Privacy-Preserving Frequent Pattern Mining Across Private Databases

Ada Wai-Chee Fu, Raymond Chi-Wing Wong
Department of Computer Science and Engineering
The Chinese University of Hong Kong
adafu,cwwong@cse.cuhk.edu.hk

Ke Wang
Department of Computer Science
Simon Fraser University, Canada
wangk@cs.sfu.ca

Abstract

Privacy consideration has much significance in the application of data mining. It is very important that the privacy of individual parties will not be exposed when data mining techniques are applied to a large collection of data about the parties. In many scenarios such as data warehousing or data integration, data from the different parties form a many-to-many schema. This paper addresses the problem of privacy-preserving frequent pattern mining in such a schema across two dimension sites. We assume that sites are not trusted and they are semi-honest. Our method is based on the concept of semi-join and does not involve data encryption which is used in most previous work. Experiments are conducted to study the efficiency of the proposed models.

1 Introduction

In recent years, there have been some major concerns about the privacy issues in data mining. While a lot of useful information can be uncovered by data mining techniques, it is a controversial issue for customers that their privacy may be compromised. Many organizations are cautious about their data and are reluctant to join in the exploration of possible benefits from data mining. Typically the mining result is statistical and refers to a large collection of cases as a summary and therefore does not pose any threat to the privacy of individuals. However, in the process of data mining, it is typical that some processing units will acquire the knowledge of data from other faculties down to each individual item, and it is there that the privacy issue may be significant.

In this paper, we study the mining of frequent patterns (itemsets) where data is resided in multiple sites. The mining of frequent patterns has significance not only in itself but also for other data mining tasks such as mining of association rules, correlations, sequences, episodes, classifiers and clusters. We shall study the case of mining frequent itemsets from a distributed database [6]. In such a distributed environment we examine the important subcase where a relational database is partitioned vertically across the sites. This is an important scenario since it is applicable to many situations where each site is responsible for a semantical segment of the entire data collection.

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 remaining tables are called the *dimension tables*. Some important applications are data warehousing and OLAP where star schemas are found mostly. [7] cites a real life example of Ford Motor and Firestone Tire where tires from a specific factory had problems in certain situation, resulting in about 800 injuries. The problem could not be found by mining the database at each party, but would have been discovered earlier if the joined information is mined. This is again a case of vertical partitioning and the center table will indicate how the tire information should be joined with the automobile information. Note that for investigating into car accidents, the contents of the tables in the figure should be constrained to data related to accidents, i.e. cars that have involved in accidents and their corresponding tires.

The benefits of the star schema include less space, efficiency, fewer tables, simplified joins, support of drilling in reports, and flexibility to meet business needs. In all the above applications of the star schema, privacy issue arises. Data warehousing or OLAP may involve different branches or locations that do not want to reveal local information to the other sites. Ford Motor and Firesone Tire would not want to disclose their own data, but they should both be interested to have the mined results from the combined set of data. The need of privacy in data integration is obvious because of multiple data owners who are from different parties.

We assume that each site contains data that is of privacy to the particular site and it is not appropriate to disclose the details of individual record in the site to the other sites. We consider the case when the sites involved in a star schema are **honest but curious** or **semi-honest** [4]. This means

1550-4786/05 $20.00 © 2005 IEEE

the sites will follow the protocol for data mining but may try to find out more knowledge from the processing. We assume that it is of benefit to both the dimension sites to obtain the proper mining results. Therefore they would all cooperate to follow the protocol. This is true in all the above mentioned applications since the sites have mutual interests. Our contributions include the following.

(1) We generalize the work for privacy-preserving frequent pattern mining for one-to-one relationship [7]: we mine frequent patterns across two parties with *many-to-many relationship* in the form of a star schema. It means that there is no cardinality constraint on the relationship.

(2) Our method is based on the concepts of semi-join and does not rely on encryption as in most other related work. To our knowledge this is the first such attempt.

2 Problem Definition

Consider a relational database with a *star schema*. There are multiple sites, say A and B, storing their own tables, T_A and T_B, respectively. There is also a special site F, which stores the relationship of the other sites in a table called the **fact table**, or simply FT. Sites A and B are called the **dimension sites** while F is called the **fact table site**. The table at each dimension site is called a **dimension table**, which has its own primary key and other attributes. The primary keys of transactions of tables T_A and T_B are denoted by a_i and b_i, respectively. For simplicity, we assume that such a primary key is also the corresponding *transaction id* ($tids$) and these transaction ids will be foreign keys in the fact table. Therefore, site F stores the relationship of T_A and T_B by storing the combinations of foreign keys a_i and b_i in FT.

The attributes-value pairs of a transaction or a record in the table T_A and T_B are denoted by x_i and y_i, respectively. For example, we may have $\{$season, summer$\}$ and $\{$occupation, academics$\}$, as some attribute-value pairs. An **itemset** is a set of attribute-value pairs, in 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.

Given a support threshold s, a **frequent itemset** from a table is one that appears at least s times in a table. We are going to mine the frequent itemsets that appear in the table resulting from joining the tables ($FT \bowtie T_A \bowtie T_B$) across all the different sites, without the risk of the disclosure of any sensitive information from one site to other sites.

We assume that site F keeps FT and each dimensional site P knows the columns in FT for attributes from P. All sites are not trusted but are assumed to be semi-honest. An example application of such a center site is a mediator in electronic trading, where the dimension sites will be the buyers and the sellers. In the car accident problem, the center site may be a mediator site for investigation about the car accidents.

3 Correct-but-Insecure Mining

First we consider a simpler version of the problem where all the tables are centralized at one site only. Therefore we do not have to worry about privacy preservation. We shall present a solution for this problem which has the property that no explicit joining of the tables is required. In the next section we shall show how this solution can be made secure. Our mechanism makes use of the concept of **semi-join** [3]. We shall make use of the common relational algebra operators of **select** (σ) and semi-join, which we denote by the symbol $\rhd$. [1] In particular, in a relation T_A, we typically return the $tids$ of tuples that contain the attributes values of X_A, which in relational algebra is written as $\sigma_{X_A}(T_A)$. For clarity, we denote this by $A\sigma(X_A)$.

The semi-join concept is used to return those tuples in a relation T_B which are the results of a "join" of T_B with another relation T_A with the conditions of either some selected tuple with tid a_n in T_A or selected attribute values X_A in T_A. However, in our case, the "join" result is actually determined in the fact table. We select those "joined tuples" which satisfy the additional conditions in T_A. We denote these two kinds of semi-join by $B \rhd (a_n)$ or $B \rhd (X_A)$. *A major strategy for privacy preservation is that we shall not semi-join the tables T_A and T_B by sending T_A or T_B to F. Instead, only the tid's that are in T_A for a_n will be sent to F for getting the required tid's in T_B.*

We have described three symbols $A\sigma(X_A)$, $B\rhd(a_n)$ and $B \rhd (X_A)$. Let Y_B be a set of attribute-value pairs in T_B. Let b_n be the tid of B. We also introduced three additional symbols $B\sigma(Y_B)$, $A \rhd (b_n)$ and $A \rhd (Y_B)$, which has a similar definition respectively.

The symbols just described so far are represented by *tid_list*, which is a list of elements in the form $tid(count)$, where tid is a transaction id in a dimension table and $count$ is the number of records with tid in the fact table FT.

In this section, we introduce a motivating example shown in Figure 1 so as to illustrate the defined symbols and to describe the overall framework of our frequent pattern mining algorithms in a star. We will make use of the concepts in this example to introduce the protocols for privacy-preserving mining in Section 4.

In this example, there are two dimension tables, T_A and T_B and one fact table FT. Each of T_A and T_B contains a primary key (or tid) and other attributes (or items). For instance, $a1$, $a2$, $a3$ and $a4$ are tid's in T_A, and $x1$, $x2$, $x3$, $x4$, $x5$ and $x6$ are items or attribute-value pairs in T_A. FT contains mappings of the tid's of T_A to the tid's of T_B.

There are the following steps in finding the frequent patterns. Details can be found in [5].

1. (a) $A\sigma(x_1) \quad = \quad \{a_1(4), a_3(2)\}, \quad A\sigma(x_3) \quad =$

[1] Recall that the semijoin of two relations r_1 and r_2, which we denote by $r_1 \rhd r_2$ is $\Pi_{R_1}(r_1 \bowtie r_2)$, where R_1 is the schema for r_1. Thus, $r_1 \rhd r_2$ selects those tuples of r_1 that contributes to $r_1 \bowtie r_2$.

T_A

tid	Items
a1	x1, x3, x5
a2	x2, x3, x6
a3	x1, x3, x6
a4	x1, x4, x6

FT

tid(A)	tid(B)
a1	b5
a1	b3
a1	b2
a3	b2
a3	b5
a1	b5

T_B

tid	Items
b1	y1, y3, y5
b2	y1, y3, y6
b3	y2, y4, y6
b4	y1, y4, y5
b5	y1, y4, y6

Figure 1. Example

$\{a_1(4), a_3(2)\}$
$A\sigma(x_1x_3) = A\sigma(x_1) \bigcap A\sigma(x_3) = \{a_1(4), a_3(2)\}$

(b) $B\sigma(y_1y_6) = \{b_2(2), b_5(3)\}$

2. $B \rhd (a_1) = \{b_2(1), b_3(1), b_5(2)\}$, $B \rhd (a_3) = \{b_2(1), b_5(1)\}$
$B \rhd (x_1x_3) = B \rhd (a_1) \bigcup B \rhd (a_3) = \{b_2(2), b_3(1), b_5(3)\}$

3 $B \rhd (x_1x_3) \bigcap B\sigma(y_1y_6) = \{b_2(2), b_5(3)\}$

4 The combined frequency = total count in the list $\{b_2(2), b_5(3)\}$ = 5

Consideration of Privacy Preservation: If the tables T_A, T_B and FT are located at different sites A, B and F, respectively, and privacy is a concern. Let us examine the above steps again. To compute $A\sigma(X_A)$, site A needs only the counts of its tid's such as a_1 and a_3 from F. If A sends the **contents** of $A\sigma(X_A)$ (meaning the corresponding tid_list) to F, F can compute $B \rhd (X_A)$, without any information about the **contents** of X_A (meaning the items or attribute-value pairs in X_A). That is, F dose not need to know that $X_A = x_1x_3$. F only gets a certain itemset ID that A has assigned to X_A. If F sends the contents of $B \rhd (X_A)$ (meaning the corresponding tid_list) to B, B can compute the intersection with $B\sigma(X_B)$ also without knowing that $X_A = x_1x_3$. Therefore the tid_lists are quite helpful in masking information about the private databases. In the following section, we introduce protocols based on these observations.

4 Privacy-Preserving Mining

There are two dimension sites A and B and a center site F. We assume that the fact table is only available to site F. Each dimension site P can know about columns in the fact table that correspond to attributes of the dimension table of P. The resulting frequent itemsets will be made known to the dimension sites but not to F. Each site is not trusted but is semi-honest. An example of such a model is in e-auction or other commercial transaction processing where the center site functions as a dealer or mediator and the dimension sites are the buyers and sellers.

1. **Round 1 -** Site A determines all the frequent itemsets at its site, which form the set F^A. Similarly B determines F^B.

   Site A computes and sends $A\sigma(X_A)$ to F for each frequent itemset in F^A.

   Site B computes $B\sigma(X_B)$ for all frequent itemsets at site B and sends these to F.

2. **Round 2 -** Site F computes $B \rhd (X_A)$ from FT for all X_A in F^A and then computes $B \rhd (X_A) \cap B\sigma(X_B)$ for all X_B in F^B.

   For all frequent itemsets X_AX_B, site F sends the id pair of X_A and X_B to B.

3. **Round 3 -** For each frequent itemset X_AX_B, site B sends id of X_A with the contents of X_B to site A.

   Site A inserts the contents of X_A to X_AX_B and forms all frequent itemsets.

4. **Round 4 -** All frequent itemsets will be sent from site A to site B but site F will not be notified of the results.

4.1 Analysis

Let us examine what is disclosed to each party in the above. Site A receives the information such as $X_Ay_1y_6$ from site B for the frequent itemset $X_Ay_1y_6$. Site A has the content of X_A and site A is allowed to know all frequent itemsets and therefore this is not extra knowledge.

Site F receives information about $A\sigma(X_A)$ and $B\sigma(X_B)$ for each frequent itemset X_A or X_B. However, F does not know the contents of X_A or X_B. Site F only knows that a certain set of tid's with their frequencies are mapped to some frequent itemset but it is not known what the frequent itemset might be.

Site B receives the information such as X_AX_B for frequent itemset X_AX_B. It also has the contents of X_B. Although site B does not get the content of a frequent itemset of site A such as X_A directly, we find that it may uncover such information in some cases. Consider the running example. Let $X_A = x_1x_3$ and $X_B = y_1y_6$. Site B knows that the ID of X_A is linked to y_1y_6. After Round 4, site B knows all the frequent itemsets. If X_AX_B, X_CX_B and X_DX_B are the only frequent itemsets that contain y_1y_6, and if it happens that $B \rhd (X_A)$ is a proper subset of $B \rhd (X_C)$ and a proper subset of $B \rhd (X_D)$, site B can deduce that $X_A = x_1x_3$ (X_C and X_D correspond to x_1 and x_3).

Hence site B may find out that $X_A = x_1x_3$. However since the id of X_A is arbitarily assigned by site A, the knowledge of this mapping does not give extra information about T_A to site B.

More Than Two Dimensions : The basic mechanism here can be generalized to cases with more than two dimension tables. Each dimension site D computes $D\sigma(X_D)$ and sends it to F for each frequent itemset X_D. Site F finds all the frequent itemsets in terms of itemset IDs X_AX_B... and sends these sets to one of the dimenstion sites. The dimension sites pass among themselves such itemsets and fill in the contents of $X_A, X_B, ...$. Again site F will not be notified of the mining results.

5 Empirical Study

The experiment was conducted with Pentium IV 3.2GHz PC with 2GB memory on the Linux platform. All algorithms were implemented in C/C++. In our experiment, we generated the synthetic data set in the same way as [5]. The following table shows the default parameter setting we used.

Parameter	Default Value
No. of Dimension Tables/Dimension Sites	2
No. of Trans in the Joined Table/Fact Table	100K
No. of Trans in Each Dimension Table	10K
No. of Attributes in Each Dimension Table	10
Size of Each Attribute Domain	10
Random Noise	50%
Support Threshold	0.01

We adopt the transfer rate of the ADSL [1] for the estimation of message cost during transmission between different sites. For simplicity, we assume both the upstream and the downstream rate is 128 kilobytes per second (kB/s) [1].

We have conducted the experiments to study the effect of the execution time of our proposed model with four factors - (1) the no. of transactions in the fact table, (2) the no. of attributes, (3) the percentage of noise and (4) the support threshold. The results are shown in Figure 2. The graphs show the execution time of the Correct-but-Insecure Mining (CIM) and the proposed privacy-preserving model - the Semi-Honest Model with Fact Table at Site F (SF). We also varied the maximum size of potentially frequent itemsets in the graphs. We tested two possible values of the maximum size of potentially frequent itemsets: 4 and 6. For intance,"Max. Size = 4" means the maximum size of potentially frequent itemsets = 4.

It is trivial to see that the execution time of the models increases when the no. of transactions in the fact table, the no. of attributes in each dimension table or the maximum size of potentially frequent itemsets increases and when the support threshold decreases because the complexity of the models increases.

The execution time of the models increases with the percentage of noise. Suppose there is a local itemset X_A which is not frequent but is near to be frequent when the data set is generated without the consideration of noise generation. After more noise is added, there will be a higher chance that this itemset X_A will become frequent. In other words, more noise can create more local frequent itemsets X_A. Thus, the step of combining frequent local itemsets from two sites, says X_A and X_B, to generate global frequent itemsets such as $X_A X_B$ leads to a longer execution time. In our experiments, if the default parameters are adopted (with the percentage of noise equal to 50%), on average, there are more than 4000 local frequent itemsets generated. If we change it to 30%, there are about 3000 local frequent itemsets.

In all graphs, the execution time of CIM is smaller than that of SF because CIM does not need any message passing.

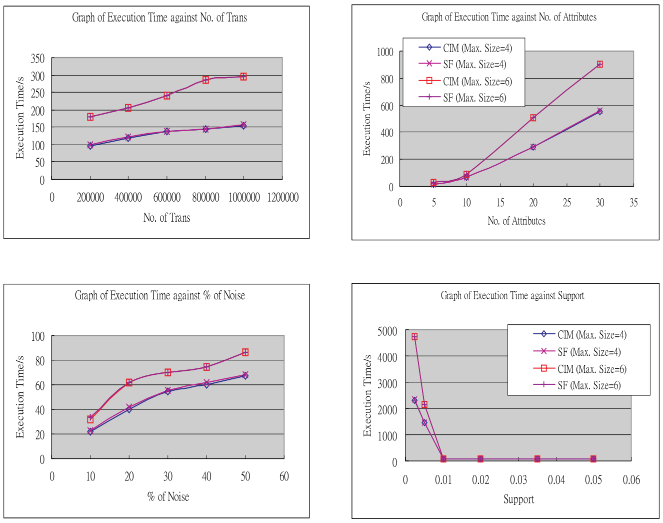

Figure 2. Graphs of Execution Time against Four Factors

6 Conclusion

In this paper, the problem of privacy-preserving frequent pattern mining in a star schema with two dimension sites is addressed. We show that with such a star schema it is possible to make use of semi-join for privacy preservation. We have also conducted some experiments to study the efficiency of the proposed models.

ACKNOWLEDGEMENTS: This research was supported by the RGC Earmarked Research Grant of HKSAR CUHK 4179/01E, and the Innovation and Technology Fund (ITF) in the HKSAR [ITS/069/03].

References

[1] List of device bandwidths. In *Wikipedia, http://en.wikipedia.org/wiki/List_of_device_bandwidths*.

[2] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *SIGMOD*, 2003.

[3] P.A. Bernstein and D. W. Chiu. Using semi-joins to solve relational queries. In *Journal of the ACM*, pages 25–40, 1981.

[4] B. Gilburd, A. Schuster, and R. Wolff. k-ttp: A new privacy model for large-scale distributed environments. In *SIGKDD*, 2004.

[5] E.K.K. Ng, A. Fu, and K. Wang. Association rule mining from stars. In *The 2002 IEEE International Conference on Data Mining (ICDM)*, pages 322–329, 2002.

[6] M.T. Ozsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 1999.

[7] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *SIGKDD*, 2002.