on LHS OR RHS.
    {
        While (conf(R) >= MCT)
        {
            If Conf(R) > 90 percent then
                Set confidence(X → Y) = (conf * 1/3);
                Set support (X → Y) = (sup * 1/3);
            If conf (R) > 1 && conf R < 90 then
                Set confidence(X → Y) = (conf * 1/10);
                Set support (X → Y) = (sup * 1/10);
        }
    }
}
End of procedure
}
End of procedure
```

CONCLUSIONS

In this paper, the author has reviewed the various published literature based on information hiding process. Industries such as banking, insurance, medicine, and retailing commonly use data mining to reduce costs, enhance research, and increase sales. While data mining in general represents a significant advance in the type of analytical tools currently available, there are limitations to its capability. Although data mining can help reveal patterns and relationships, it does not tell the user the value or significance of these patterns. It does not tell the users which patterns are sensitive and which are not.

REFERENCES

[1]. Belwal, Varsheney, Khan, Sharma, Bhattacharya. Hiding sensitive association rules efficiently by introducing new variable hiding counter. Pages 130-134, 978-2008, IEEE.

[2]. Shyue-Liang Wang, Yu-Huei Lee, Steven Billis, Ayat Jafari Hiding Sensitive Items in Privacy Preserving Association Rule Mining, 2004. IEEE International Conference on Systems.

[3]. Vi-Hung Wu, Chia-Ming Chiang, and Arbee L.P. Chen, Senior Member, IEEE Computer Society Hiding Sensitive Association Rules with Limited Side Effects , VOL. 19, NO.1, January 2007. IEEE transactions on knowledge and data engineering,

[4]. J. Vaidya and C. Clifton. Privacy preserving naive bayes classifier for vertically partitioned data. In M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, editors, Proceedings of the 4th SIAM International Conference on Data Mining, pages 522–526, Lake Buena Vista, Florida, USA, April 2004. SIAM.

[5]. Ila Chandrakar, Manasa, Usha Rani, and Renuka. Hybrid Algorithm for Association Rule mining. Journal of Computer Science 6(12), pages 1494-1498, 2010.

[6]. Dansana Jayanti, DeyDebadutta and Kumar Raghvendra (2013) "A Novel Approach: CART Algorithm for Vertically Partitioned Database in Multi-Party Environment", Proc. of IEEE Conference on Information and Communication Technologies (ICT), pp. 829-834.

[7]. D. Karthikeswarant, V.M. Sudha , V.M. Suresh and A. Javed Sultan (2012) "A Pattern Based Framework For Privacy Preservation Through Association Rule Mining", IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM) , pp.816-821.

[8]. Kaur Gurtaptish, Malhotra Sheenam (2013) "A Hybrid Approach for Data Hiding using Cryptography Schemes", International Journal of Computer Trends and Technology (IJCTT).4(8),pp. 2917-2923.

[9]. HeMiao, Vittal Vijayand Zhang Junshan (2013) "Online Dynamic Security Assessment With Missing PMU Measurements: A Data Mining Approach", Proc. of IEEE Transaction On Power System.28 (2), pp. 1969-1977.

[10]. Gurpreet Kaundal, Sheveta Vashisht, Disquisition of a Novel Approach to Enhance Security in Data Mining, ISSN 2320-6802, Vol. 1, Issue X, Nov. 2013.

[11]. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In Proceedings the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 639–644, Edmonton, Alberta, Canada, July 2002. ACM Press.

[12]. C. Clifton and D. Marks. Security and privacy implications of data mining. In Workshop on Data Mining and Knowledge Discovery, pages 15–19, Montreal, Canada, February 1996. University of British Columbia, Department of Computer Science.

[13]. J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 206–215, Washington, D.C., USA, August 2003. ACM Press.

[14]. W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In V. Raskin, S. J. Greenwald, B. Timmerman, and D. M. Kienzle, editors, Proceedings of the New Security Paradigms Workshop, pages 13–22, Cloudcroft, New Mexico, USA, September 2001. ACM Press.

[15]. Y. Lindell and B. Pinkas. Privacy preserving data mining. In CRYPTO-00, volume 1880, pages 36–54, Santa Barbara, California, USA, 2000. Springer Verlag Lecture Notes in Computer Science.

[16]. Y. Li and M. Chen, "Enabling Multi-Level Trust in Privacy Preserving Data Mining," IEEE transactions on knowledge and data engineering, sept. 2012, pp 1598-1612.

[17]. Y. Saygin, V. S. Verykios, and A. K. Elmagarmid. Privacy preserving association rule mining. In Z. Yanchun, A. Umar, E. Lim, and M. Shan, editors, Proceedings of the 12th International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E- Business Systems (RIDE'02), pages 151–158, San Jose, California, USA, February 2002. IEEE Computer Society.