Hiding Sensitive Association Rules on Stars

Shyue-Liang Wang[1], Tzung-Pei Hong[2], Yu-Chuan Tsai[3], Hung-Yu Kao[3]
[1]*Department of Information Management*
[2]*Department of Computer Science and Information Engineering*
National University of Kaohsiung
Kaohsiung, Taiwan 81148
[3]*Department of Computer Science and Information Engineering*
National Cheng Kung University
Tainan, Taiwan 70101

Abstract

Current technology for association rules hiding mostly applies to data stored in a single transaction table. This work presents a novel algorithm for hiding sensitive association rules in data warehouses. A data warehouse is typically made up of multiple dimension tables and a fact table as in a star schema. Based on the strategies of reducing the confidence of sensitive association rule and without constructing the whole joined table, the proposed algorithm can effectively hide multi-relational association rules. Examples and analyses are given to demonstrate the efficacy of the approach.

1. Introduction

Recent developments in privacy preserving data mining have proposed many efficient and practical techniques for hiding sensitive information that could have been discovered by data mining algorithms. There are roughly four broad areas of research in the field of privacy preserving data mining: privacy preserving data publishing, privacy preserving applications, utility issues, and distributed privacy with adversarial collaboration. In privacy-preserving applications area, it corresponds to designing data management and mining algorithms in such way that the results of association rule or classification rule mining can preserve the privacy of data. A classic example of such technique is association rule hiding, in which some of the association rules are suppressed in order to preserve privacy.

For a single data set, given specific rules or patterns to be hidden, many data altering techniques for hiding association rules have been proposed. They can be categorized into three basic approaches. The first approach [3,9] hides one rule at a time. It first selects transactions that contain the items in a give rule. It then tries to modify items, transaction by transaction, until the confidence or support of the rule falls below minimum confidence or minimum support. The modification is done by either removing items from the transaction or inserting new items to the transactions. The second approach deals with groups of restricted patterns or sensitive association rules at a time. It first selects the transactions that contain the intersecting patterns of a group of restricted patterns. Depending on the disclosure threshold given by users, it sanitizes a percentage of the selected transactions in order to hide the restricted patterns. The third approach [10,11] deals with hiding certain constrained classes of association rules. Once the proposed hiding items are given, the approach integrates the rule selection process into the hiding process. It hides one rule at a time by calculating the number of transactions required to sanitize and modify them accordingly.

However, in real life, a database is typically made up of multiple tables. For example, there are multiple dimension tables and a fact table in a star schema in a data warehouse. Although efficient mining techniques have been proposed to discover frequent itemsets and multi-relational association rules from multiple tables, few works have concentrated on hiding sensitive association rules on multi-relational databases. In this work, we present a novel algorithm for hiding sensitive association rules in data warehouses with star schema. Based on the strategies of reducing the confidence of sensitive association rule and without constructing the whole joined table, the proposed algorithm can

effectively hide multi-relational association rules. Examples and analyses are given to demonstrate the efficacy of the approach.

The rest of the paper is organized as follows. Section 2 presents the statement of the problem. Section 3 presents the proposed algorithm for hiding sensitive association rules from multiple tables. Section 4 shows an example of the proposed approach. Concluding remarks and future works are described in section 5.

2. Problem description

Association rule mining was first introduced in [1,2]. Let $I = \{ i_1, i_2, \cdots, i_m \}$ be a set of literals, called items. Given a set of transactions *D*, where each transaction T in *D* is a set of items such that $T \subseteq I$, an association rule is an expression $X \Rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \phi$. The confidence of an association rule is calculated as $| X \cup Y | / | X |$, where $|X|$ is the number of transactions containing *X* and $|X \cup Y|$ is the number of transactions containing both *X* and *Y*. The support of the rule is the percentage of transactions that contain both *X* and *Y*, which is calculated as $| X \cup Y | / N$, where *N* is the number of transactions in *D*. The problem of mining association rules is to find all rules that are greater than the user-specified minimum support and minimum confidence.

As an example, for a given database with six transactions {T_1={*ABC*}, T_2={*ABC*}, T_3={*ABC*}, T_4={*AB*}, T_5={*A*}, T_6={*AC*}}, a minimum support of 33% and a minimum confidence of 70%, nine association rules can be found as follows: *B*=>*A* (66%, 100%), *C*=>*A* (66%, 100%), *B*=>*C* (50%, 75%), *C*=>*B* (50%, 75%), *AB*=>*C* (50%, 75%), *AC*=>*B* (50%, 75%), *BC*=>*A*(50%, 100%), *C*=>*AB*(50%, 75%), *B*=>*AC*(50%, 75%), where the percentages inside the parentheses are supports and confidences respectively.

The objective of data mining is to extract hidden or potentially unknown but interesting rules or patterns from databases. However, the objective of privacy preserving data mining is to hide certain sensitive information so that they cannot be discovered through data mining techniques [3,9-11]. For association rule hiding, given a transaction database *D*, a minimum support, a minimum confidence and a set of sensitive association rules *X*, the objective is to minimally modify the database *D* such that no association rules in *X* will be discovered.

Continue from previous example with minimum support 33%, minimum confidence 70%, and a sensitive association rule { *C*=>*B* }, if transaction T_1 is modified from *ABC* to *AC*, then the rule *C*=>*B* (33%, 50%) will be hidden. However, rules *B*=>*C* (33%, 66%), *AB*=>*C* (33%, 66%), *B*=>*AC* (33%, 66%), *AC*=>*B* (33%, 50%), *C*=>*AB* (33%, 50%),will be lost as side effects.

The techniques in association rule mining has been extended to work on numerical data, categorical data, and others in more conventional databases. In a relational database, a set of relational tables may exist. A star schema in a data warehouse is typical made up of multiple dimension tables and a fact table. Consider the following star schema [6] with fact table, *ATMactivity(acct#, atm#, amount)*, and two dimension tables, *Customer(acct#, balance, zipcode, age)* and *ATM(atm#, type, zipcode, limit)*, as shown in Figure 1.

Customer Table

acct#	balance	zipcode	age
01	1000..1999	10023	20..29
02	1000..1999	10047	20..29
03	1000..1999	10035	20..29
04	2000..5000	10035	30..39
05	2000..5000	10023	30..39
06	1000..1999	10047	30..39

ATM Table

atm#	type	zipcode	limit
A	drive	10023	0..9999
B	out	10035	0..9999
C	out	10047	0..9999
D	in	10023	10000..19999
E	in	10035	10000..19999
F	in	10047	10000..19999

ATMactivity Table

acct#	atm#	amount
01	A	15..25
01	A	15..25
02	A	15..25
02	C	50..100
02	C	50..100
03	A	15..25
03	B	15..25
04	B	50..100
04	E	500..1000
05	A	15..25
05	A	15..25
05	D	50..100
06	C	50..100
06	F	500..1000

Figure 1. Fact table and dimension tables in star schema

If limited to single table, the association rule mining algorithms on transaction data can be easily extended and discover rules such as: *age(20..29) => balance(1000..1999)* from table *Customer*, and *type(in) => limit(10000..19999)* from table *ATM*, for each individual table. However, to discover cross table association rules such as *limit(0..9999) => balance(1000..1999)*, *limit(0..9999) =>age(20..29)*, all three tables must be joined. The significant redundancy in such a joined table would seriously degrade the performance of multi-relational association rule mining. To efficiently discover frequent itemsets and association rules across multiple tables, many techniques have been proposed [4-8]. In this work, we consider the problem of efficiently hiding sensitive multi-relational association rules. More specifically, given a fact table and a set of dimension tables in a star schema, a minimum support, a minimum confidence,

and a set of sensitive association rules to be hidden, the objective is to minimally modify the dimension tables such that no sensitive rules will be discovered. For example, given the three tables in Figure 1, minimum support = 0.4, minimum confidence = 0.6, and sensitive association rule *limit(0..9999) => balance(1000..1999)* to be hidden, if *balance(1000..1999)* from *acct#01* in *Customer* table is deleted (or suppressed), the rule *limit(0..9999) => balance(1000..1999)* will be hidden.

3. Proposed algorithm

To hide an association rule efficiently on multiple tables, two issues must be addressed. The first issue is how to calculate supports of itemsets efficiently and the second issue is how to reduce the confidence of an association rule by minimal modification of dimension tables.

To calculate the support of an itemset, one trivial approach is to join all tables together and calculate the supports using any frequent itemset mining algorithm for transaction data. We assume that quantitative attribute values (e.g. age and monetary amounts) are partitioned and treated as items. It is obvious that joining all tables will increase in size many folds. In large applications, the joining of all related tables cannot be realistically computed because of the many-to-many relationship blow up and large dimensionality. In addition, increase in both size and dimensionality presents a huge overhead to already expensive frequent itemset mining, even if the join can be computed. Instead of "joining-then-mining", we will adopt "mining-then-joining" approach in this work [6,7].

To reduce the confidence of an association rule $X=>Y$ with minimal modification, the strategy in the support-based and confidence-based distortion schemes is to either decrease its supports, ($|X|/N$ or $|X \cup Y|/N$), to be smaller than pre-specified minimum support or decrease its confidence ($|X \cup Y|/|X|$) to be smaller than pre-specified minimum confidence. To decrease the confidence of a rule, two strategies can be considered. The first strategy is to increase the support count of X, i.e., the left hand side of the rule, but not support count of $X \cup Y$. The second strategy is to decrease the support count of the itemset $X \cup Y$. For the second strategy, there are in fact two options. One option is to lower the support count of the itemset $X \cup Y$ so that it is smaller than pre-defined minimum support count. The other option is to lower the support count of the itemset $X \cup Y$ so that $|X \cup Y|/|X|$ is smaller than pre-defined minimum confidence. In addition, in the record containing both X and Y, if we decrease the support of Y only, it would reduce the confidence faster than reducing the support of X. In fact, we can pre-calculate the number of records required to hide the rule. If there is not enough record to lower the confidence of the rule, then the rule cannot be hidden. To decrease support count of an item, we will remove one item at a time in the selected record by deleting it or suppressing it (replaced by *). In this work, we will adopt the second strategy for the proposed algorithm.

Algorithm *MRDC*

Input: (1) a fact table *FT*, and a set of dimension tables DT_1, DT_2…,
(2) minimum support,
(3) minimum confidence,
(4) a set of sensitive association rules,

Output: fact table and a transformed set of dimension tables where the sensitive rules are hidden.

1. Scan fact table and build one V_t vector for each FK. A V_t stores the number of occurrences of FK values appeared in fact table;
2. For each dimension table, //for in-table 1-itemsets
   Build *TID* lists for each 1-itemset and calculate support counts of each itemset with respect to fact table using V_t;
3. For each hidden rule, $X \rightarrow Y$,
   //for in-table 2 or higher itemsets, use *TID* lists and V_t;
   //for cross-table 2 or higher itemsets, scan FT
   Find support counts of X, $X \cup Y$;
   If (confidence of $X \rightarrow Y \geq$ min_conf), then
      Calculate the number of Y items to be deleted from its dimension table;
      Find the *TIDs* containing item Y and delete the item Y from those transactions;
4. Output the fact table and modified dimension tables.

4. Example

This section shows an example to demonstrate the proposed algorithm in hiding sensitive association rules from multiple tables of star schema.

Given dimension tables *Customer*, *ATM*, fact table *ATMactivity* with min_support = 0.4, min_conf=0.6, and hidden rules = { *limit(0..9999) => balance(1000..1999), limit(0..9999) =>age(20..29)*}, the execution of the proposed algorithm is shown as follows.

Step 1: Scan fact table and build V_t for each FK acct# and atm#.

$V_{Customer}$

acct#	count
01	2
02	3
03	2
04	2
05	3
06	2

V_{ATM}

atm#	count
A	6
B	2
C	3
D	1
E	1
F	1

Step 2: Scan each dimension table, build *TID* lists for each 1-itemset, and calculate support counts of each itemset with respect to fact table using V_t;

From *Customer*

	1-item		*TID(acct#)*	V_t counts	Support count
x_1	balance	1000..1999	01,02,03,06	2+3+2+2	9
x_2	balance	2000..5000	04,05	2+3	5
x_3	zipcode	10023	01,05	2+3	5
x_4	zipcode	10035	03,04	2+2	4
x_5	zipcode	10047	02,06	3+2	5
x_6	age	20..29	01,02,03	2+3+2	7
x_7	age	30..39	04,05,06	2+3+2	7

From *ATM*

	1-item		*TID(atm#)*	V_t counts	Support count
y_1	type	drive	A	6	6
y_2	type	out	B,C	2+3	5
y_3	type	in	D,E,F	1+1+1	3
y_4	zipcode	10023	A,D	6+1	7
y_5	zipcode	10035	B,E	2+1	3
y_6	zipcode	10047	C,F	3+1	4
y_7	limit	0..9999	A,B,C	6+2+3	11
y_8	limit	10000..19999	D,E,F	1+1+1	3

Step 3: For each hidden rule, { *limit(0..9999) => balance(1000..1999), limit(0..9999) =>age(20..29)*}, calculate the supports and confidence. The supports of each of the itemsets are: support(*limit(0..9999)*)*=11/14*, support(*balance(1000..1999)*)*=9/14*, support(*age(20..29)*)*=7/14*. Since *limit(0..9999) and balance(1000..1999)* are from different tables, we scan fact table again to calculate support counts of 2 and higher cross-table itemsets. The support count of (*limit(0..9999), balance(1000..1999)*) is 8. Similarly, we can get support count of (*limit(0..9999), age(20..29)*) as 7, and support(*limit(0..9999), balance(1000..1999)*)*=8/14*, support(*limit(0..9999), age(20..29)*)*=7/14*. The confidence of the values are *confidence(limit(0..9999) => balance(1000..1999))= 8/11, confidence(limit(0..9999) => age(20..29))= 7/11*.

To reduce the confidence of the rule *limit(0..9999) => balance(1000..1999), 8/11≈ 0.73*, to be less than 0.6, we need to delete at least 2 transactions with *balance(1000..1999)*. This is because 6/11 ≈ 0.55 is the greatest confidence we can get that is less than minimum confidence 0.6. From the *TID* list of 1-itemset *balance(1000..1999)*, we find *acct#01* appears twice in fact table. Therefore we delete *balance(1000..1999)* from *acct#01* in the dimension table *Customer*. The confidence of the rule becomes 0.55 and the rule is hidden.

Similarly, for the rule *limit(0..9999) =>age(20..29)*, its confidence is *7/11≈0.64*. We need to delete at least one transaction with *age(20..29)*. However from the *Customer* list of 1-itemset *age(20..29)*, we find *acct#01* appears at least twice in fact table. Therefore we delete *age(20..29)* from *acct#01* in the dimension table *Customer*. The confidence of the rule becomes 0.45 and the rule is hidden. Therefore, by deleting *balance(1000..1999)* and *age(20..29)* from dimension table *Customer*, we can hide two the two rules *limit(0..9999) => balance(1000..1999), limit(0..9999) =>age(20..29).*

5. Conclusion

In this work, we have studied the association rule hiding problem on multi-relational databases. Current technology for association rule hiding mostly applies to single transaction table. We discuss two important issues to extend current techniques to deal with multi-table association rule hiding. A "mining-then-joining"-based algorithm is proposed. Examples illustrating the proposed approach are shown. In the future, we will examine and improve the side effects of the proposed approach. We will also consider utilizing better data structures and reducing the database scanning for better efficiency.

6. References

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", Proceedings of ACM SIGMOD International Conference on Management of Data, 207–216, *1993.*

[2] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases.", Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, 487-499, 1994.

[3] E. Dasseni, V. Verykios, A. Elmagarmid and E. Bertino, "Hiding Association Rules by Using Confidence and Support" in Proceedings of 4th

Information Hiding Workshop, 369-383, Pittsburgh, PA, 2001.

[4] L. Dehaspe and L. De Raedt. “Mining Association Rules in Multiple Relations”, Proceedings of the 7th International Workshop on Inductive Logic Programming, 125–132. 1997.

[5] J.F. Guo, W.F. Bian, and J. Li, “Multi-relational Association Mining with Guidance of User”, Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 704-709, 2007.

[6] V. C. Jensen, N. Soparkar, “Frequent Itemset Counting Across Multiple Tables”, Proceedings of the 4th Pacific-Asia Conference of Knowledge Discovery and Data Mining, Current Issues and New Applications, 49−61, 2000.

[7] K.K. Ng, W.C. Fu, K. Wang, “Mining Association Rules from Stars”, Proceedings of the 2002 IEEE International Conference on Data Mining, 322−329, 2002.

[8] S. Nijssen and J. Kok, “Faster Association Rules for Multiple Relations”, Proceedings of the 17th International Joint Conference on Artificial Intelligence, 891−896, 2001.

[9] V. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, “Association Rules Hiding”, IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, 434-447, April 2004.

[10] S.L. Wang, D. Patel, A. Jafari, and T.P. Hong, “Hiding Collaborative Recommendation Association Rule”, Applied Intelligence, Volume 27, No. 1, 67-77, August 2007.

[11] S.L. Wang, T.Z. Lai, T.P. Hong, and Y.L. Wu, “Hiding Collaborative Recommendation Association Rules on Horizontally Partitioned Data”, Intelligent Data Analysis, Vol. 14, No. 1, January 2010, 47 – 67.

[12] L.J. Xu, K.L. Xie, “A Novel Algorithm for Frequent Itemset Mining in Data Warehouses”, Journal of Zhejang University Science A, 7(2), 216-224, 2006.