Abstract - Privacy-preserving data mining [22], is a novel research direction in data mining and statistical databases, where data mining algorithms are analyzed for the side effects they incur in data privacy. For example, through data mining, one is able to infer sensitive information, including personal information or even patterns, from non-sensitive information or unclassified data. There have been two types of privacy concerning data mining. The first type of privacy is that the data is altered so that the mining result will preserve certain privacy. The second type of privacy is that the data is manipulated so that the mining result is not affected or minimally affected.

Given specific rules to be hidden, many data altering techniques for hiding association, classification and clustering rules have been proposed. However, to specify hidden rules, entire data mining process needs to be executed. For some applications, we are only interested in hiding certain sensitive predicative rules that contain given items. In this work, we assume that only sensitive items are given and propose two algorithms, ISL (Increase Support of LHS) and DSR (Decrease Support of RHS), to modify data in database so that sensitive predicative rules containing specified items on the left hand side of rule cannot be inferred through association rule mining. Examples illustrating the proposed algorithms are given. The characteristics of the algorithms are analyzed. The efficiency of the proposed approach is further compared with Verykios etc approach. It is shown that our approach required less number of databases scanning and prune more number of hidden rules. However, our approach must hide all rules containing the hidden items on the left hand side, where Verykios etc approach can hide any specific rule.

Keywords: privacy preserving data mining, predictive association rule.

1 Introduction

The concept of privacy preserving data mining has been recently proposed in response to the concerns of preserving personal information from data mining algorithms [3,4,5,15,16,22]. There have been two types of privacy concerning data mining. The first type of privacy is that the data is altered so that the mining result will preserve certain privacy. Many techniques have been proposed for this type of output privacy [1,6,7,8,9,17,18,20,23]. For example, perturbation, blocking, aggregation or merging, swapping, and sampling are some alternation methods that have recently been proposed. The second type of privacy is that the data is manipulated so that the mining result is not affected or minimally affected [10,11,12,13,21]. For example, the cryptography-based techniques like secure multiparty computation allow users access to only a subset of data while global data mining results can still be discovered. The reconstruction-based technique where the original distribution of the data can be reconstructed from the randomized data is another method for this type of input privacy.

Given specific rules to be hidden, many data altering techniques for hiding association, classification and clustering rules have been proposed. However, to specify hidden rules, entire data mining process needs to be executed. For some applications, we are only interested in hiding certain sensitive predicative rules that contain given items. In this work, we assume that only sensitive items are given and propose two algorithms to modify data in database so that sensitive predicative rules containing specified items on the left hand side of rule cannot be inferred through association rule mining. The proposed algorithms are based on modifying or perturbing the database transactions so that the confidence of the association rules can be reduced. Examples demonstrating the proposed algorithms are shown. The characteristics of the proposed algorithms are analyzed. The efficiency of the proposed approach is further compared with Verykios etc approach. It is shown that our approach required less number of databases scanning and prune more number of hidden rules. However, our approach must hide all rules containing the hidden items on the left hand side, where Verykios etc approach can hide any specific rule.

The rest of the paper is organized as follows. Section 2 presents the statement of the problem and the notation used in the paper. Section 3 presents the proposed algorithms for hiding sensitive predictive association rules that contain the specified items. Section 4 shows some examples of the proposed algorithms. Section 5 analyses the characteristics of proposed algorithms and further compare with Verykios etc approach. Concluding remarks and future works are described in section 6.
2 Problem Statement

2.1 Predicative Association Rules

The problem of mining association rules was introduced in [2]. Let \( I = \{i_1, i_2, \ldots, i_m\} \) be a set of literals, called items. Given a set of transactions \( D \), where each transaction \( T \) 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 \( X \) and \( Y \) are called respectively the body (left hand side) and head (right hand side) of the rule. An example of such a rule is that 90% of customers buy hamburgers also buy Coke. The 90% here is called the confidence of the rule, which means that 90% of transaction that contains \( X \) also contains \( Y \). The confidence is calculated as \( \frac{|X \cup Y|}{|X|} \). The support of the rule is the percentage of transactions that contain both \( X \) and \( Y \), which is calculated as \( \frac{|X \cup Y|}{N} \), where \( N \) is the number of transactions in \( D \). In other words, the confidence of a rule measures the degree of the correlation between itemsets, while the support of a rule measures the significance of the correlation between itemsets. 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 in Table 1, a minimum support of 33% and a minimum confidence of 70%, nine association rules can be found as follows: \( B \Rightarrow A \) (66%, 100%), \( C \Rightarrow A \) (66%, 100%), \( B \Rightarrow C \) (50%, 75%), \( C \Rightarrow B \) (50%, 75%), \( A \Rightarrow B \) (50%, 75%), \( B \Rightarrow AC \) (50%, 75%).

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_1</td>
<td>A BC</td>
</tr>
<tr>
<td>T_2</td>
<td>A BC</td>
</tr>
<tr>
<td>T_3</td>
<td>A BC</td>
</tr>
<tr>
<td>T_4</td>
<td>AB</td>
</tr>
<tr>
<td>T_5</td>
<td>A</td>
</tr>
<tr>
<td>T_6</td>
<td>AC</td>
</tr>
</tbody>
</table>

However, mining association rules usually generates a large number of rules, most of which are unnecessary for the purpose of prediction. For example, given itemset for prediction \( P = \{C\} \), the rule set that contains only two rules \( C \Rightarrow A \) (66%, 100%), \( C \Rightarrow B \) (50%, 75%), will generate the same predicted itemset \( Q = \{A, B\} \) as the nine association rules found from Table 1. A predicative association rule set (or informative rule set) [14] can be informally defined as the smallest rule set that makes the same prediction as the association rule set by confidence priority.

2.2 Problem Description

The objective of data mining is to extract hidden or potentially unknown 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 [1,4-12,16]. In this work, we assume that only sensitive items are given and propose two algorithms to modify data in database so that sensitive predictive association rules cannot be inferred through association rule mining. More specifically, given a transaction database \( D \), a minimum support, a minimum confidence and a set of sensitive items \( X \), the objective is to modify the database \( D \) such that no predictive association rules containing \( X \) on the left hand side will be discovered.

As an example, for a given database in Table 1, a minimum support of 33%, a minimum confidence of 70%, and a hidden item \( X = \{C\} \), if transaction \( T \) is modified as \( AC \), then the following rules that contain item \( C \) on the left hand side will be hidden: \( C \Rightarrow B \) (50%, 60%), \( AC \Rightarrow B \) (50%, 60%), \( C \Rightarrow AB \) (50%, 60%).

The following notation will be used in the paper. Each database transaction has three elements: \( T = \{TID, list\_of\_elements, size\} \). The TID is the unique identifier of the transaction \( T \) and list\_of\_elements is a list of all items in the database. However, each element has value 1 if the corresponding item is supported by the transaction and 0 otherwise. Size means the number of elements in the list\_of\_elements having value 1. For example, if \( I = \{A, B, C\} \), a transaction that has the items \( \{A, C\} \) will be represented as \( t = <T1,101,2> \). In addition, a transaction \( t \) supports an itemset \( I \) when the elements of \( t.list\_of\_elements \) corresponding to items of \( I \) are all set to 1. A transaction \( t \) partially supports an itemset \( I \) when the elements are not all set to 1. For example, if \( I = \{A, B, C\} = [111] \), \( p = <T1,[111],3> \), and \( q = <T2,[001],1> \), then we would say that \( p \) supports \( I \) and \( q \) partially supports \( I \).

3 Proposed Algorithms

In order to hide an association rule, we can either decrease its support or its confidence to be smaller than pre-specified minimum support and minimum confidence. To decrease the confidence of a rule, we can either (1) increase the support of \( X \), i.e., the left hand side of the rule, but not support of \( X \cup Y \); or (2) decrease the support of the itemset \( X \cup Y \). For the second case, if we only decrease the support of \( Y \), the right hand side of the rule, it would reduce the confidence faster than simply reducing the support of \( X \cup Y \). To decrease support of an item, we will modify one item at a time in a selected transaction by changing from 1
Based on these two strategies, we propose two data-mining algorithms for hiding sensitive predictive association rules, namely Increase Support of LHS (ISL) and Decrease Support of RHS (DSR). The first algorithm tries to increase the support of left hand side of the rule. The second algorithm tries to decrease the support of the right hand side of the rule. The details of the two algorithms are described as follow.

Algorithm ISL

Input: (1) a source database \( D \),
(2) a \( \min_{\text{support}} \),
(3) a \( \min_{\text{confidence}} \),
(4) a set of hidden items \( X \)

Output: a transformed database \( D' \), where rules containing \( X \) on LHS will be hidden

1. Find large 1-item sets from \( D \);
2. For each hidden item \( x \in X \)
   3. If \( x \) is not a large 1-item set, then \( X := X - \{x\} \);
4. If \( H \) is empty, then \( \text{EXIT}; \) // no AR contains \( X \) in LHS
5. Find large 2-itemsets from \( D \);
6. For each \( x \in X \) {
   7. For each large 2-itemset containing \( x \) {
      8. Compute confidence of rule \( U \), where \( U \) is a rule like \( x \rightarrow h \);
      9. If \( \text{confidence}(U) < \min_{\text{conf}} \), then
         10. Go to next large 2-itemset;
      11. Else {
         12. Find \( T_x = \{ t \in D \mid t \text{ does not support } U \} \);
         13. Sort \( T_x \) in ascending order by the number of items;
         14. While \( \{ \text{confidence}(U) \geq \min_{\text{conf}} \text{ and } T_x \text{ is not empty} \} \) {
            15. Choose the first transaction \( t \) from \( T_x \);
            16. Modify \( t \) so that \( h \) is not supported;
            17. Compute support and confidence of \( U \);
            18. Remove and save the first transaction \( t \) from \( T_x \);
            19. } // end While
         20. } // end if
      21. If \( T_x \) is empty, then {
         22. Can not hide \( x \rightarrow h \);
      23. Restore \( D \);
      24. Go to next large 2-itemset;
      25. } // end if \( T_x \) is empty
   26. } // end of for each large 2-itemset
   27. Remove \( x \) from \( X \);
   28. } // end of for each \( x \)
29. Output updated \( D \), as the transformed \( D' \);

Algorithm DSR

Input: (1) a source database \( D \),
(2) a \( \min_{\text{support}} \),
(3) a \( \min_{\text{confidence}} \),
(4) a set of hidden items \( X \)

Output: a transformed database \( D' \), where rules containing \( X \) on LHS will be hidden

1. Find large 1-item sets from \( D \);
2. For each hidden item \( x \in X \)
   3. If \( x \) is not a large 1-item set, then \( X := X - \{x\} \);
4. If \( H \) is empty, then \( \text{EXIT}; \) // no AR contains \( X \) in LHS
5. Find large 2-itemsets from \( D \);
6. For each \( x \in X \) {
   7. For each large 2-itemset containing \( x \) {
      8. Compute confidence of rule \( U \), where \( U \) is a rule like \( x \rightarrow h \);
      9. If \( \text{confidence}(U) < \min_{\text{conf}} \), then
         10. Go to next large 2-itemset;
      11. Else {
         12. Find \( T_x = \{ t \in D \mid t \text{ fully support } U \} \);
         13. Sort \( T_x \) in ascending order by the number of items;
         14. While \( \{ \text{confidence}(U) \geq \min_{\text{conf}} \text{ and } T_x \text{ is not empty} \} \) {
            15. Choose the first transaction \( t \) from \( T_x \);
            16. Modify \( t \) so that \( h \) is supported;
            17. Compute support and confidence of \( U \);
            18. Remove and save the first transaction \( t \) from \( T_x \);
            19. } // end While
         20. } // end if
      21. If \( T_x \) is empty, then {
         22. Can not hide \( x \rightarrow h \);
      23. Restore \( D \);
      24. Go to next large 2-itemset;
      25. } // end if \( T_x \) is empty
   26. } // end of for each large 2-itemset
   27. Remove \( x \) from \( X \);
   28. } // end of for each \( x \)
29. Output updated \( D \), as the transformed \( D' \);

4 Examples

This section shows four examples for demonstrating the two proposed algorithms in hiding sensitive predictive association rules in the association rule mining.

For a given database in Table 1, a minimum support of 33% and a minimum confidence of 70%, the first two examples hide the sensitive rules using the ISL algorithm. The difference of the two examples is that the order of hiding item is different. The first example hides item C and then item B. The second example hides item B and then item C. The result is given in section 4.1.

The third and fourth examples hide the sensitive predictive association rules using DSR algorithm. The difference is also the order of items to be hidden. The result is given in section 4.2.
### Table 2: Database $D$ using the specified notation

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>111</td>
<td>3</td>
</tr>
<tr>
<td>T2</td>
<td>111</td>
<td>3</td>
</tr>
<tr>
<td>T3</td>
<td>111</td>
<td>3</td>
</tr>
<tr>
<td>T4</td>
<td>110</td>
<td>2</td>
</tr>
<tr>
<td>T5</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>T6</td>
<td>101</td>
<td>2</td>
</tr>
</tbody>
</table>

### 4.1 Examples Running ISL Algorithm

**Example 1**  Assuming that the min_supp = 33% and min_conf = 70%, the result of hiding item $C$ and then item $B$ using ISL algorithm is as follows. To hide item $C$, the rule $C \Rightarrow B$ (50%, 75%) will be hidden if transaction $T_5$ is modified from 100 to 110 using ISL (Increase Support of LHS). The new database $D_1$ is shown in Table 3. The rule $C \Rightarrow B$ will have support = 50% and confidence = 60%. However, rules $C \Rightarrow A$, $B \Rightarrow A$, $B \Rightarrow C$ cannot be hidden by ISL algorithm.

**Example 2**  As in example 1, reversing the order of hiding items, the result of hiding item $B$ and then item $C$ using ISL algorithm is as follows. To hide item $B$, the rule $B \Rightarrow C$ (50%, 75%) will be hidden if transaction $T_6$ is modified from 101 to 111 using ISL. The new database $D_2$ is shown in Table 4. The rule $B \Rightarrow C$ will have support = 50% and confidence = 60%. However, rules $B \Rightarrow A$, $C \Rightarrow A$, $C \Rightarrow B$ cannot be hidden by ISL algorithm.

One observation we can make is that different sequences of hiding items will result in different transformed databases, i.e., $D_1$ and $D_2$ for ISL algorithm.

**Table 3**  Databases before and after hiding item $C$ and item $B$ using ISL

<table>
<thead>
<tr>
<th>TID</th>
<th>$D$</th>
<th>$D_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T2</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T3</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T4</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>T5</td>
<td>100</td>
<td>101</td>
</tr>
<tr>
<td>T6</td>
<td>101</td>
<td>101</td>
</tr>
</tbody>
</table>

### 4.2 Examples Running DSR Algorithm

**Example 3**  Assuming that the min_supp=33% and min_conf=70%, the result of hiding item $C$ and then item $B$ using DSR algorithm is as follows. To hide item $C$, the rules $C \Rightarrow A$ (60%, 100%), $C \Rightarrow B$ (50%, 75%), $B \Rightarrow C$ (50%, 75%), $B \Rightarrow A$ (60%, 100%) will be hidden if transaction $T_6$ is modified from 101 to 001, $T_1$ is modified from 111 to 011, $T_1$ is modified from 011 to 010, and $T_6$ is modified from 110 to 101, using DSR. The new database $D_3$ is shown in Table 5.

**Example 4**  As in example 3, reversing the order of hiding items, the result of hiding item $B$ and then item $C$ using DSR algorithm is as follows. To hide item $B$, the rules $B \Rightarrow A$ (60%, 100%), $B \Rightarrow C$ (50%, 75%), $C \Rightarrow A$ (60%, 100%), $C \Rightarrow B$ (50%, 75%) will be hidden if transaction $T_4$ is modified from 110 to 010, $T_1$ is modified from 111 to 011, $T_1$ is modified from 011 to 010, $T_6$ is modified from 101 to 001, using DSR. The new database $D_4$ is shown in Table 6.

One observation is that different sequences of hiding items will result in different transformed databases, i.e., $D_3$ and $D_4$ for DSR algorithm.

**Table 5**  Databases before and after hiding item $C$ and item $B$ using DSR

<table>
<thead>
<tr>
<th>TID</th>
<th>$D$</th>
<th>$D_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>111</td>
<td>001</td>
</tr>
<tr>
<td>T2</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T3</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T4</td>
<td>110</td>
<td>010</td>
</tr>
<tr>
<td>T5</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>T6</td>
<td>101</td>
<td>001</td>
</tr>
</tbody>
</table>

**Table 6**  Databases before and after hiding item $B$ and item $C$ using DSR

<table>
<thead>
<tr>
<th>TID</th>
<th>$D$</th>
<th>$D_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>111</td>
<td>010</td>
</tr>
<tr>
<td>T2</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T3</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T4</td>
<td>110</td>
<td>010</td>
</tr>
<tr>
<td>T5</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>T6</td>
<td>101</td>
<td>001</td>
</tr>
</tbody>
</table>

### 5 Analysis

This section analyzes some of the characteristics of the proposed algorithms and compares with the algorithms proposed in Verykios et al’s [9,23]. The first characteristic we observe is the item ordering effect. The transformed databases are different under different ordering of hiding items, even though the same set of sensitive items is specified. The second characteristic we observe is the algorithm effect. The transformed databases will be different under different algorithm. These characteristics

are demonstrated in the four examples in section 4 and summarized in Table 7. Databases $D_1$ and $D_2$ are resulting databases using ISL algorithm and $D_3$ and $D_4$ are resulting databases using DSR algorithm.

Table 7 Databases before and after hiding items $B$ and $C$ using ISL and DSR

<table>
<thead>
<tr>
<th>TID</th>
<th>D</th>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_3$</th>
<th>$D_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>001</td>
<td>010</td>
</tr>
<tr>
<td>T2</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T3</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>T4</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>T5</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>T6</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>001</td>
<td>001</td>
</tr>
</tbody>
</table>

The third characteristic we analyze is the efficiency of the proposed algorithm compared with the Verykios etc algorithms. Even though it is the hidden rules, instead of hidden items, that are specified in [9,23], we compare the number of database scanning and the number of rules pruned between the two approaches. Table 8 summarizes the results.

For ISL algorithm, the number of database scanning comes from the calculation of large one itemsets, large two itemsets, and transactions $T_i$. The rules pruned are $AC \Rightarrow B$ and $C \Rightarrow AB$. For Verykios etc’s 1a algorithm, the number of database scanning comes from the calculation of large one itemsets, large two itemsets, large three itemsets, and partial support transactions $T$. No rules are pruned in the Verykios etc’s algorithm. It can be seen that the ISL algorithm requires less database scanning and prune more number of association rules. Similar results are obtained for comparing DSR algorithm and Verykios etc’s 1b algorithm.

Table 8 Database scans and rules pruned in hiding item $C$ using ISL

<table>
<thead>
<tr>
<th></th>
<th>DB Scans</th>
<th>Rules Pruned</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISL</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Dasseni’s 1a</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

One of the reasons that Verykios etc’s approach does not prune rules is that hidden rules are given in advance and the algorithms try to hide every single rule without checking to see if rules can be pruned after some transactions have been changed.

However, our approach needs to hide all rules containing hidden items on the left hand side. But Verykios etc’s approach can hide some of the rules containing hidden item on the left hand side. For example, for hidden item $C$, Verykios etc’s approach can hide $C \Rightarrow A$, but show $C \Rightarrow B$, whereas our approach must hide both $C \Rightarrow A$ and $C \Rightarrow B$.

The fourth characteristic we analyze is efficiency comparison of the ISL and DSR algorithms. One observation we conclude from the examples in section four is that DSR algorithm seems to be more effective when the support count of the hidden item is large. This is due to when support of right hand side of the rule is large; increase support of left hand side usually does not reduce the confidence of the rule. However, decrease support of right hand side usually decreases the confidence of the rule.

6 Conclusions

In this work, we have studied the database privacy problems caused by data mining technology and proposed two naïve algorithms for hiding sensitive predictive association rules in association rules mining. The proposed algorithms are based on modifying the database transactions so that the confidence of the association rules can be reduced. Examples demonstrating the proposed algorithms are shown. The item ordering and algorithm ordering characteristics of the proposed algorithms are analyzed. The efficiency of the proposed approach is further compared with Verykios etc’s [9,23]. It was shown that our approach required less number of database scanning and prune more number of hidden rules. However, our approach must hide all rules containing the hidden items on the left hand side, where Verykios etc’s approach can hide any specified rule. Currently we are performing numerical simulation on the time effects and side effects (the number of lost rules and new rules due to alternation of the database) and will be included in this work. In the future, we will examine and compare with other alternation techniques for hiding predictive association rules based on current approach

References


[4] Ljiljana Brankovic and Vladimir Estivill-Castro, “Privacy Issues in Knowledge Discovery and Data Mining”,


