Privacy Preservation in Association Rule Mining

J. Sumithra Devi[1] and M.Ramakrishnan[2]

[1]Research Scholar, Department of Computer Science, Bharathiyar University,
Coimbatore, Tamil Nadu,India
[1] sumithrathanoj2008@gmail.com
[2]Chairperson, School of Information Technology, Madurai Kamaraj University,
Madurai, Tamil Nadu, India,
[2]ramkrishod@gmail.com

ABSTRACT

Data mining services require exact input data for their results to be meaningful. The privacy concerns may influence users to provide spurious information. With respect to mining association rules, whether users can be encouraged to provide correct information by ensuring that the mining process cannot, with any reasonable degree of certainty, violate their privacy. We present a scheme, based on probabilistic distortion of user data that can simultaneously provide a high degree of privacy to the user and retain a high level of accuracy in the mining results. The performance of the scheme is validated against representative real and synthetic datasets.

Key words: Data mining, probabilistic distortion, synthetic datasets

1. INTRODUCTION

Association rule mining is an important data mining concept which finds the interesting correlations among various items in the item sets. The explosion of new data mining techniques have increased the privacy risks because now it is possible to efficiently combine and interrogate enormous data stores, available on the web, in the search of previously unknown hidden patterns[1]. In order to make a publicly available system secure, we must ensure not only that private sensitive data have been trimmed out, but also to make sure that certain inference channels have been blocked as well. In other words it is not only the data but the hidden knowledge in this data, that should be made secure. Moreover, the need for making our system as open as possible to the degree that data sensitivity is not jeopardized[2].

There may be some sensitive information that can be extracted by malicious users. The sensitive information can be extracted in the form of association rules with now popular association rule mining tools. Although this seemingly generic rule does not contain any personal information, it jeopardizes the privacy of the female customers since knowing this rule, someone can sniff a Prosac using female customer. Some other rule could be very critical for the company itself such as buying patterns of very rich customers. Sensitivity of a rule is a semantic notion and it has a temporal dimension too[3].

For hiding sensitive rules, it is more desirable to replace a real value by an unknown value instead of placing a false value. This is said to be data sanitization. Misleading rules could be obtained from this sanitized data by using data mining tools. Critical applications require that the sanitization process place unknown values instead of false values[4]. Even for non-critical applications unknown values are preferable to false values when the data is going to be sold to another company. This is due to the fact that insertion of false values would degrade the quality of the released data. It also is not desirable to apply a trivial algorithm that hides data by deleting randomly since it will hide lots of rules as a side effect, degrading the quality of the released or sold data[5].

In this paper, we propose a framework for preserving privacy in association rules when the data set contains unknown values and we propose an innovative technique for hiding rules from a data set using unknown values. The rest of the paper is organized as follows. Paper starts by giving introduction and need for privacy preservation in association rule mining process. Section 2 provides literature review about the privacy need in ARM process. Section 3 introduces the proposed method and details explanation about the privacy preservation in association rule mining by reducing support and confidence is provided in section 4. The experimental results are discussed in section 5 and paper ends by briefing the findings as conclusion in section 6.

2. BACKGROUND

The problem expressed here is closely related to privacy preservation problem in data mining which many of the researchers have already been defined. Dasseni et. al.[6] have considered the problem of privacy preserving mining of association rules. The paper demonstrates how certain sensitive rules can be hidden by some data modification

techniques and they have proposed efficient heuristics for solving this problem.

Chang and Moskowitz [7] consider a solution of the database inference problem by using a new paradigm where decision tree analysis is combined with parsimonious downgrading. In their scheme high decides what not to downgrade based upon the rules that it thinks Low can infer and upon the importance of the information that Low should receive.

Clifton[8]investigates the techniques to address the basic problem of using non-sensitive data to infer sensitive data in the context of data mining. The goal was to accomplish privacy by ensuring that the data available to the adversary is only a sample of the data on which the adversary would like the rules to hold. Moreover, the security officer is able to draw a relationship between the sample size and the likelihood that the rules are correct.

Agrawal and Srinkant [9] proposed a data mining technique that incorporates privacy concerns and they consider the concrete case of building a decision tree classifier from training data in which the values of individual records have been perturbed. The idea is to use the perturbed data to accurately estimate the original distribution of the data values. By doing this, they are able to build classifiers whose accuracy is comparable to the accuracy of classifiers built with the original data.

Atallah et. al. [10] proved that the problem is NP-Hard. In the current work we are considering the same problem but instead of allowing random data modification, we have restricted ourselves to introducing "?" a special symbol that indicates that information is missing. Some changes to the original association rule discovery program are necessary for the introduction of heuristics based on this idea.

3 PROPOSED METHOD

Idea is to maintain privacy during association rule mining. First, we need to introduce a new symbol in the alphabet of an item. The possible set of values of an item in the new setting becomes {0,1,?}. For example, the value in the i^{th} position of a transaction is 1 if the transaction contains the i^{th} item and, the value is 0 otherwise. A "?" mark in the i^{th} position of a transaction means that we do not have any information regarding whether the transaction contains the i^{th} item or not. Here, we have a support interval, [minsup(A),maxsup(A)] where the actual support of itemset A can be any value between minsup(A) and maxsup(A). The minsup(A) is the percentage of the transactions that contain 1's for all the items in A and maxsup(A) is the percentage of the transactions that contain either 1 or "?" mark for all the items in A. The confidence formula should also be modified since it will also have a degree of uncertainty. Instead of a single value for the confidence of a rule A ⇒ B, we have a confidence interval [minconf(A ⇒ B),maxconf(A ⇒ B)], where the actual confidence of a rule A ⇒ B can be any value between minconf(A ⇒ B) and maxconf(A ⇒ B).

3.1 Sensitive Association Rule Hiding

The purpose of the rule hiding algorithms is to make the sensitive rules invisible to the association rule mining algorithms to keep the data quality as high as possible. In order to hide a rule A ⇒ B, we can either decrease the support of the itemset AB below the minimum support threshold, or we can decrease the confidence below the minimum confidence threshold[11]. This can be accomplished by placing "?" marks in place of the actual values to increase the uncertainty of the support and confidence of the rules (i.e., length of the support and confidence intervals). Considering the support interval and the minimum support threshold (MST) which is a point, we may have the following cases for an itemset A:

- A is hidden when the minsup(A) is greater than or equal to MST,
- A is still visible when maxsup(A) is smaller than MST,
- A is visible with a degree of uncertainty when minsup(A) ≤ MST ≤ maxsup(A)

The same reasoning applies to the confidence interval and the minimum confidence threshold (MCT). Note that it is possible for the support of a rule to be above the MST, and for the confidence to have a degree of uncertainty and vice versa. Also, both the confidence and the support may be above the threshold. From a rule hiding point of view, in order to hide a rule A ⇒ B by decreasing its support, the only way is to replace 1's by "?" marks for the items in AB. In this way, we will only change the minimum support value while the maximum support value will be the same. As we replace 1's by "?" marks for the items in AB, the minimum support value of A ⇒ B will decrease and after some point it will go below the minimum support threshold. In order to hide a rule, A ⇒ B, by decreasing its confidence, we can replace both 1's and 0's by the "?" mark.

4. ALGORITHMS FOR RULE HIDING

The algorithm for rule hiding is twofold. The first phase of the algorithm concentrates on hiding the rules by reducing the minimum support of the itemsets that generated these rules. Reducing the minimum confidence of the rules is done in the second phase. Support hiding is adequate against an association rule mining algorithm that uses support pruning to reduce the search space of rules which is usually the case for the currently available commercial products[12]. However, algorithms that can efficiently extract high confidence rules without support pruning have recently been developed. Therefore, we have also proposed an algorithm that hides rules by reducing their confidence. Based on the concepts of interval support and interval confidence that we introduced, we would like to reduce the minimum support and minimum confidence values below the MST, and MCT correspondingly by a certain safety margin SM. So, for a rule A ⇒ B, after the

hiding process the following inequalities should hold; minsup(A ⇒ B) ≤ MST − SM, and minconf(A ⇒ B) ≤ MCT − SM.

4.1 Rule Hiding by Reducing the Support

This algorithm hides sensitive rules by decreasing the minimum support of their generating itemsets until the minimum support is below the MST by SM. The item with the largest minimum support is hidden from the minimum length transaction. The generating itemsets of the rules in R h (set of sensitive rules) are considered for hiding. The generating itemsets of the rules in R h are stored in L h (set of large itemsets) and they are hidden one by one by decreasing their minimum support. The itemsets in L h are first sorted in descending order of their size and minimum support. Then, they are hidden starting from the largest itemset. If there are more than one itemsets of maximum size, then the one with the highest minimum support is selected for hiding. The algorithm works like follows: Let Z be the next itemset to be hidden. Algorithm hides Z by decreasing its support. The algorithm first sorts the items in Z in descending order of their minimum support, and sorts the transactions in T Z (transactions that support Z) in ascending order of their size. The size of a transaction is determined by the number of items it contains. At each step the item i ∈ Z, with highest minimum support is selected and a "?" mark is placed for that item in the transaction with minimum size. The execution stops after the support of the current rule to be hidden goes below the MST by SM.

After hiding an item from a transaction, the algorithm updates the minimum support of the remaining itemsets in L h together with the list of transactions that support them. The algorithm chooses the item with highest minimum support for placing a "?" mark with the intention that an item of high minimum support will have less side effects since it has many more transactions that support it compared to an item of low minimum support. The idea behind choosing the shortest transaction for removal is that, a short transaction will possibly have less side effects on the other itemsets than a long transaction. If rule A ⇒ B needs to be hidden, then we need to choose one of the transactions in {T1 ,T4 ,T5 }. When the shortest transaction, T4 among T1 ,T4 , and T 5 is chosen, then placing a question mark for either item A, or item B in T4 will only effect the rule A ⇒ B. However, if we had chosen T1 or T5 , then rules A ⇒ D and B ⇒ D would also be affected by a modification in the transaction.

4.2 Rule Hiding by Reducing the Confidence

We propose two approaches for rule hiding using confidence reduction. The first approach is based on replacing 1s by "?" marks, while the second approach replaces
0s with "?" marks. It is important to have these two different approaches for the safety of the rule hiding. If we only used the first approach, then it would be obvious that all the "?" marks are actually 1's. Therefore, the two approaches should be used in an interleaved fashion for rule hiding via confidence reduction. The simplest way of Interleaving could be to hide the first half of sensitive rules by the first approach and the second half using the second approach.

The first algorithm shown hides a sensitive rule r by decreasing the support of the generating itemset of r. The hiding process goes on until the minsup(r) or the minconf(r) goes below the MST and MCT thresholds by SM. The algorithm first generates the set T r of transactions that support r, and then counts the number of items supported by each transaction. T r is then sorted in ascending order of transaction size. In order to select the item in which we are going to place a "?" mark, we consider the impact on the rest of the rules. As a heuristic, the algorithm places a "?" mark for the item with the highest support in the minimum size transaction because of the same reason as we described in the case of rule hiding when the support of the generating itemsets was reduced.

The algorithm for rule hiding by support reduction are as follows

INPUT: a set L of large itemsets, the set L h of large itemsets to hide, the database D,
MST, and SM
OUTPUT: the database D modified by the deletion of the large itemsets in L h
Begin
1. Sort L h in descending order of size and minimum support of the large itemsets
Foreach Z in L h
{
2. Sort the transactions in T Z in ascending order of transaction size
3. N iterations = |T Z | − (MST − SM) × |D|
For k = 1 to N iterations do
{
4. Place a ? mark for the item with the largest minimum support of Z in the next transaction in T Z
5. Update the supports of the affected itemsets
6. Update the database, D
}
}
End

The second algorithm hides a rule r by increasing the maxsup(l r) via placing "?" marks in the place of the 0 values of items in l r . By increasing the maxsup(l r) will cause the minconf(r) to decrease. Given a rule r, the algorithm first generates the set T 0 l r of transactions that partially support l r but that do not support r r (the right hand side of the rule r). Then the number of items in l r contained in each transaction is counted. The transaction t that contains the highest number of items in l r is selected for processing, in order to make the minimum impact on the database. The 0 values for the items of l r that are not supported by t is replaced by "?" marks to in-

The algorithm for rule hiding using confidence reduction are as follows

INPUT: a set R h of rules to hide, the source database D, MCT, MST, and SM

OUTPUT: the database D transformed so that the rules in R h cannot be mined
Begin
Foreach rule r in R h do
{
1. T r = {t in D/t fully supports r}
2. for each t in T r count the number of items in t
3. sort the transactions in T r in ascending order of the number of items supported
Repeat until (minconf(r) < MCT − SM)
{
4. Choose the first transaction t ∈ T r
5. Choose the item j in r r with the highest minsup
6. Place a ? mark for the place of j in t
7. Recompute the minsup(r)
8. Recompute the minconf(r)
9. Recompute the minconf of other affected rules
10. remove t from T r
}
11. Remove r from R h
}

All the algorithms first sort a subset of transactions in the database with respect to the items they have or with respect to the particular items they support. Sorting N numbers is O(NlogN) in the general case, however in our case the length of the transactions has an upper bound which is very small compared to the size of the database. In such a case we can sort N transactions in O(N).

5 EXPERIMENTS

To test the efficiciency of the algorithm, Web usage data of the Microsoft web siteis used. The data was created by sampling and processing the www.microsoft.com logs and donated to the Machine Learning Data Repository stored at University of California at Irvine Web site. The Web log data keeps track of the use of Microsoft Web site by 38000 anonymous, randomly-selected users. For each user, the data records list all the areas of the Web sites that the user visited in a one week time frame. We used the training set only which has 32711 instances. Each instance represents an anonymous, randomly selected user of the Web site and corresponds to the transactions in market basket data. The number of attributes is 294 where each attribute is an area of the www.microsoft.com Web site and each attribute corresponds to an item in the store in the context of market basket data. Data cleaning is performed by removing the instances with less than or equal to non-zero attribute values and the resulting data set contained about 22k transactions.

The naive algorithm is used as a base for comparison with the rule and support reduction algorithms. As a first step, we run an Apriori based mining algorithm on the data with support 0.1%. We then obtained the rules out of the resulting large itemsets with 50% minimum support. The minimum confidence and support values are chosen with regard to typical minimum confidence and support thresholds from the literature. Sensitive rules should be determined by the domain experts. We did not have the necessary domain knowledge, therefore we randomly selected 5 different rules and assumed that they are sensitive in order to test the hiding strategies. We measured the CPU time requirement of the hiding strategies for different confidence values and it is observed that the hiding strategies hide the given rule set successfully in less than a second that is considerably less than the time for mining which takes 57 seconds for 0.1% support. The performance of the hiding strategies in terms of the side effects were also observed.

6 CONCLUSION

A new set of concepts for making association rules sensitive is presented in this paper, and association rule mining process is extended to account for sensitive association rules. Association rules is one category of data mining techniques; other data mining techniques should also be considered for securing both data and knowledge in a virtual e-business environment that is open-ended and vulnerable to all kinds of different malevolent attacks. Our experimental results indicate that deterministic algorithms for privacy preserving association rules are a promising framework for controlling disclosure of sensitive data and knowledge.

REFERENCES

[1] A. Amiri, 2007, Dare to share: Protecting sensitive knowledge with data sanitization. Decision Support Systems, pages 181–191.

[2] B. Parikh S.-L.Wang and A. Jafari, 2007, Hiding informative association rule sets. Expert Systems with Applications, pages 316–323.

[3] E. Bertino-Y. Saygin V. S. Verykios, A. K. Emagarmid and E. Dasseni, 2004, Association rule hiding. IEEE Transactions on Knowledge and Data Engineering, pages 434–447.

[4] C. M. Chiang Y. H. Wu and A. L. P. Chen, 2007, Hiding sensitive association rules with limited side effects. IEEE Transactions on Knowledge and Data Engineering, pages 29–42.

[5] V. S. Verykios Y. Saygin and C. W. Clifton, 2001, Using unknowns to prevent discovery of association rules. ACM SIGMOD Record, pages 45–54.

[6] D. Elena, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. Hiding Association Rules by using Confidence and Support. To appear in the Proceedings of Information Hiding Work shop, 2001.

[7] L. Chang and I. S. Moskowitz. Parsimonious Downgrading and Decision Trees Applied to the Inference Problem. Proceedings of the Workshop of New Security Paradigms, pages 82–89, 1999.

[8] C. Clifton. Using Sample Size to Limit Exposure to Data Mining. Journal of Computer Security, 8(4), 2000.
[9] R. Agrawal and R. Srikant. Privacy Preserving Data Mining. Proceedings of SIGMOD Conference, pages 45–52, 2000.
[10] M. J. Atallah, E. Bertino, A. K. Elmagarmid, M. Ibrahim, and V. S. Verykios. Disclosure Limitation of Sensitive Rules. Proceedings of IEEE Knowledge and Data Engineering Workshop, pages 45–52, November 1999.
[11] V. S. Verykios Y. Saygin and A. K. Elmagarmid, 2002, Privacy preserving association rule mining. In Proceedings of the 2002 International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE), pages 151–163.
[12] G.Gratzer, 2011, Lattice Theory: Foundation, 2010 Mathematics Subject Classification, Springer, Basel, AG. [22] G. Tuncel, G. Alpan, Risk assessment and management for supply chain networks: a case study, Computers in Industry 61(2010)250–259.

