Hiding Sensitive Patterns in Association Rules Mining

Guanling Lee*, Chien-Yu Chang and Arbee L.P Chen
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan 974, ROC
Email: guanling@mail.ndhu.edu.tw

Abstract

Data mining techniques have been developed in many applications. However, it also causes a threat to privacy. We investigate to find an appropriate balance between a need for privacy and information discovery on association patterns. In this paper, we propose an innovative technique for hiding sensitive patterns. In our approach, a sanitization matrix is defined. By multiplying the original transaction database and the sanitization matrix, a new database, which is sanitized for privacy concern, is gotten. Moreover, a set of experiments is performed to show the effectiveness of our approach.

Keywords: association patterns, privacy preservation, sanitized database, data mining.

1.Introduction

Data mining techniques have been developed in many applications and researches. However, it also brings the problem of privacy. A motivating example is discussed in [ASEG02]. Suppose we have a server and many clients in which each client has a set of data. The clients want the server to gather statistical information about association among items in order to provide recommendations to the customers. However, the clients do not want the server to know some *sensitive patterns*. Sensitive pattern is the frequent itemset that contain highly sensitive knowledge. Thus, when a client sends its database to the server, some sensitive patterns are hidden from its database according to some specific privacy policies. Therefore, the server only can gather statistical information from the modified database.

In recent years, more and more researchers emphasize the seriousness of the problem about privacy. The privacy problem can be classified into two classes: *data privacy* problem and *information privacy* problem. Data privacy is to protect the privacy of sensitive data, while information privacy is investigated privacy of patterns that contain highly sensitive knowledge.

Privacy-preserving mining in the context of data privacy for classification rules has been investigated in [AS00]. By using a randomizing function with Gaussian or Uniform perturbations, the sensitive values in user's record will be perturbed. They

---

* To whom all correspondence should be sent

proposed a reconstruction procedure to estimate the distribution of original data values. Based on probabilistic distortion of user data, [RH02] demonstrates a scheme. In [EGS03], the problem of how to avoid privacy breaches in privacy preserving data mining is introduced.

Information privacy preserving problem is to hide the sensitive patterns or rules by updating the original database and with as little effect on non-sensitive patterns as possible. This problem is proved to be NP-Hard [ABE99]. Similar to [ABE99], the other heuristic method is proposed in [SVC01]. They falsify some value or replace known values with unknown values such as question marks.

In [SO02], a framework is proposed to enforce privacy in mining frequent itemsets. They bring up a new threshold "disclosure threshold" controlled by users. In the approach, the victim items that should be eliminated for each restrictive pattern are selected. And transaction retrieval engine is used to identify sensitive transactions for each restrictive pattern. Based on the disclosure threshold, the number of sensitive transactions is computed and the victim items are removed from the select transactions. In this paper, we propose an innovative technique for hiding sensitive patterns. By observing the relationship between sensitive patterns and non-sensitive patterns, a *sanitization matrix* is defined. By setting the entries in sanitization matrix to appropriate values and multiplying the original transaction database to the sanitization matrix, a *sanitized database* is gotten. The sanitized database is the database that has been modified for hiding sensitive patterns with privacy concern. Moreover, the non-sensitive patterns should be preserved as many as possible.

The reminder of this paper is organized as follows. The problem and the framework of our approach is presented in section 2. In section 3, the sanitizing algorithms are discussed. The metrics to estimate the performance of our approach is introduced in section 4. The experimental results are also reported in section 4. We conclude with a summary and directions for future work in section 5.

2. Basic Concept

2.1 Problem Formulation

The problem of discovering association patterns is defined as finding relationships between the occurrences of items within transactions[AS94]. For example, an association pattern might be "bread, milk support=10%", which means there are 10% of all transactions contain both items. In the association patterns, each pattern should have a measure of certainty associated with it that assesses the validity of the pattern. It is called support. The support of an association pattern refers to the percentage of task-relevant transaction for which the rule is true. Therefore, *minimum support* is defined to be the minimum threshold for an association pattern to be meaningful. A *frequent pattern* is the pattern that satisfies the minimum support.

In our approach, a transaction database **D** is represented as a matrix in which the rows represent transactions and the columns represent the items. If **D** contains m transactions and n kinds of items, **D** is represented as an mxn matrix. The entry $D_{t,i}$ is set to 1 if item i is purchased in transaction t. Otherwise, it is set to 0.

Our problem can be formulated as follows. Let **D** be a transaction database, **P** be the set of frequent patterns that can be mined from **D**. Let $\mathbf{P_h}$ denote a set of sensitive patterns that need to be hidden according to some security policies, and $\mathbf{P_h} \subset \mathbf{P}$. $\sim\mathbf{P_h}$ is the set of non-sensitive patterns. $\sim\mathbf{P_h} \cup \mathbf{P_h} = \mathbf{P}$. Our problem is to transform **D** into a sanitized database **D'** such that only the patterns belong to $\sim\mathbf{P_h}$ can be mined from **D'**.

There are three potential errors afetr transforming **D** into **D'**. The first error is that some sensitive patterns are hidden unsuccessfully. That is, some sensitive patterns can still be mined form **D'**. The second error is that some non-sensitive patterns cannot be mined from **D'**. And the third, new patterns may be produced in the sanitized database **D'**. Our goal is to eliminate the sensitive patterns with as little effect to the non-sensitive patterns as possible.

$$D' = D \quad X \quad S$$

Where

$$\mathbf{D=}\begin{pmatrix} 110\ldots \\ 111\ldots \\ 110\ldots \\ 101\ldots \\ 011\ldots \\ 100\ldots \end{pmatrix}$$ **: transaction database**

$$\mathbf{S=}\begin{pmatrix} 11\text{-}1.. \\ 11\ 1\ .. \\ \text{-}101.. \\ 1\text{-}10.. \\ 1\ 11.. \end{pmatrix}$$ **: sanitization matrix**

$$\mathbf{D'=}\begin{pmatrix} 100\ldots \\ 101\ldots \\ 100\ldots \\ 100\ldots \\ 010\ldots \\ 100\ldots \end{pmatrix}$$ **: sanitized database**

Figure1. Sanitization Framework

2.2 Sanitization matrix

In our approach, original database **D** is multiplied by a sanitization matrix (**S**) to get a sanitized database **D'**. Figure 1 shows the framework of the sanitization process. If **S** is an identity matrix (i.e., S_{ij} is 1 if i=j, otherwise, S_{ij} is 0), **D'** will be equal to **D**. By setting S_{ij} where $i \neq j$ to appropriate value, a sanitized database **D'** will be gotten. In the following, the basic concept of our approach is discussed.

2.2.1 New definition for the matrix multiplication

In our approach, the matrix multiplication method is defined as follows:

1. If D_{ti} equals zero, no multiplication proceeds on it. That is, D'_{ti} is set to 0 directly. This is because our goal is to hide the sensitive pattern by decreasing its support. Therefore, we only need to take care of how and when an entry with value of 1 in **D** should be converted to 0 in **D'**. Moreover, if an entry with value zero can be converted to 1, new patterns may be produced.
2. If the resulting value larger than 1, set it to 1.
3. If the resulting value smaller than 0, set it to 0.

2.2.2 The Setting of "-1"

A sensitive pattern is hidden by decreasing its support. The support of pattern {i, j} can be decreased by reducing the correlation between items {i} and {j} in **D**. That is, if D_{ti} and D_{tj} are both equal to 1, set D_{ti} or D_{tj} be 0 can reduce the support of {i, j}. If a sufficient amount of such entries could be modified, {i, j} will no longer be a frequent pattern. Refer to Figure 2. let minimum support be 50% and {1, 2} be a sensitive pattern. If S_{21} is set to -1, D'_{21}, D'_{41} will become 0. Oppositely, if S_{12} is set to -1, D'_{22}, D'_{42} will become 0.Therefore, the support of {1,2} can be decreased by setting S_{21} or S_{12} to -1. Moreover, if S_{ij} is set to –1, then for a transaction t, where D_{ti} and D_{tj} are both equal to 1, D'_{tj} will be 0.

$$\begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array}\\ \begin{array}{c}t1\\t2\\t3\\t4\end{array} & \begin{pmatrix}1&0&1\\ \mathbf{1}&\mathbf{1}&0\\0&0&1\\ \mathbf{1}&\mathbf{1}&1\end{pmatrix}\end{array} \mathrm{X} \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array}\\ \begin{array}{c}1\\2\\3\end{array} & \begin{array}{ccc}1&0&0\\ \mathbf{-1}&1&0\\0&0&1\end{array}\end{array} = \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array}\\ \begin{array}{c}t1\\t2\\t3\\t4\end{array} & \left(\begin{array}{ccc} & & \\ & & \\ & & \\ & & \end{array}\right.\end{array}$$

Figure 2. Setting r_{21}=-1 in S

2.2.3 The Setting of "1"

Setting appropriate entries in **S** to –1 can reduce the support of the sensitive patterns. However, it also leads to accidentally conceal the non-sensitive patterns. We remedy this defect by setting some entries in **S** to 1 to minimize the effect on losing non-sensitive patterns. Continue above example, the frequent patterns in **D** are {1,2} and {1,3}. Let {1,2} and {1,3} be the sensitive and non-sensitive pattern, respectively. If S_{21} is set to –1, D'_{21} and D'_{41} will be 0. As a result, the support of {1, 3} will be decreased to 25% in **D'** and no longer be a frequent pattern. To reserve the non-sensitive pattern {1,3} in **D'**, S_{31} is set to 1 to make D'_{t1} keep the same value as D_{t1} for those transaction t where D_{t1}=1and D_{t3}=1 as shown in Figure 3. The purpose of setting the specific entry to 1 is to reinforce the relation of {1, 3} and avoid eliminating {1, 3} accidentally. Setting corresponding entries between any two items contained in non-sensitive patterns in **S** to 1 can preserve non-sensitive patterns after the sanitization process.

$$
\begin{array}{c} \\ t1 \\ t2 \\ t3 \\ t4 \end{array}
\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 1 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \end{array}
\times
\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}
\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \mathbf{R} \\ \begin{pmatrix} 1 & 0 & 0 \\ \mathbf{-1} & 1 & 0 \\ \mathbf{1} & 0 & 1 \end{pmatrix} \end{array}
=
\begin{array}{c} \\ t1 \\ t2 \\ t3 \\ t4 \end{array}
\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \mathbf{D'} \\ \begin{pmatrix} 1 & 0 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 1 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \end{array}
$$

D **S** **D'**

Figure3. Setting S_{21}=-1 and S_{31}=1 in S

3.The Sanitization Algorithms

In this section, three algorithms for hiding sensitive patterns are proposed.

3.1 Hidden-First Algorithm

The main idea of Hidden-First algorithm, denoted by HF, is to eliminate all patterns in $\mathbf{P_h}$ from **D** by setting proper entries in **S** to –1. The entries of **S** are set to the proper values according to the following rules.

1. S_{ii} =1, diagonal entry.
2. S_{ij} = **-1**, If $\exists\ \rho \in \mathbf{P_h}$, such that $\{i, j\} \subseteq \rho$ and $\forall\ \rho' \in \mathbf{\sim P_h}$, $\{i, j\} \not\subset \rho'$. (That is, {i, j} is a subpatttern of some patterns belong to $\mathbf{P_h}$ but not a subpattern of some patterns belong to $\mathbf{\sim P_h}$.) Moreover, the number of patterns containing {j} in $\mathbf{\sim P_h}$ is smaller than that of the patterns containing {i} in $\mathbf{\sim P_h}$. The reason is that by setting S_{ij} to –1, the support of item j will be reduced. Moreover, item j has smaller effect on $\mathbf{\sim P_h}$ than item i.
3. S_{ij} = 0, otherwise.

<u>Hidden-First Algorithm</u>

Input: Ph, ~Ph, D, S

Output: D'

Step 1: Set the values of the entries in S according to the rules.

Step 2: (matrices multiplication)

For every transaction i in **D** do

  For j=1 to number of items do

    If (D_{ij}=0) D'_{ij}=0;

    Else $D'_{ij}=\max\left(\sum_{k=1}^{number\ of\ items} D_{ik} \times S_{kj}, 0\right)$

  end-for

end-for

Refer to Figure 4, minimum support is 30%, $\mathbf{P_h}$: {{4,5}, {1,2,5}}, ~$\mathbf{P_h}$: {{2,4}, {1,3,5}}. After sanitization process, $\mathbf{P_h}$: {ϕ}, ~$\mathbf{P_h}$: {{4}, {3,5}}. HF algorithm can hide all the sensitive patterns successfully. However, some non-sensitive patterns may be accidentally hidden due to setting the value of some entries in **S** to –1.

$$
\begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6 \end{array} &
\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}
\end{array}
\times
\begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & -1 & 1 \end{pmatrix}
\end{array}
=
\begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6 \end{array} &
\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}
\end{array}
$$

Figure 4. Illustration of HF

3.2 The Non-Hidden-First Algorithm (NHF)

The main idea behind the Non-Hidden-First algorithm, denoted by NHF, is to reserve all non-sensitive patterns and endeavor to hide sensitive patterns from **D** at the same time.

The entries in **S** are set according to the following rules.

1. S_{ii} =1, diagonal entry.
3. S_{ij} = **-1**, If ∃ ρ ∈ $\mathbf{P_h}$, such that {i, j} ⊆ ρ and ∀ ρ' ∈ ~$\mathbf{P_h}$, {i, j} ⊄ ρ'. (That is, {i, j} is a subpatttern of some patterns belong to $\mathbf{P_h}$ but not a subpattern of some patterns belong to ~$\mathbf{P_h}$.) Moreover, the number of patterns containing {j} in ~$\mathbf{P_h}$ is smaller than that of the patterns containing {i} in ~$\mathbf{P_h}$. The reason is that by setting S_{ij} to –1, the support of

item j will be reduced. Moreover, item j has smaller effect on **~P$_h$** than item i.

2. S_{ij}=1, If $\forall\ \rho \in$ **P$_h$**, $\{i, j\} \not\subset \rho$ and $\exists\ \rho' \in$ **~P$_h$**, such that $\{i, j\} \subseteq \rho'$. (That is, {i, j} is not a subpatttern of some patterns belong to **P$_h$** but is a subpattern of some patterns belong to **~P$_h$**)
3. S_{ij} = 0, otherwise.

Non-Hidden-First Algorithm

Input: P$_h$, ~P$_h$, D, S

Output: D'

Step 1: Set the values of the entries in S according to the rules.

Step 2: (matrices multiplication)

For every transaction i in **D** do

    For j=1 to number of items do

        If (D_{ij}=0) D'_{ij}=0;

        Else {

$$\text{Temp} = \sum_{k=1}^{number\ of\ items} D_{ik} * S_{kj}$$

          If (Temp $\geq$1) D'_{ij}=1;

          Else D'_{ij}=0; }

    end-for

end-for

Refer to Figure 5, minimum support is 30%, **P$_h$**: {{4,5}, {1,2,5}}, **~P$_h$**: {{2,4}, {1,3,5}}. After sanitization process, **P$_h$**: {ϕ}, **~P$_h$**: {{2,4}, {1,3,5}}.

$$
\begin{array}{c}
 \\ t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1
\end{pmatrix}
\end{array}
\ X\
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{pmatrix}
1 & 0 & 1 & 0 & 0 \\
-1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & -1 & 1 & -1 & 1
\end{pmatrix}
\end{array}
=
\begin{array}{c}
 \\ t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1
\end{pmatrix}
\end{array}
$$

Figure 5. Illustration of NHF

However, the sanitized database produced by NHF algorithm may contains sensitive patterns. That is, not all the sensitive patterns can be hidden successfully by applying NHF algorithm.

3.3 HPCME Algorithm

The main idea of HPCME algorithm (Hiding sensitive Patterns Completely with Minimum side Effect on non-sensitive patterns), is to combine the advantages in HF and NHF. All sensitive patterns will be hidden with minimal side effect on non-sensitive patterns. To achieve this goal, the idea of *restoration probability* (p_r) is introduced to ensure hiding total sensitive patterns successfully.

Restoration probability

Based on HF algorithm, NHF algorithm set proper entries in **S** to 1 to avoid canceling non-sensitive patterns accidentally. However, some sensitive patterns may be hidden unsuccessfully**.** Therefore, a new factor restoration probability ($0 \le p_r \le 1$) is introduced to decide whether the value of D'_{tj} would follow the multiplication result when the multiplication result is 1 and there exist a S_{kj} = -1 (1≤k≤number of items), (If such entry exists in S, it means that {k, j} is a subpattern of some patterns containing by $\mathbf{P_h}$ and the support of j should be reduced to decrease the support of {k, j}).

When p_r=1 and p_r=0, HPCME algorithm works like NHF and HF algorithm, respectively. A higher value of p_r will let HPCME algorithm tend to preserve non-sensitive patterns, and vice versa. Because our goal is to hide all the sensitive patterns with minimum side effect on non-sensitive patterns, p_r is set to a small value in HCPME algorithm. Moreover, the entries in **S** are set according to the rules defined in NHF algorithm.

HPCME Algorithm

Input: $\mathbf{P_h}$, ~ $\mathbf{P_h}$, D, S, p_r

Output: D'

Step 1: Set the values of the entries in S according to the rules.

Step 2: (matrices multiplication)

For every transaction i in **D** do

For j=1 to number of items do

If (D_{ij}=0) D'_{ij}=0;

Else {

Temp = $\sum_{k=1}^{number\ of\ items} D_{ik} * S_{kj}$

If (Temp ≤ 0) D'_{ij}=0

Else {

if (∃ S_{kj}= -1, 1≤k≤numebr of items) {

D'_{ij}=1 with probability p_r
D'_{ij}= 0 with probability 1-p_r }
else D'_{ij}=1 } }
end-for
end-for

4.Performance Evaluation

4.1The Metrics for Quantifying Performance

As mentioned in section 2, there are three potential errors after the sanitization process. Therefore, three metrics are introduced to evaluate the effectiveness of our algorithms.

Error 1 : some sensitive patterns can still be discovered after sanitization process. The hiding accuracy is measured by

$$Accuracy = \frac{number\ of\ patterns\ in\ \mathbf{P_h}\ which\ are\ hidden\ successfully}{number\ of\ patterns\ in\ \mathbf{P_h}}$$

A sensitive pattern p_s is said to be hidden successfully if there is not exist a pattern p, such that p can be discovered from **D’** where p is a subpattern of p_s and p is not a subpattern of any non-sensitive pattern.

Error 2 : some non-sensitive patterns are hidden after sanitization process. The hiding wrongness is measured by

$$Wrongness = \frac{number\ of\ patterns\ in\ \sim \mathbf{P_h}\ which\ are\ disappeared\ after\ the\ sanitization\ process}{number\ of\ patterns\ in\ \sim \mathbf{P_h}}$$

Error 3 : some artificial patterns are generated after the sanitization process.

$$New\ pattern = \frac{number\ of\ new\ patterns\ can\ be\ discovered\ in\ \mathbf{D'}}{number\ of\ patterns\ in\ \{\mathbf{P} \cup \sim \mathbf{P_h}\}}$$

Moreover, overlap rate is defined as follows for evaluating our approach.

$$overlap\ rate = \frac{|\,item(\mathbf{P_h}) \cap item(\sim \mathbf{P_h})\,|}{|\,item(\mathbf{P_h} \cup \sim \mathbf{P_h})\,|}$$

Where item(P) denotes the set of items contained by P and |X| denotes the cardinality of set X.

4.2 Experiment Results

The test dataset is generated by the IBM synthetic data generator. The test dataset contains 200 different items, with 100K transactions. Moreover, p_r is set to 0.35 in our experiments.

Figure 6 shows the accuracy of algorithms HF, HPCME and NHF. As shown in the result, HF and HPCME algorithm approach at 100% accuracy no matter what the values of overlap rate. In other words, HF and HPCME can hide all the sensitive patterns. NHF works like HF and HCME when the overlap rate is low. However, as overlap rate increases, its accuracy decreases.

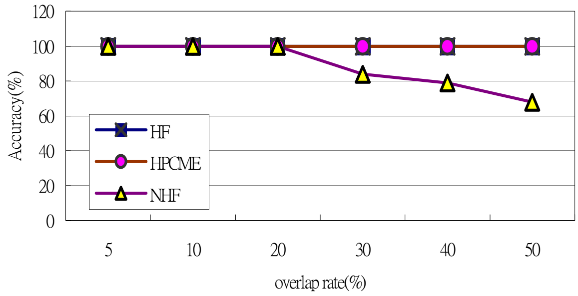

Figure 6 Effect of overlapped rate on Accuracy

Figure 7 shows the wrongness of algorithms for HF, HPCME and NHF. NHF performs much better than HF and HPCME. The more the sensitive patterns need to be hidden, the more the entries in CM are set to –1. As a result, non-sensitive patterns are missed easily. However, HPCME performs better than HF.

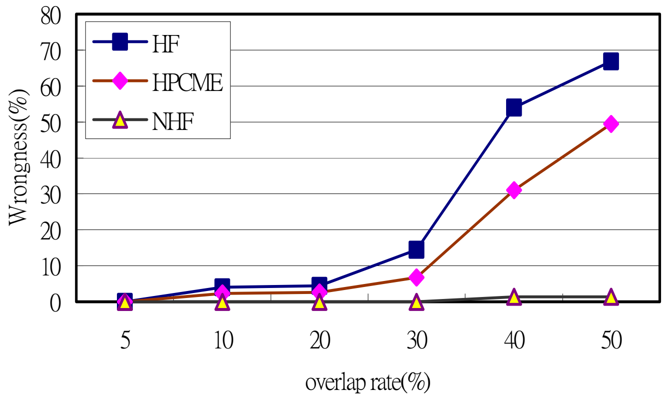

Figure 7 Effect of overlapped rate on Wrongness

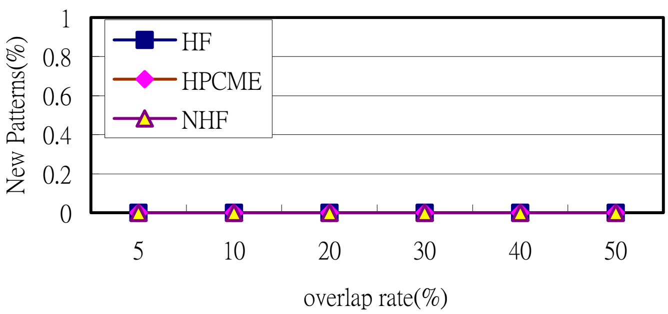

Figure 8 Effect of overlapped rate on New Patterns

Figure 8 shows the new patterns generated by algorithms HF, HPCME and NHF.

There is no new pattern generated by three algorithms because we redefine the multiplication of matrices (IF D_{ij}=0, D'_{ij} is set to 0). In other words, the support for any items will not be increased after the sanitization process.

5. Conclusion and Future Works

In this paper, a new framework is presented for enhancing privacy in mining frequent patterns. The idea of sanitization matrix is introduced. By setting the entries in the sanitization matrix to appropriate values, and multiplying original DB and sanitization matrix, a sanitized database is gotten.

According to different settings in sanitization matrix, we bring up three sanitization algorithms for hiding sensitive patterns successfully (HF algorithm, HPCME Algorithm) or for no legitimate pattern missing (NHF algorithm).

In the future, a new optimal algorithm that minimizes the impact in the sanitized database will be considered.

Reference

[ABEIV99] M. J. Atallah, E. Bertino , A . Elmagarmid , M. Ibrahim and V. S. Verykios. “Disclosure Limitation of Sensitive Rules”. Proceeding of IEEE Knowledge and Data Engineering Exchange Workshop, November 1999.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 487–499, Santiago, Chile, September 1994.

[AS00] R.Agrawal and R.Srikant “Privacy Preserving Data Mining ”, Proceeding of ACM SIGMOD Conference on Management of Data, May 2000.

[ASEG02] R. Agrawal, R. Srikant, A. Evfimievski and J. Gehrke “Privacy Preserving of Association Rules ”, Proceeding of 8th ACM SIGMOD Conference on Knowledge Discovery and Data Mining (KDD), July 2002.

[EGS03] A. Evfimievski, J. Gehrke and R. Srikant “Limiting Privacy Breaches in Privacy Preserving Data Mining”, Proceeding of ACM PODS, 2003.

[RH02] Shariq J. Rizvi and Jayant R. Haritsa , “Maintaining Data Privacy in Association Rule Mining”, Proceedings of the 28th VLDB Conference,2002.

[SO02] Stanley R. M. Oliveira , Osmar R. Zaiane “ Privacy Preserving Frequent Itemset Mining” , IEEE International Conference on Data Mining Workshop on Privacy , Security ,and Data Mining , 2002.

[SVC01] Y. Saygin, V. Verkios and C. Clifton, “Using Unknown to Prevent Discovery of Association Rules”, ACM SIGMOD Records, Vol.30, no.4, 2001.