Data Reduction Approach for Sensitive Associative Classification Rule Hiding

Juggapong Natwichai[1,4] **Xingzhi Sun**[2] **Xue Li**[3]

[1] Computer Engineering Department, Faculty of Engineering,
Chiang Mai University, Thailand,
Email: juggapong@eng.cmu.ac.th

[2] IBM China Research Laboratory,
Beijing, China,
Email: sunxingz@cn.ibm.com

[3] School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, Australia,
Email: xueli@itee.uq.edu.au

[4] Biomedical Engineering Center,
Chiang Mai University, Thailand

Abstract

When a business unit shares data with another unit, there could be some sensitive patterns which should not be disclosed. In order to remove or "hide" a sensitive pattern in data sharing scenario, the data set needs to be modified such that the sensitive pattern becomes uninteresting according to the pre-specified "interestingness" threshold(s). However, data quality of the given data set should also be maintained, otherwise, the sharing will be meaningless. Existing data modification algorithms usually use data perturbation approach, i.e. changing some data values in a given data set from an original value to another value. Though, it could hide sensitive patterns and maintain data quality, such the approach could not be applied in a situation where real data are required. In this paper, we explore an alternate approach for sensitive pattern hiding problem, data reduction, i.e. removing the whole selected tuples. By data reduction, every tuple in modified data sets is real data without any change. The focused pattern type is associative classification rule. The impact on data quality is denoted as the numbers of false-dropped rules and ghost rules. The experiments are conducted to evaluate the approach and the results have shown that data reduction approach can produce data sets with high data quality, thus it is applicable to the problem.

1 Introduction

Recently, data sharing becomes a common business practice. Data could be exchanged between cooperating organizations, or could be released publicly. With these shared data, useful patterns, or knowledge, can be discovered by available data mining techniques. Subsequently, the business operations can be improved by such the knowledge. However, data sharing can also causes the privacy issue. First, the released data may contain sensitive private information of individuals, e.g. persons' identifiers or payroll information. To prevent the disclosure of the sensitive data, techniques such as data transformation to conform k-anonymity standard (Sweeney 2002), can be applied. Besides the privacy concern for sensitive data, there exists another form of threat, i.e. the disclosure of sensitive patterns discoverable from data.

In (Fule & Roddick 2004), the authors present a motivating example which sensitive patterns can damage reputation of the individuals in the data. In this example, suppose that a data set is released publicly. A rule **"(PostCode = 5409) ∧ (Age = 18 to 25) ∧ (Gender = Male) → HepBStatus = Yes** is discovered from the data set, suppose that the postcode 5409 referred to an indigenous community or the national parliament. This rule may be considered an offense to the population in the area and should be "hidden" before the data set is released.

In data sharing scenario, in order to hide sensitive patterns, the given data set needs to be modified so that the sensitive pattern becomes uninteresting against the pre-specified "interestingness" thresholds. To address this problem, not only sensitive patterns should be hidden, data modification algorithms should also maintain the characteristics of the given data set, as required by the releasing purpose. In this paper, we refer such the characteristics as data quality. Apparently, failing to preserve data quality means that the data sharing is useless.

Typically, the existing data modification algorithms (Oliveira & Zaïane 2003, Verykios, Elmagarmid, Bertino, Saygin & Dasseni 2004, HajYasien & Estivill-Castro 2006) apply data perturbation approach, i.e. changing some data values in a given data set from an original value to another value. For example, a data value "male" in gender attribute of a tuple could be changed into "female" in order to hide a sensitive pattern. Although such approach could hide sensitive patterns and possibly maintain data quality, it has the following drawbacks. First, some of the data values in a perturbed data set are not "real" values. Further, there is no method to distinguish real data and modified data within the perturbed data sets. This drawback could reduce the creditability of modified data sets. Finally, the perturbation may modify some tuples and cause some uninteresting patterns to become interesting.

In this paper, we propose a data reduction approach to address the problem of hiding sensitive patterns. In this approach, data modifications algo-

rithms will remove the whole selected tuples in order to hide sensitive patterns. Comparing with the data perturbation approach, all data values in a modified data set are real. So, this approach produces credible data sets in detail level. Also, if some uninteresting patterns become interesting by a data reduction, it is often the case that for these patterns, at least one of their interestingness measures have reached the threshold, but some tuples may block the patterns from being interesting against the other interestingness measure(s). This is different from the perturbation approach in which this situation may occur from the artificial perturbed data.

We explore the sensitive pattern hiding problem, in which the focused pattern type is associative classification rule (Li, Han & Pei 2001, Liu, Hsu & Ma 1998). This type of pattern can be discovered based on support and confidence scheme as association rule mining (Agrawal, Imielinski & Swami 1993), but having a designate attribute as class label. For the hidden condition of sensitive rules, as mentioned in (Agrawal et al. 1993), the confidence of a rule is its strength, while the support is its statistical interestingness. A rule is worth consideration if its support is higher than the minimum threshold. Therefore, in this paper, a sensitive rule is hidden successfully if its support is fallen below the pre-specified support threshold. While, the impact on data quality is represented in term of the number of false-dropped rules and ghost rules, both of which are well-known as the measurements for data quality (Verykios et al. 2004). False-dropped rules are non-sensitive rules whose support or confidence falls below the support or confidence threshold by data modification unintentionally. While, ghost rules are artificially generated by data modification. To maintain data quality, data modification algorithms should keep the sum of two numbers as low as possible.

After defining the problem, we conduct experiments to illustrate the applicability of the data reduction approach. For the investigation purpose, we implement an exhaustive search algorithm which hides sensitive rules and meanwhile optimizes data quality. The data quality is investigated in various different situations, i.e. standard data sets with different characteristics, sensitive rules with different supports, and different numbers of sensitive rules to be hidden. Also, the numbers of remaining tuples from the reduction process are investigated.

The contribution of the paper is twofold. First, we introduce and define the problem of hiding sensitive associative classification rules by data reduction. Second, we conduct extensive experiments and the results show that the data reduction based approach work very effectively for our problem.

The organization of this paper is as follows. Related work is reviewed in the next section. The basic notations used in this paper are introduced in Section 3. A data reduction approach is discussed in Section 4. And the experimental results are reported in Section 5. Finally, we provide conclusion and outlooks to future works in Section 6.

2 Related work

The sensitive pattern hiding problem is one of the problems addressed in Privacy Preservation Data Mining (PPDM) research area. In PPDM, the problems to be focused can be categorized into two types, the problem of individual privacy preservation and the threat from sensitive patterns. For individual privacy, every record within a given data set should not be re-identified when data mining algorithms are applied to the data set. There are several works proposed to address the de-identifying problem (Sweeney 2002, Wong, Li, Fu & Wang 2006, Machanavajjhala, Gehrke, Kifer & Venkitasubramaniam 2006)

For the sensitive pattern hiding problem, it was originally introduced in (Atallah, Elmagarmid, Ibrahim, Bertino & Verykios 1999). As discussed in Section 1, not only sensitive patterns should be hidden, but also data quality should be preserved. When an optimal data quality is required, the sensitive pattern hiding problem is proven as an NP-hard problem (Atallah et al. 1999). There are a few approaches to modify data sets for sensitive pattern hiding problem, i.e. data perturbation (Verykios et al. 2004, Oliveira & Zaïane 2003, Sun & Yu 2005), data reconstruction (Evfimievski, Srikant, Agarwal & Gehrke 2004), data reduction (Clifton 2000) and data blocking (Saygin, Verykios & Clifton 2001).

Existing works usually addressed the problem by data perturbation approach in the context of association rules and frequent itemsets. In (Verykios et al. 2004), the authors presented a few heuristic algorithms to modify the data set to hide sensitive association rules. By their proposed algorithms, the selected values in the data set will be perturbed to decrease the support and/or the confident values of the sensitive rules. The rules will be hidden successfully if their support and/or confident values are less than the specific thresholds. The authors proposed to hide sensitive association rules through the two following options, (1) decrease the confidence of the rule, and (2) decrease the support of the rule. In the first option, the authors presented the analysis of the association rules's confidence formulation, that is, $\frac{|X \cup Y| \times 100}{|X|}$ for a rule $|X \rightarrow Y|$. Then, the authors suggested two ways to decrease the confidence; first, perturb the item Y from 1 to 0 in the transactions which partially support the rule $X \rightarrow Y$ to decrease the numerator part of formulation ($|X \cup Y|$). This will fix the value in denumerator part of the formulation ($|X|$). The other way of hiding is to increase the value of the denumerator of the formulation which will make the confidence value decrease. It can be done by perturbing the item X from 0 to 1 in the partially supported transactions of the rule. The second option is achieved by perturbing the item X or either Y from 1 to 0 to decrease the support of the rule $X \rightarrow Y$. The authors also presented the experiment results of these proposed ways.

Oliveira and Zaïane proposed three heuristic algorithms to hide sensitive frequent itemsets, i.e. Minimal Frequency Item, the Maximal Frequency Item, and the Item Grouping algorithms (Oliveira & Zaïane 2002, Oliveira & Zaïane 2003). The algorithms have four steps as follows. First, the algorithms identify the transactions which support the sensitive patterns. Second, the victim items in the identified transactions are selected for perturbation by different criteria. In the Minimal Frequency Item algorithm, items which have minimal support values are selected to be removed. While the Maximal Frequency Item algorithm removes the item that has maximal support value. In the last algorithm, the Item Grouping algorithm, tries to select an item that is common among sensitive frequent itemsets, then, removing the item can help hide many sensitive frequent itemsets at once. Third, the number of transactions to be perturbed is determined by the disclosure threshold. The last step is the actual perturbation. For each sensitive pattern, it begins with the sorting of the supporting transactions of the pattern by the degree of conflict, i.e. the number of sensitive rules which a transaction supports. Then, it perturbs a numbers of transactions from the third step by changing the item value from 1 to 0 in the victim items.

There are a few works proposed to solve the problem based on the concept of the border of itemsets. In (Sun & Yu 2005), a border-based algorithm to hide the sensitive frequent itemsets is proposed. the algorithm is proposed to hide the lower border of the sensitive itemsets, instead of hiding every sensitive itemset. In (Moustakides & Verykios 2006), the MinMax algorithm is proposed. In this work, given many sensitive itemsets, the authors suggest hiding the sensitive itemsets with minimum support first, because they are the closest to the borders. Then, among the sensitive minimum support itemsets, the highest (or maximum) support itemsets will be selected. From such the highest support itemsets, the victim items to be modified can be selected.

Typically, when a set of sensitive patterns is given, sensitive pattern-hiding algorithms will consider hiding the rules on a one-by-one basis. In (HajYasien & Estivill-Castro 2006), the authors proposed two techniques for sensitive item selection which considered many sensitive itemsets at the same time. The first technique considers the sensitive item's numbers of occurrence among sensitive itemsets, that is, it selects for modification the items with the highest occurrence. The second technique considers the cardinalities of the sensitive itemsets: it selects for modification those itemsets with the smallest cardinality. The experiments showed that the first technique has less impact on the data quality in term of false-dropped itemsets.

In (Wu, Chiang & Chen 2007), the authors presented a different view to the problem of sensitive association rule hiding. Instead of hiding the sensitive association rules and minimizing the impact on data quality (ghost rules and false-dropped rules), the authors suggested that the impact can damage the applicability of the modified data set. Therefore, an algorithm to modify the data set such that there is no side effect, and minimize the number of disclosed sensitive rules is proposed.

Another work related to sensitive rule hiding is the inference problem with regard to classification mining in (Wang, Fung & Yu 2005). Instead of sensitive rule hiding, the authors address a problem of blocking inference channel in the form $< IC \rightarrow \pi, h >$, where IC is a set of attributes, π is a class label, and h is a confidence threshold, for example $< \{Poscode, Age, Gender\} \rightarrow HepBStatus = Yes, 75\% >$. The authors also presented an algorithm to block inference channel by modifying the data in a top-down basis, i.e. a sensitive attribute value will firstly be transformed into the most general value, then it will be transformed into a more specific value when the algorithm proceeds further. This work can be considered as a more general problem than the works which address sensitive pattern hiding.

Instead of the sensitive pattern hiding problem in the data sharing scenario, the problem in rule-sharing scenario is presented in (Oliveira, Zaïane & Saygin 2004). The authors proposed the Downright Sanitizing Algorithm (DSA) which can be used to decide which other frequent itemsets (aside from the sensitive frequent itemsets) must be also hidden. These other frequent itemsets can be (1) subsets of the sensitive frequent itemsets, (2) supersets of the sensitive frequent itemsets.

3 Basic Notation

In this section, we introduce the basic notation required for our consideration: Data Set and Classification.

Definition 1 (Data Set) *Let a data set D be a collection of tuples, $D = \{d^1, d^2, \ldots, d^n\}$, and $I = \{1, \ldots, n\}$ be a set of identifiers for elements of D. Tuples in a table is not necessary to be unique.*

The data set D is defined on a schema $\boldsymbol{A} = \{A_1, A_2, \ldots, A_k\}$, and $J = \{1, \ldots, k\}$ be a set of identifiers for elements of $\boldsymbol{A}$.

For each $j \in J$, domain of A_j, denoted as $dom(A_j) \subseteq \mathcal{N}$, where $\mathcal{N}$ is the set of natural number.

For each $i \in I$, $d^i(\boldsymbol{A}) = (d^i(A_1), d^i(A_2), \ldots, d^i(A_k))$, denoted as $(d^i_1, d^i_2, \ldots, d^i_k)$.

Let C be a set of class labels, such that $C = \{c_1, c_2, \ldots, c_o\}$, and $M = \{1, \ldots, o\}$ be a set of identifiers for elements of C. For all $m \in M$, $c_m \in \mathcal{N}$, where $\mathcal{N}$ is the set of natural numbers.

The label is just an identifier of a class. A class which is labelled as c_m defines a subset of tuples which is described by data assigned to the class. The class label of a tuple d^i is denoted as $d^i.Class$. The classification problem is to establish a mapping from D to C.

Please note that we are defining the general data set for the traditional associative classification problem. In the data set, we allow duplication (i.e. two data entries are identical in terms of tuple and class label) and conflict (i.e. two data entries can have the same tuple information but with different class label).

Definition 2 (Classification) *A literal p is a pair, consisting of an attribute A_j and a value v in $dom(A_j)$. A tuple d^i will satisfy the literal $p(A_j, v)$ iff $d^i_j = v$.*

Given a data set D, and a set of class labels C, let R be a set of classification rules, such that $R = \{r_1, r_2, \ldots, r_q\}$, and $L = \{1, \ldots, q\}$ be a set of identifiers for elements of R.

For all $l \in L$, $r_l : \bigwedge p \rightarrow c_m$, where p is the literal, and c_m is a class label. The left hand side (LHS) of the rule r_l is the conjunction of the literals, denoted as $r_l.LHS$. The right hand side (RHS) is a class label of the rule r_l, denoted as $r_l.RHS$.

A tuple d^i satisfies the classification rule r_l iff it satisfies all literals in $r_l.LHS$, and has a class label c_m as $r_l.RHS$.

*A tuple d^i which satisfies the classification rule r_l is called supporting tuple of r_l. The **support** of the rule r_l, denoted as $Sup(r_l)$, is the number of supporting tuples of r_l. The **supporting set** $D_l = \{d^i \in D | d^i$ that satisfies $r_l\}$. The **confidence** of rule r_l, denoted as $Conf(r_l)$, is the ratio between $Sup(r_l)$ and the total number of tuples which satisfy all literals in LHS of r_l.*

Typically, a number of classification rules which satisfy minimal support and confidence values can be large (Li et al. 2001). The set of rules should be pruned by removing some "redundant" rules before being applied in the classification for the target data set. So, in the context of sensitive rule hiding problem, we should deal with only unpruned rules. In this paper, we hide and address the quality on the concept of "general rules" as follows.

Definition 3 (General rule) *Given a data set D, a set of classification rules R satisfying minimal support minsup, and minimal confidence minconf. A classification rule $r_l \in R$ is a general rule if there does not exist a classification rule $r'_l \in R$ which $r_l.RHS = r'_l.RHS$ and $r_l.LHS \supset r'_l.LHS$.*

4 Data Reduction

In this section, we present the rule hiding definition, impact on data quality and problem statement.

Firstly, we present here an example data set. It will be used through this paper. Suppose we are dealing with the 3-attributes data set, and two classes as shown in Table 1.

Table 1: An example data set

Tuple ID	A_1	A_2	A_3	C
1	1	0	1	0
2	1	1	1	0
3	1	0	1	0
4	0	1	1	1
5	0	0	0	1
6	0	1	1	1
7	1	1	0	1
8	1	0	0	1
9	0	1	0	1
10	1	1	0	1

With the minimal support and minimal confidence set at 2 and 90% respectively, we can derive a set of general rules from the example data set by an associative classification algorithm as shown in Table 2. We can see that non-generals are not listed, for example, a rule $(A_2 = 0) \wedge (A_3 = 0) \rightarrow 1$ which has support 2 and 100 % confidence is not listed because a rule r_1, $(A_3 = 0) \rightarrow 1$ is more general.

The "hidden" condition for a sensitive rule is defined as follows.

Definition 4 (Hidden Rule) *Given a data set D with a set of class labels C, let R be the set of general classification rules from D satisfying a minimal support threshold $minsup$, and a minimal confidence threshold $minconf$. Let $R_s \subset R$ be a set of sensitive classification rules. A sensitive rule r_s is hidden if its support in D' less than $minsup$.*

To hide a rule, we need to remove its supporting tuples until its support value is below the minimal support threshold.

Example 1 *From our running data set in Table 1 and the set or rules in Table 2, suppose that the data owner want to hide rule r_2:$(A_1 = 0) \rightarrow 1$. To satisfy the hidden rule condition, 3 out of 4 tuples in D_2 must be removed ($D_2 = \{d^4, d^5, d^6, d^9\}$).*

Once the sensitive rules are guaranteed to be hidden by the condition, the data quality should be maximize, in the other words, the impact on the data quality by data reduction should be minimized. The impact is represented in terms of the number of false-dropped rules and the number of ghost rules in the modified data set. The impact is defined as follows.

Definition 5 (False-dropped rules) *A false-dropped rule is a non-sensitive general rule in $R - R_s$ in the original data set D which can not be derived from the modified data set D' by using minimal support $minsup$ and minimal confidence $minconf$.*

When we remove a supporting tuple, for the non-sensitive rules which are supported by the tuple, both their support and confidence are decreased. If the support or confidence value of any non-sensitive rule is less than the threshold, the rule can not be derived in the modified data set and becomes a false-dropped rule.

Example 2 *From our running example, suppose that the data owner wants to hide a sensitive rule r_3:$(A_1 = 1) \wedge (A_3 = 1) \rightarrow 0$, which has supporting tuples $\{d^1, d^2, d^3\}$. If a tuple d^1 (or d^3) is selected, rule r_4:$(A_2 = 0) \wedge (A_3 = 1) \rightarrow 0$ will be lost, because the selected tuple also support a non-sensitive rule r_4. Moreover, the support of r_4 is exactly equal to the minimal support threshold. So, the number of false-dropped rules from removal of d_1 (or d^3) to hide the rule is 1.*

Definition 6 (Ghost Rules) *A ghost rule is a rule which can not be derived from the original data set D by using minimal support $minsup$ and minimal confidence $minconf$, but can be derived from the modified data set D'.*

The impact of ghost rules can be considered as the opposite impact to the false-dropped rules. In a data set, there may exist some classification rules whose support is greater than $minsup$ but confidence below $minconf$. When a supporting tuple of a sensitive rule r_l is removed, it may increase confidence of this type of rule if the tuple satisfies the left-hand-side of the rule, but the rule has different class label. If the increasing confidence of a rule can satisfy the minimal confidence threshold, the rule will become a ghost rule.

Example 3 *Suppose that the data owner wants to hide a sensitive rule r_1:$(A_3 = 0) \rightarrow 1$ in the running example, its supporting tuples which we can remove are $\{d^5, d^7, d^8, d^9, d^{10}\}$. If tuple d^8 is removed, the rules $(A_2 = 0) \rightarrow 0$, $(A_1 = 1) \rightarrow 0$, and $(A_1 = 1) \wedge (A_2 = 0) \rightarrow 0$ can be come ghost rules because d^8 satisfies their literals, but d^8 has different class (these rules satisfy minimal support threshold). However, they are not derived in the first place because the confidence values of these rules are less than minimal confidence threshold. Considering the data set, d^8 removal will cause the confidences of rule $(A_1 = 1) \wedge (A_2 = 0) \rightarrow 0$ to increase and satisfy the minimal confidence threshold. After the removal, such the ghost rule $(A_1 = 1) \wedge (A_2 = 0) \rightarrow 0$ is generated. The number of ghost rules from removal of d_8 to hide the rule r_1 is 1.*

Remember that in the associative classification problem, we only consider the most general interesting classification rules as the mining result. This will lead to some additional circumstances in which false-dropped rule and ghost rules can be generated during the hiding process. First, the false-dropped rules can also be caused by the confidence increase of previously uninteresting rules. For example, suppose that r_0 is an uninteresting rule, and during the hiding process, it becomes interesting due to the increase of its confidence (that is, r_0 is a ghost rule). If r_0 is more general than some interesting non-sensitive rules, these less general rules should be removed from the result set and therefore, become the false-dropped rules. Similarly, if a non-sensitive rule r_1 becomes uninteresting due to the decrease of its confidence (i.e. r_1 is a false-dropped rule), some rules which are less general than r_1 but with the support and confidence above the thresholds could appear in the result set because they are now the most general interesting rules. According to our definition, these rules are ghost rules.

From the above definitions, we formalize sensitive associative classification rule hiding problem by data reduction approach as follows.

Problem 1 *Given a data set D with set of class labels C, let R be the set of associative classification rules from D and for any rule $r \in R$, $Sup(r) > minsup$ and $Conf(r) > minconf$, where $minsup$ and $minconf$ are two given thresholds. In addition, let $R_s \subset R$ be a set of sensitive classification rules. The problem is to transform D into D' by removing some tuples from D such that 1) any rule $r_s \in R_s$ is invalid in D' in terms of the threshold $minsup$ and 2) the*

Table 2: Associative classification rules on the example data set

Rule No.	Content	Support	Confidence
1.	$(A_3 = 0) \rightarrow 1$	5	100%
2.	$(A_1 = 0) \rightarrow 1$	4	100%
3.	$(A_1 = 1) \wedge (A_3 = 1) \rightarrow 0$	3	100%
4.	$(A_2 = 0) \wedge (A_3 = 1) \rightarrow 0$	2	100%

impact, i.e. the summation of the number of false-dropped rules and the number of ghost rules, of removal is minimized.

Note here that the impact is defined as the summation for simplicity. It could be adjusted according to the application. For example, in medical domain, ghost rules could lead to the wrong treatment (Wu et al. 2007), so it should be weighted as the higher impact on data quality, then data modification algorithms will prefer to generate false-dropped rules.

5 Data Reduction Approach Evaluation

In this section, first, we present an exhaustive search algorithm which hides sensitive rules and guarantees optimal data quality. Then, we give the experiment setting for our evaluation. At the end, we present the experiment results with discussions.

5.1 An Exhaustive Algorithm

Figure 1 shows the pseudo code of the exhaustive algorithm 1. This algorithm removes the selected tuples for hiding sensitive rules and meanwhile guarantees the optimal quality for the modified data set. For each sensitive rule r_s, we first compute the set D_s of tuples that support r_s. Next, we create $D\_Removal_s$, which is the set of all possible candidate tuple sets to hide r_s. Precisely, each element in $D\_Removal_s$ is the subset of D_s and has cardinality $Sup(r_s) - minsup + 1$. After finding $D\_Removal_s$ for each sensitive rule, we create our search space $Global\_D\_Removal$, which is the set that enumerates all possible tuple sets for hiding the complete set R_s of sensitive rules. The element of $D\_Removal_s$, denoted as GDR, is formed by selecting an element (one set of tuples that can hide r_s) from each $D\_removal_s$ and then computing the union of them. Finally, we evaluate every candidate solution in the search space. That is, for each GDR, we compute the corresponding modified data set $Intermediate\_D$. Then, we recompute the classification rules for $Intermediate\_D$ and determine the impact. The modified data set with minimal impact is selected as the output of the algorithm.

It is apparent that exhaustive algorithm explores very large search space and is not practical for large data sets. However, since our focus in this paper is to demonstrate that the data reduction approach works very effectively for the problem of hiding sensitive classification rules, we apply this algorithm to find the optimal solution. Many heuristics could be used to find the near-optimal result much more quickly.

5.2 Experiment Setting

The experiments are conducted on an 3 GHz Intel Pentium 4 PC with 1024 megabytes main memory running Microsoft Window XP. The exhaustive algorithm is implemented by using JDK 5.0 based on Weka Data Mining Software (Witten & Frank 2005).

The experiment is performed on three real-life data sets from UCI repository (Blake & Merz 1998) i.e. mushroom, credit screening, and voting data sets. All data sets are transformed into binary data sets. Tuples with missing values are removed. The features of the data sets used in experiments are summarized in Table 3. Here we also give the rule summary under given parameter settings on minimal support and minimal confidence. Note that support listed in the table is the ratio of support values to the total number of tuples.

Table 3: Features of data sets

Detail	Data set		
	Voting	Credit Screening	Mushroom
#Tuples	232	653	5644
#Attributes	15	16	22
minsup	0.4	0.15	0.1
minconf	0.5	0.5	0.4
#General Rules	14	12	9
#All Rules	71	52	66
Support Range	0.40-0.51	0.16-0.44	0.1-0.33
Average #Literals	1.546	3.44	4.67

5.3 Data Quality

We investigate data quality of modified data sets by two factors: numbers of sensitive rules to be hidden ($|R_s|$), and support range of sensitive rules (the range of $sup(r_s)$). When we consider the effect of $|R_s|$, the range of $sup(r_s)$ will be fixed. In the same way, we fix $|R_s|$, when we consider the effect of $sup(r_s)$. In each experiment, for five times, we randomly select sensitive rules according to a specified setting of $|R_s|$ and $sup(r_s)$. Then, for each random selection, we compute the optimal impact by the exhaustive algorithm. Finally, we report the five-time-average impact on data quality, i.e., the average number of false-dropped rules and ghost rules for the given setting.

First, we report the effect of $|R_s|$ on the data quality in Table 4. The fixed ranges of $sup(r_s)$ are 0.42-0.44, 0.18-0.25, and 0.18-0.27 for voting, credit screening, and mushroom data sets respectively.

From Table 4, it can be seen that the data reduction approach can provide modified data sets with high data quality. Further, given a data set, for each experiment setting, we compute the percentage between the average number of false-dropped rules

Input:
D: a data set
$minsup$: a support threshold
$minconf$: a confidence threshold
R: the set of associative classification rules in D (satisfying $minsup$ and $minconf$)
R_s: the set of sensitive classification rules, $R_s \subset R$
Output:
D': the output data set, from which R_s can not be derived,
and the impact on data quality is minimal
Method:

```
1   Initialize Min_Impact;
2   for each rule r_s ∈ R_s
3     Compute D_s;
4     Determine the set D_Removal_s, such that the elements of D_Removal_s are
5      all the subset of D_s with the cardinality (Sup(r_s) − minsup + 1);
6   end for
7   Determine the set Global_D_Removal, whose elements are all possible tuple sets
8    that can hide the set of sensitive rules R_s, each element, denoted as GDR,
9    is formed by selecting one element from each D_removal_s and union them;
10  for each GDR ∈ Global_D_Removal do
11    Intermediate_D = D − GDR;
12    Recompute classification rules for data set Intermediate_D;
13    Determine impact;
14    if impact < Min_Impact
15      D' = Intermediate_D;
16    end if
17  end for
```

Figure 1: An exhaustive search algorithm.

Table 4: Impact on data quality in terms of $|R_s|$

Data set	$\|R_s\|$	# False-dropped rules	# Ghost rules
Voting	1	1.00	0.00
	2	1.40	0.00
	3	2.00	0.00
	4	1.00	0.00
	5	0.40	0.00
Credit Screening	1	0.40	1.00
	2	0.60	1.20
	3	0.80	2.00
	4	1.00	1.00
	5	0.00	1.00
Mushroom	1	1.60	0.00
	2	1.80	0.00
	3	1.40	0.00
	4	1.00	0.00
	5	0.40	0.00

(ghost rules) and the number of non-sensitive general rules. Then, we compute the average percentage for these settings. In our result, the average percentages between the numbers of false-dropped rules and the number of non-sensitive general rules are 10.0%, 6.2%, and 19.8% for voting, credit screening, and mushroom data sets respectively. For the ghost rules, these average percentages are 0%, 14.8%, and 0% for the three data sets.

We observe that when the number of sensitive rules is increased, the number of false-dropped rules is also increased for some period, then, it starts to decrease. The reason is because the numbers of derivable general rules in different data sets are the certain constant numbers (14, 12, 9 for voting, credit screening, and mushroom respectively). Therefore, the more general rules to be hidden, the less number of non-sensitive rules will be false-dropped.

We also observe that there is no ghost rule generated from voting and mushroom data set. For voting data set, the reason is because a high minimal support is used (0.42-0.44), when we hide sensitive rules, many more tuples will be removed. This means the potential ghost rules are also removed. While mushroom data set is very sparse, so, it is hard to find a new ghost rule when data reduction approach is used.

Table 5: Impact on data quality in terms of the range of $sup(r_s)$

Data set	range of $sup(r_s)$	# False-dropped rules	# Ghost rules
Voting	0.40-0.41	0.2	0.0
	0.42-0.43	1.4	0.0
	0.44-0.45	1.6	0.0
	0.46-0.47	2.6	0.0
	0.48-0.50	2.8	0.0
Credit Screening	0.20-0.24	0.2	0.2
	0.25-0.29	0.8	0.8
	0.30-0.34	1.0	1.2
	0.35-0.39	1.2	1.4
	0.40-0.45	1.8	1.6
Mushroom	0.10-0.15	0.0	0.0
	0.14-0.20	1.0	0.0
	0.19-0.25	1.6	0.0
	0.24-0.30	2.0	0.0
	0.29-0.34	2.2	0.0

In Table 5, the effect of $sup(r_s)$ to the data quality is shown. The numbers of sensitive rules $|R_s|$ are fixed at 2 for all data sets.

Unlike $|R_s|$, we can see that the impact on data quality increases when we increase $sup(r_s)$, though, it is still considered relatively low. Since we fix the number of sensitive rules at 2, from the results when we hide the rules with highest ranges of support, we still can preserve the majority of the non-sensitive general rules, i.e. the average numbers of preserved non-sensitive rules are 9.2 out of 12, 8.2 out of 10, and 4.8 out of 7 rules for voting, credit screening, and mushroom data sets respectively.

5.4 Size of Modified Data Set

Since there exists the risk that a modified data set by data reduction approach can become very small, in this section, we consider the number of remaining tuples in the modified data sets ($|D'|$). In this experiment, the ranges of $sup(r_s)$ are fixed at 0.42-0.44,

Table 6: Size of Modified Data Set

Data set	$\|D'\|/\|D\|$	Impact
Voting	0.885	# False-dropped rules = 1.6 # Ghost rules = 0.0
Credit Screening	0.907	# False-dropped rules = 0.2 # Ghost rules = 0.2
Mushroom	0.932	# False-dropped rules = 2.0 # Ghost rules = 0.0

0.18-0.25, and 0.18-0.27 for Voting, Credit Screening, and Mushroom data sets respectively. $|R_s|$ is set as 2 for the random sensitive rules selection in every data set. We consider that these settings are common situations for the problem, where the numbers of sensitive rules and their supports are moderate. We report the average ratio between $|D'|$ and the size of an original data set $|D|$. To help illustrating, we also report the average number of false-dropped rules and ghost rules under each experiment setting.

From Table 6, it can be seen that the sizes of the resulting data sets are not much different from the originals. We can see that voting data set has the least number of remaining tuples comparatively. This is because of the high support is used, thus, the more number of tuples need to be removed to hide sensitive rules. On the other hand, the size of the modified mushroom data set is comparatively high because of its sparseness.

6 Conclusions and Future Work

In this paper, we address the problem of hiding sensitive classification rules by data reduction. We focus on the problem in the context of associative classification rule mining. The data quality of a modified data set is defined by the number of false-dropped rules and the number of ghost rules. The main contributions are as follows. Firstly, we have introduced and defined the problem of hiding associative classification rules based on data reduction. Secondly, we have conducted the experiments on real data sets, and the results show that the data reduction approach can hide sensitive rules very effectively, i.e, the modified data sets have high data quality after the hiding process.

In our future work, we will focus on the efficiency of algorithms by this approach. Some heuristics will be applied to efficiently search the near-optimal solution. Also, we will target on addressing the problem where the given data sets have the attributes with richer domains, e.g., categorical or continuous attributes.

References

Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, *in* 'SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data', ACM Press, New York, NY, USA, pp. 207–216.

Atallah, M., Elmagarmid, A., Ibrahim, M., Bertino, E. & Verykios, V. (1999), Disclosure limitation of sensitive rules, *in* 'KDEX '99: Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange', IEEE Computer Society, Washington, DC, USA, pp. 45–52.

Blake, C. & Merz, C. (1998), 'UCI repository of machine learning databases'. *http://www.ics.uci.edu/∼mlearn/MLRepository.html

Clifton, C. (2000), 'Using sample size to limit exposure to data mining', *Journal of Computer Security* **8**(4), 281–307.

Evfimievski, A., Srikant, R., Agarwal, R. & Gehrke, J. (2004), 'Privacy preserving mining of association rules', *Information Systems* **29**(4), 343–364.

Fule, P. & Roddick, J. F. (2004), Detecting privacy and ethical sensitivity in data mining results, *in* 'ACSC '04: Proceedings of the 27th Australasian conference on Computer science', Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 159–166.

HajYasien, A. & Estivill-Castro, V. (2006), Two new techniques for hiding sensitive itemsets and their empirical evaluation, *in* 'Proceedings of 8th International Conference on Data Warehousing and Knowledge Discovery', Lecture Notes in Computer Science, Springer, pp. 302–311.

Li, W., Han, J. & Pei, J. (2001), Cmar: Accurate and efficient classification based on multiple class-association rules, *in* 'Proceedings of the 2001 IEEE ICDM International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 369–376.

Liu, B., Hsu, W. & Ma, Y. (1998), Integrating classification and association rule mining, *in* 'Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining', AAAI Press, pp. 80–86.

Machanavajjhala, A., Gehrke, J., Kifer, D. & Venkitasubramaniam, M. (2006), ℓ-diversity: Privacy beyond κ-anonymity, *in* 'ICDE '06: Proceedings of the 22nd International Conference on Data Engineering', IEEE Computer Society, Washington, DC, USA, p. 24.

Moustakides, G. V. & Verykios, V. S. (2006), A max-min approach for hiding frequent itemsets, *in* 'Workshops Proceedings of the 6th IEEE ICDM nternational Conference on Data Mining', IEEE Computer Society, pp. 502–506.

Oliveira, S. R. M. & Zaïane, O. R. (2002), Privacy preserving frequent itemset mining, *in* 'Proceedings of the IEEE international conference on Privacy, security and data mining', Australian Computer Society, Inc., pp. 43–54.

Oliveira, S. R. M. & Zaïane, O. R. (2003), Protecting sensitive knowledge by data sanitization., *in* 'Proceedings of the 3rd IEEE ICDM International Conference on Data Mining', IEEE Computer Society, pp. 613–616.

Oliveira, S. R. M., Zaïane, O. R. & Saygin, Y. (2004), Secure association rule sharing., *in* 'PAKDD '04: Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining', Springer-Verlag, pp. 74–85.

Saygin, Y., Verykios, V. S. & Clifton, C. (2001), 'Using unknowns to prevent discovery of association rules', *SIGMOD Rec.* **30**(4), 45–54.

Sun, X. & Yu, P. S. (2005), A border-based approach for hiding sensitive frequent itemsets., *in* 'Proceedings of the 5th IEEE ICDM International Conference on Data Mining', IEEE Computer Society, pp. 426–433.

Sweeney, L. (2002), 'k-anonymity: a model for protecting privacy', *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* **10**(5), 557–570.

Verykios, V. S., Elmagarmid, A. K., Bertino, E., Saygin, Y. & Dasseni, E. (2004), 'Association rule hiding', *IEEE Transactions on Data and Knowledge Engineering* **16**(4), 434–447.

Wang, K., Fung, B. C. M. & Yu, P. S. (2005), Template-based privacy preservation in classification problems, *in* 'ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 466–473.

Witten, I. H. & Frank, E. (2005), *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wong, R. C.-W., Li, J., Fu, A. W.-C. & Wang, K. (2006), (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing, *in* 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 754–759.

Wu, Y.-H., Chiang, C.-M. & Chen, A. L. P. (2007), 'Hiding sensitive association rules with limited side effects', *IEEE Transactions on Knowledge and Data Engineering* **19**(1), 29–42.