

Safely Delegating Data Mining Tasks

Ling Qiu^{1,a}Kok-Leong Ong²Siu Man Lui^{1,b}

¹ School of Maths, Physics & Information Technology
James Cook University

^a Townsville, QLD 4811, Australia

^b Cairns, QLD 4870, Australia

Email: {ling.qiu, carrie.lui}@jcu.edu.au

² School of Engineering & Information Technology
Deakin University

Geelong, VIC 3217, Australia

Email: leong@deakin.edu.au

Abstract

Data mining is playing an important role in decision making for business activities and governmental administration. Since many organizations or their divisions do not possess the in-house expertise and infrastructure for data mining, it is beneficial to delegate data mining tasks to external service providers. However, the organizations or divisions may lose of private information during the delegating process. In this paper, we present a Bloom filter based solution to enable organizations or their divisions to delegate the tasks of mining association rules while protecting data privacy. Our approach can achieve high precision in data mining by only trading-off storage requirements, instead of by trading-off the level of privacy preserving.

Keywords: Delegating, privacy preserving, Bloom filter, data mining.

1 Introduction

1.1 Background and Motivation

Data mining, as one of the IT services most needed by organizations, has been realized as an important way for discovering knowledge from the data and converting “data rich” to “knowledge rich” so as to assist strategic decision making. Padmanabhan *et al.* (2003) demonstrated the use of data mining for CRM (customer relationship management) applications in e-commerce. The benefits of using data mining for business and administrative problems have been demonstrated in various industries and governmental sectors, e.g., banking, insurance, direct-mail marketing, telecommunications, retails, and health care (Apte, C., Liu, Pednault & Smyth 2002). Among all the available data mining methods, the discovery of associations between business events or transactions is one of the most commonly used data mining techniques. Association rule mining has been an important application in decision support and marketing strategy (Lin, Q.-Y., Chen, Chen & Chen 2003).

We consider a typical application scenario as follows. In an organization (e.g., a governmental sec-

tor), there are several divisions including an IT division which provides IT services for the whole organization. A functional division may have to delegate its data mining tasks to the IT division because of two reasons: lack of IT expertise and lack of powerful computing resources which are usually centrally managed by the IT division. The data used for the data mining usually involves privacy that the functional division may not want to disclose to anyone outside the division. To preserve the data privacy, this division should first convert (or encrypt) the source data to another format of presentation before transferring to the IT division. Therefore, there are two factors which are important for enabling a functional division to delegate data mining tasks to the IT division: (1) the computational time of data conversion is less than that of data mining; otherwise it is not at all worthwhile to do so; and (2) the storage space of converted data should be acceptable (the less the better though, several times more is still acceptable and practical).

This scenario can be extended to a more general circumstance in which all divisions are individually independent organizations or companies. This is because in today’s fast-paced business environment, it is impossible for any single organization to understand, develop, and implement every information technology needed. It can also be extended to online scenarios, e.g., a distributed computing environment in which some edge servers undertaking delegated mining tasks may be intruded by hacking activities and may not be fully trusted.

When delegating mining tasks¹, we should protect the following three elements which may expose data privacy: (1) the source data which is the database of all transactions; (2) the mining requests which are itemsets of interests; and (3) the mining results which are frequent itemsets and association rules.

People have proposed various methods to preserve customer privacy in data mining for some scenarios, such as a distributed environment. However, those existing methods cannot protect all three elements simultaneously. This is because when a first party² delegates its mining tasks to a third party³, it has to provide the source database (which might be somewhat encrypted) together with some additional infor-

¹Without further specification, we always refer to association rule mining tasks.

²This is the party that delegates its data mining tasks. It may be a functional division in the scenario discussed above or a center server in a distributed environment with client-server architecture.

³This is the party that is authorized by the first party to undertake the delegated data mining tasks. It may be the IT division of an organization or an edge server in a distributed environment with client-server architecture.

mation (e.g., plain text of mining requests) without which this third party may not be able to carry out the mining tasks. Given this situation, those proposed methods are unable to efficiently prevent the exposure of private information to the third party, or unable to prevent the third party from deciphering further information from the mining results (which would be sent back to the first party) with the additional information.

1.2 Our Solution

In this paper, we present a Bloom filter based approach which provides an algorithm for privacy preserving association rule mining with computation efficiency and predictable (controllable) analysis precision. The Bloom filter (Bloom, B. 1970) is a stream (or a vector) of binary bits. It is a computationally efficient and irreversible coding scheme that can represent a set of objects while preserving privacy of the objects (technical details will be presented in Section 3.1).

With our approach, firstly the source data is converted to Bloom filter representation and handed over to a third party (e.g., the IT division of the organization) together with mining algorithms. Then the first party sends its mining requests to the third party. Mining requests are actually candidates of frequent itemsets which are also represented by Bloom filters. Lastly, the third party runs the mining algorithms with source data and mining requests, and comes out the mining results which are frequent itemsets or association rule represented by Bloom filters. In the above mining process, what the first party exposes to the third party does not violate privacy (Kantarcioglu, M., Jin & Clifton 2004); that is, the third party would not be able to distill down private information from Bloom filters. Therefore all the three elements mentioned above are fully protected by Bloom filters.

The goal of privacy preserving can be achieved by Bloom filter because it satisfies simultaneously the following three conditions. First, transactions containing different numbers of items are mapped to Bloom filters with the same length. This prevents an adversary from deciphering the compositions of transactions by analyzing the lengths of transactions. Second, Bloom filters support membership queries. This allows an authorized third party to carry out data mining tasks with only Bloom filters (i.e., Bloom filters of either transactions or candidates of frequent itemsets). Third, without knowing all possible individual items in the transactions, it is difficult to identify what items are included in the Bloom filter of a transaction by counting the numbers of 1's and 0's. This is because the probability of a bit in a Bloom filter being 1 or 0 is 0.5 given that the parameters of the Bloom filter are optimally chosen (see detailed mathematical analysis in (Qiu, L., Li & Wu 2006)).

The experimental results show that (1) the data conversion time is much less than mining time, which supports the worthiness to delegate mining tasks; (2) there is a tradeoff between storage space and mining precision; (3) there is a positive relationship between privacy security level and mining precision; (4) the converted data does not require more storage space compared with its original storage format.

1.3 Organization of the Paper

The remaining sections are organized as follows. Firstly in Section 2 we review the related work on privacy preserving data mining. After that in Section 3 we present our solution which uses a technique of keyed Bloom filters to encode the raw data, the data

mining requests, and also the results of data analysis during the data exchanges for privacy preserving. Next, we demonstrate in Section 4 the implementation of the proposed solution over a point-of-sale dataset and a web clickstream dataset. We present experiments which investigate the tradeoffs among the level of privacy control, analysis precisions, computational requirements, and storage requirements of our solution with comparisons over other mining methods. Lastly in Section 5, we conclude the paper with discussions of different application scenarios made possible by the solution, and point out some directions for further study.

2 Literature Review

Association rule mining has been an active research area since its introduction (Agrawal, R., Imilienski & Swami 1993). Various algorithms have been proposed to improve the performance of mining association rules and frequent itemsets. An interesting direction is the development of techniques that incorporate privacy concerns.

One type of these techniques is perturbation based, which perturbs the data to a certain degree before data mining so that the real values of sensitive data are obscured while statistics properties of the data are preserved. An early work of Agrawal and Srikant (2000) proposed a perturbation based approach for decision tree learning. Some recent work (Evfimievski, A., Srikant, Agrawal & Gehrke 2002, Rizvi, S. & Haritsa 2002, Atallah, M., Bertino, Elmagarmid, Ibrahim & Verykios 1999, Oliveira, S. & Zaiane 2003, Saygin, Y., Verykios & Clifton 2001) investigates the tradeoff between the extent of private information leakage and the degree of data mining accuracy. One problem of perturbation based approach is that it may introduce some false association rules. Another drawback of this approach is that it cannot always fully preserve privacy of data while achieving precision of mining results (Kargupta, H., Datta, Wang & Sivakumar 2003), the effect of the amount of perturbation of the data on the accuracy of mining results is unpredictable.

The second type of these techniques is distributed privacy preserving data mining (Pinkas, B. 2002, Vaidya, J. & Clifton 2002, Kantarcioglu, M. & Clifton 2002) based on secure multi-party computation. This approach is only applicable when there are multiple parties among which each possesses partial data for the overall mining process and wants to obtain any overall mining results without disclosing their own data source. Moreover, this method needs sophisticated protocols (secure multi-party computation based). These make it infeasible for our scenario.

Both types of techniques are designed to protect privacy by masquerading the original data. They are not designed to protect data privacy from the mining requests or the mining results, which are accessible by data miners.

Recently, Agrawal *et al.* (2004) presented an order-preserving encryption scheme for numeric data that allows comparison operations to be directly applied on encrypted data. However, encryption is time consuming and it may require auxiliary indices. It is only designed for certain type of queries and may not be suitable for complex tasks such as association rule mining.

3 Our Solution: Bloom Filter-Based Approach

From the literature, there is no single method that can enable organizations to delegate data mining tasks

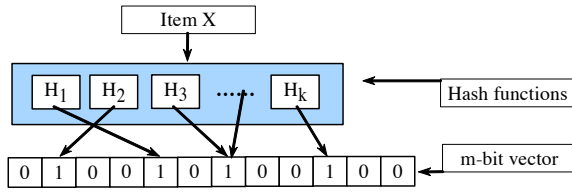


Figure 1: Constructing a Bloom filter of an item

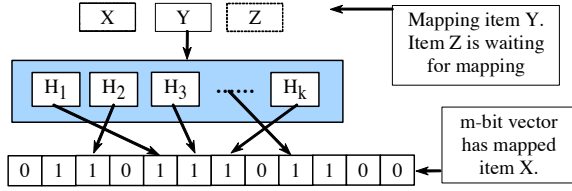


Figure 2: Constructing a Bloom filter of a transaction

while protecting all three elements involving the disclosure of privacy in the process of delegating data mining tasks. Our main objective in this paper is to propose a computationally feasible and efficient solution for the scenario.

A large number of examples from different industries (such as financial, medical, insurance, and retail) can be used for the study of the thread of privacy and business knowledge disclosure. In this paper, we consider the well-known association rule mining (Agrawal, R. et al. 1993), also known as market basket analysis (Chen, Y.-L., Tang, Shena & Hu 2005) in business analytics. Association rule mining can be performed by two steps: (1) mining of frequent itemsets, followed by (2) mining of association rules from frequent itemsets. Currently a well-known algorithm for mining of frequent itemsets is Apriori algorithm proposed by Agrawal and Srikant in (Agrawal, R. & Srikant 1994). Based on Apriori algorithm, in our early study (Qiu, L. et al. 2006) we investigated the feasibility of using a Bloom filter based approach for mining of frequent itemsets with privacy concerns. In this paper, we propose a solution with concerns of privacy protection by extending the Bloom filter-based approach to the whole process of association rule mining and applying to delegating scenario.

In what follows, we first introduce the mechanisms of constructing Bloom filters and membership queries over Bloom filters with discussion on the feature of privacy preserving. We then present algorithms of frequent itemset mining and association rule mining.

3.1 Bloom Filters

The Bloom filter (Bloom, B. 1970) is a computationally efficient hash-based probabilistic scheme that can represent a set of objects with minimal memory requirements. It can be used to answer membership queries with *zero* false negatives (i.e., without missing of useful information) and low false positives (i.e., with incurring of some extra results that are not of interests).

The mechanism of a Bloom filter contains (1) a binary vector (or stream) with length m and (2) k hash functions h_1, h_2, \dots, h_k of range from 1 to m . Given an item x , the Bloom filter of x , denoted as $B(x)$, is constructed by the following steps: (1) initialize by setting all bits of the vector with 0 and (2) set bit $h_i(x)$ (where $1 \leq i \leq k$) of the vector with 1. For example, if $h_2(x) = 7$, then bit 7 (i.e., the 7th bit) of the vector is set with 1. It is possible that several hash functions set the same bit of the vector, i.e., $h_i(x) = h_j(x)$. Thus, after conversion, the number of

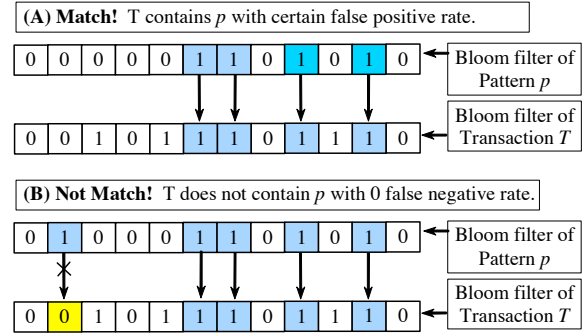


Figure 3: Membership query over Bloom filters

1's in $B(x)$ is not greater than k . Figure 1 illustrates how to construct a Bloom filter of an item.

It is similar to construct a Bloom filter of a transaction $T = \{X, Y, Z\}$ (or an itemset). Figure 2 illustrates the process in which item Y (or Z) is mapped onto the binary vector onto which item X (or items X and Y) has already been mapped. This process can be presented as $B(T) = B(X) \oplus B(Y) \oplus B(Z)$ where operator \oplus stands for bitwise *OR*⁴. It should be pointed out that converting the data into Bloom filters is an irreversible process. Any unauthorized party accessing to Bloom filters will have no way to know/infer the original value of the data represented by Bloom filters unless they have the access to the original data, the hash functions, and the secret keys (introduced later in this subsection).

To check whether a pattern p is contained by a transaction T , we examine whether $B(p) \otimes B(T) = B(p)$ holds, where operator \otimes stands for bitwise *AND*⁵. If $B(p) \otimes B(T) \neq B(p)$, then p is definitely not contained by T ; otherwise, p is a member of T with very low probability of false (i.e., false positive rate). Figure 3 shows the process of membership query with Bloom filters.

A Bloom filter does not incur any false negative, meaning that it will not suggest that a pattern is not in T if it is; but it may yield a false positive, meaning that it may suggest that a pattern is in T even though it is not. In our application, the false positive rate is upper-bounded by 0.5^k , where k is the number of hash functions, and the optimal value of k is given by $k = \frac{m}{n} \ln 2$, where m is the length of Bloom filters (i.e., the number of bits in the binary vectors), and n is the average length of transactions (i.e., the average number of items in transactions). Technical details for deriving the optimal value of k can be found in (Qiu, L. et al. 2006). Therefore, the false positive rate decreases exponentially with linear increase of the number of hash functions or the length of Bloom filters. For many applications, this is acceptable as long as the false positive rate is sufficiently small.

The privacy of data can be preserved by Bloom filters due to the irreversible feature. Given the above parameters of a Bloom filter, there are m^k possible mappings (for example, if we set the length of a Bloom filter $m = 80$ and the number of hash functions $k = 25$, then there are totally 80^{25} possible mappings in constructing the Bloom filter). Thus a Bloom filter can against some straightforward attacks (e.g., unknown-text attack and brute-force attack). It is certain that some other encryption algorithms (e.g., DES or RSA) are more secure; however, the computational cost is much more higher than

⁴The bitwise *OR* operation is defined as: $0 \oplus a = a$ and $1 \oplus a = 1$ where a is a binary variable 0 or 1.

⁵The bitwise *AND* operation is defined as: $0 \otimes a = 0$ and $1 \otimes a = a$ where a is a binary variable 0 or 1.

that of our method. The length of Bloom filters is a tradeoff between security level and computational cost. To enhance the security level, we insert a secret key K into each itemset or transaction before constructing its Bloom filter. The secret key K should not be chosen from the items. This amendment can be represented as $B_K(T) = B(T) \oplus B(K)$, in which $B_K(T)$ is referred to a keyed Bloom filter (see (Qiu, L. et al. 2006)). Without further mention, we always assume that Bloom filters are constructed with a secret key.

With the membership query mechanism of Bloom filters described above, we are able to conduct association rule mining with access to only Bloom filters. Given the irreversible feature of Bloom filters, a first party can convert all data involving disclosure of privacy to Bloom Filters and safely delegate the mining tasks to a third party without disclosing any value of the data in the database, the mining requests, and the mining results. We do not need to worry about missing of useful information (i.e., frequent itemsets and strong association rules in our application) due to *zero* false negative rate; but we may get some extra information (which may confuse data hacker while not affecting the quality of mining results) with low probability of false positive rate (see detailed mathematical analysis in (Qiu, L. et al. 2006)).

3.2 Mining Processes and Algorithms

The procedure of mining frequent itemsets is the process of membership queries over Bloom filters. Based on Apriori algorithm, a frame work of our method is shown in Algorithm 1. Algorithm 1 can be divided into three phases: *counting phase* (lines 3–5), *pruning phase* (lines 6–8), and *candidates generating phase* (lines 9–10) in each round ℓ , where ℓ indicates the size of each candidate itemset dealt with. In the counting phase, each candidate filter is checked against all transaction filters⁶ and the candidate's count is updated. In the pruning phase, any Bloom filter is eliminated from the candidate set if its count (i.e., $\text{Support}(x)$) is less than the given threshold $N \cdot \tau$. Finally, in the candidates generating phase, new candidate Bloom filters are generated from the Bloom filters discovered in the current round. The new candidates will be used for data mining in the next round. With the results of frequent itemsets, the mining of association rules is relatively simple, which is shown in Algorithm 2.

The complete process of association rule mining is given as follows. (1) The first party hands over to the third party the application software that performs frequent itemset mining together with the database of transactions represented by Bloom filters. (2) The first party sends to the third party mining requests which include candidate itemsets and the threshold of minimum support. The generation of candidates is done at the first party side by running Apriori_gen (Agrawal, R. & Srikant 1994) which is the critical step of Apriori algorithm. This step has to be done in the first party side because it involves data privacy (Qiu, L. et al. 2006). (3) The third party carries out mining tasks with the data received and finally returns the mining results which are Bloom filters of frequent itemsets together with their supports. (4) With frequent itemsets and their supports returned from the third party, it is easy to generate strong association rules with thresholds of minimum confidence. This job can be performed by the first party itself, or by the third party. If it is performed by the first party,

⁶All transactions are organized in a tree hierarchy so as to minimize the times of membership queries. See details in (Qiu, L. et al. 2006).

Algorithm 1 Mining of frequent itemsets from Bloom filters

```

1:  $C_1 = \{B(I_1), \dots, B(I_d)\}$ 
   //  $B(I_i)$  is the Bloom filter of item  $I_i$ 
2: for ( $\ell = 1$ ;  $C_\ell \neq \emptyset$ ;  $\ell++$ ) do
3:   for each  $B(S) \in C_\ell$  and each transaction filter  $B(T_i)$  do
4:     if  $B(S) \otimes B(T_i) = B(S)$  then
       Support( $S$ )++
       //  $S$  is a candidate frequent  $\ell$ -itemset
5:   end for
6:   for each  $B(S) \in C_\ell$  do
7:     if Support( $S$ ) <  $N \cdot \tau$  then
       delete  $B(S)$  from  $C_\ell$ 
       //  $N$  is transaction number in the database,
       // and  $\tau$  the threshold of minimum support
8:   end for
9:    $F_\ell = C_\ell$  //  $F_\ell$  is the collection of Bloom
                // filters of all "frequent"  $\ell$ -itemsets
10:   $C_{\ell+1} = \text{can\_gen}(F_\ell)$ 
        // generate filters of candidate
        // itemsets for the next round
11: end for
12: Answer =  $\bigcup_\ell F_\ell$ 
    // all filters of frequent itemsets

```

Algorithm 2 Mining of association rules from Bloom filters of frequent itemsets

```

1:  $AR = \emptyset$ ;  $F = \{B(F_1^1), \dots, B(F_{d_1}^1), B(F_1^2), \dots,$ 
    $B(F_{d_2}^2), \dots, B(F_1^k), \dots, B(F_{d_k}^k)\}$ 
   //  $B(F_i^s)$  is the Bloom filter of frequent
   //  $s$ -itemset  $F_i^s$  where  $1 \leq s \leq k$  and  $1 \leq i \leq d_s$ 
2: for ( $s = 1$ ;  $s < k$ ;  $s++$ ) do
3:   for ( $t = s + 1$ ;  $t \leq k$ ;  $t++$ ) do
4:     for each  $B(F_i^s)$  and each  $B(F_j^t)$  do
5:       if  $B(F_i^s) \otimes B(F_j^t) = B(F_i^s)$  and
         Support( $F_j^t$ )/Support( $F_i^s$ )  $\geq \xi$ 
           //  $\xi$  is the minimum threshold
           // of confidence
6:         then  $F_i^s \Rightarrow F_j^t - F_i^s$  is a strong association
           rule and is added to  $AR$ 
7:       end for
8:     end for
9:   end for
10: return  $AR$  // All strong association rules

```

there is no need to convert frequent itemsets to Bloom filters. If it is performed by the third party, for privacy considerations all data has to be converted to Bloom filters.

4 Experiments

4.1 Experimental Settings

We implement the solution and evaluate it with experiments on two real datasets BMS-WebView-2 and BMS-POS which are publicly available for research communities⁷. Dataset BMS-POS contains several years of point-of-sale data from a large electronic retailer; whereas dataset BMS-WebView-2 contains several months of clickstream data from an e-commerce website. Table 1 shows the number of items, the average size of transactions, and the number of transactions included in these datasets. Figure 4 shows the distribution of transaction sizes of the datasets. For dataset BMS-POS, a transaction

⁷Downloadable at <http://www.ecn.purdue.edu/KDDCUP>.

Table 1: Characteristics of real datasets

Dataset	Distinct items	Max-size	Average size	Number of transactions
BMS-POS	1,657	164	6.53	515,597
BMS-WebView-2	3,340	161	4.62	77,512

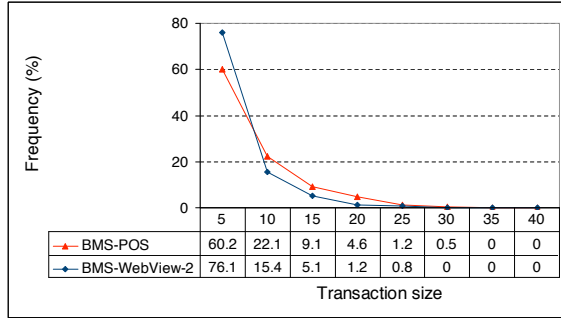


Figure 4: Distribution of transaction sizes

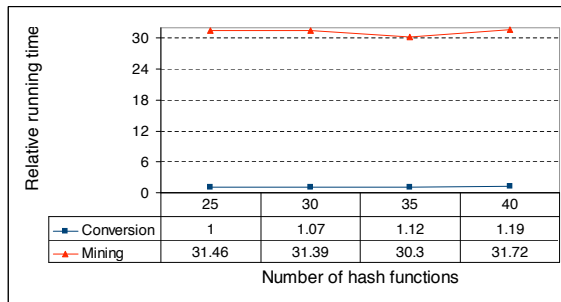


Figure 5: Data conversion time vs. mining time on dataset BMS-POS

is a list of items purchased in a basket; whereas for dataset BMS-WebView-2, a transaction is a browsing session which contains a list of webpages visited by a customer. The experiments are run on a Compaq desktop computer with Pentium-4 CPU clock rate of 3.00 GHz, 3.25 GB of RAM and 150 GB harddisk, with Microsoft Windows XP Professional SP2 as the operating system.

We have qualitatively analyzed the privacy preserving feature of Bloom filters in Section 3.1 (further theoretical analysis and discussions can be found in (Qiu, L. et al. 2006)). Therefore the emphasis of this set of experiments is to investigate the relationship among the level of privacy protection (determined by the number of hash functions), storage requirement, computation time, and analysis precision. In the experiments, we set the threshold of minimum support $\tau = 1\%$ and cluster the transactions in each dataset into 4 groups based on their transaction sizes (refer to (Qiu, L. et al. 2006) for technical details of grouping). We change k the number of hash functions used for Bloom filters from 25 to 40 in the experiments.

4.2 Experimental Results

Figures 5 and 6 show that the time of mining frequent itemsets is much more than the time of converting data to Bloom filter presentations, meaning that the mining process takes the major part of running time. This result verifies the worthiness in terms of running time for data format conversion before delegating mining tasks (satisfying the first factor enabling to delegate mining tasks as mentioned in Section 1).

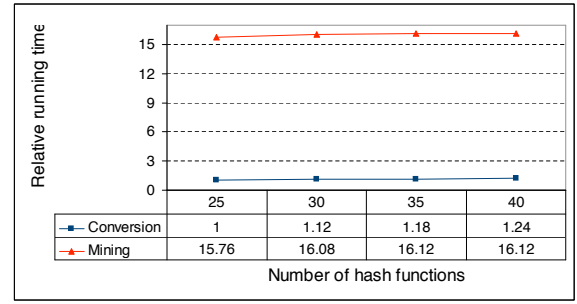


Figure 6: Data conversion time vs. mining time on dataset BMS-WebView-2

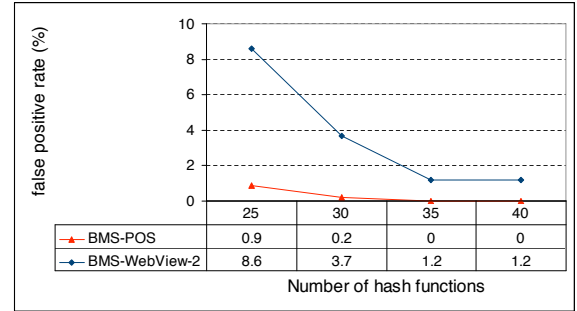


Figure 7: Mining precision vs. number of hash functions

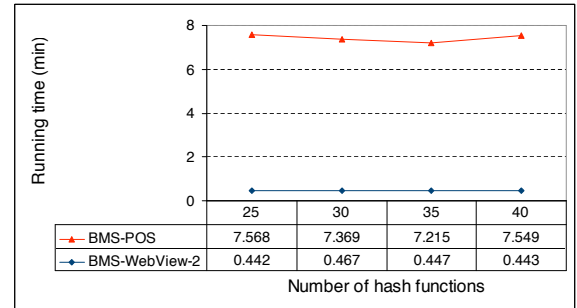


Figure 8: Running time vs. number of hash functions

Figure 7 shows the mining precisions with the change of k . There is a globally decreasing trend of false positive rates for each real dataset. For dataset BMS-POS, the false positive rate is less than 1% for $k \geq 25$. For dataset BMS-WebView-2 the false positive rate is below 10% for $k = 25$ and less than 4% for $k \geq 30$.

Figure 8 shows that the running time changes slightly with hash function number k . The running time is around 8 minutes for dataset BMS-POS and within 0.5 minute for dataset BMS-WebView-2, because comparatively dataset BMS-POS contains 7 times as many as transactions.

Figure 9 shows that the storage requirement is linearly increasing with k for both datasets. The reason is that the optimal value of k is given by $k = \frac{m}{n} \ln 2$ where m is the length of Bloom filters (Qiu, L. et al. 2006). The results of this experiment show that high mining precision can be achieved by increasing the number of hash functions. Consequently, the storage requirement increases linearly due to the use of longer Bloom filters.

Figure 10 shows a comparison of the average storage space required by a transaction under difference storage formats. The results show that the storage space of Bloom filter format is practical, i.e., it is less

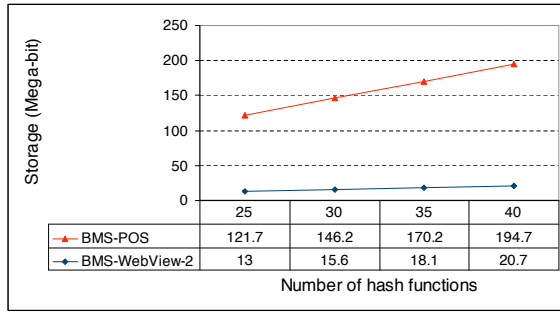


Figure 9: Storage requirement vs. number of hash functions

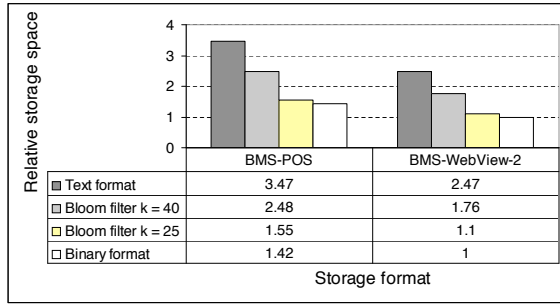


Figure 10: Average storage space of a transaction

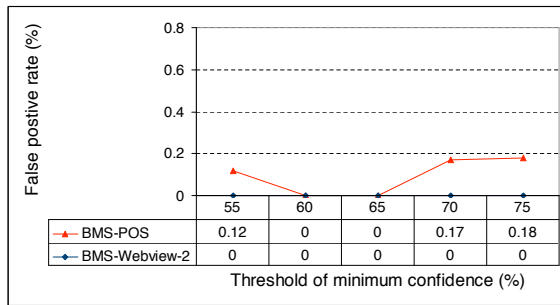


Figure 11: Precision of mining association rules

than text format but a bit more than binary format depending on the precision requirement (adjustable by k). This satisfies the second factor that it is worthwhile in term of storage space to adopt Bloom filter presentations as mentioned in Section 1. We can achieve further saving of storage space without decreasing mining precision with some techniques proposed by Qiu *et al.* (2006) (e.g., δ -folding and grouping).

With the mining results of frequent itemsets returning from the third party, we continue the mining of association rules with given threshold of minimum confidence. In this experiment, we let $k = 30$ under which the false positive rates of frequent itemsets are lower than 5% for both datasets. We vary the threshold of minimum confidence from 55% to 75%. The false negative rates are zero for both datasets, meaning that our approach does not miss useful information. As shown in Figure 11, the false positive rate is zero for dataset BMS-WebView-2 and lower than 0.2% for dataset BMS-POS. The running time for any dataset is less than 0.1 second.

5 Conclusions

In this paper, we have discussed and identified the risks of exposing data privacy in the scenario of del-

egating data mining tasks. We have also identified the factors that enable us to delegate mining tasks. We have proposed a privacy persevering data mining method and applied it to association rule mining in this delegation scenario. As compared with other existing methods, the metrics of our method include: (1) our approach is effective in protecting of three elements that can expose data privacy in the process of delegating mining tasks; (2) there is a positive relationship between the privacy security level and the analysis precision; (3) to increase the privacy security level, we only need to sacrifice data storage space; and (4) the solution is scalable, i.e., the storage space increases linearly with the privacy protection level or the analysis precision.

In our current study, we have developed a privacy protection method for association rule mining with a single (centralized) database. We can also apply our method to other mining tasks (e.g., mining of some other rules that are of interest to researchers). Further study to investigate the feasibility and implementation of the proposed solution in a multiple (distributed) databases environment is needed. Another future research direction could be investigating the feasibility of using the keyed Bloom filter approach in other tasks of business analytics.

References

- Agrawal, R., Imilienski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, in 'Proceedings of the ACM SIGMOD ICMD', pp. 207–216.
- Agrawal, R., Kiernan, J., Srikant, R. & Xu, Y. (2004), Order preserving encryption for numeric data, in 'Proceedings of the ACM SIGMOD ICMD', pp. 563–574.
- Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules in large databases, in 'Proceedings of VLDB'94', pp. 487–499.
- Agrawal, R. & Srikant, R. (2000), Privacy-preserving data mining, in 'Proceedings of the ACM SIGMOD ICMD', pp. 439–450.
- Apte, C., Liu, B., Pednault, E. & Smyth, P. (2002), 'Business applications of data mining', *Communications of the ACM* **45**(8), 49–53.
- Atallah, M., Bertino, E., Elmagarmid, A. K., Ibrahim, M. & Verykios, V. S. (1999), Disclosure limitation of sensitive rules, in 'Proceedings of the IEEE KDEE', pp. 45–52.
- Bloom, B. (1970), 'Space time tradeoffs in hash coding with allowable errors', *Communications of the ACM* **13**(7), 422–426.
- Chen, Y.-L., Tang, K., Shena, R.-J. & Hu, Y.-H. (2005), 'Market basket analysis in a multiple store environment', *Decision Support Systems* **40**(2), 339–354.
- Evfimievski, A., Srikant, R., Agrawal, R. & Gehrke, J. (2002), Privacy preserving mining of association rules, in 'Proceedings of the 8th ACM SIGKDD KDD 2002', pp. 217–228.
- Kantarcioglu, M. & Clifton, C. (2002), Privacy preserving distributed mining of association rules on horizontally partitioned data, in 'Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery', pp. 24–31.

- Kantarcioğlu, M., Jin, J. & Clifton, C. (2004), When do data mining results violate privacy?, *in* 'Proceedings of the 10th ACM SIGKDD KDD 2004', pp. 599–604.
- Kargupta, H., Datta, S., Wang, Q. & Sivakumar, K. (2003), On the privacy preserving properties of random data perturbation techniques, *in* 'Proceedings of the 3rd IEEE ICDM', pp. 99–106.
- Lin, Q.-Y., Chen, Y.-L., Chen, J.-S. & Chen, Y.-C. (2003), 'Mining inter-organizational retailing knowledge for an alliance formed by competitive firms', *Information & Management* **40**(5), 431–442.
- Oliveira, S. & Zaiane, O. (2003), Protecting sensitive knowledge by data sanitization, *in* 'Proceedings of the 3rd IEEE ICDM', pp. 211–218.
- Padmanabhan, B. & Tzhilin, A. (2003), 'On the use of optimization for data mining: theoretical interactions and eCRM opportunities', *Management Science* **49**(10), 1327–1343.
- Pinkas, B. (2002), 'Cryptographic techniques for privacy preserving data mining', *ACM SIGKDD Explorations* **4**(2), 12–19.
- Qiu, L., Li, Y. & Wu, X. (2006), 'Preserving privacy in association rule mining with Bloom filters', *Journal of Intelligent Information Systems*. In press.
- Rizvi, S. & Haritsa, J. (2002), Maintaining data privacy in association rule mining, *in* 'Proceedings of VLDB'02', pp. 682–693.
- Saygin, Y., Verykios, V. S. & Clifton, C. (2001), 'Using unknowns to prevent discovery of association rules', *Sigmod Record* **30**(4), 45–54.
- Vaidya, J. & Clifton, C. (2002), Privacy preserving association rule mining in vertically partitioned data, *in* 'Proceedings of the 8th ACM SIGKDD KDD', pp. 639–644.