

Template-Based Privacy Preservation in Classification Problems *

Ke Wang
Simon Fraser University
BC, Canada V5A 1S6
wangk@cs.sfu.ca

Benjamin C. M. Fung
Simon Fraser University
BC, Canada V5A 1S6
bfung@cs.sfu.ca

Philip S. Yu
IBM T. J. Watson Research Center
Hawthorne, NY 10532
psyu@us.ibm.com

Abstract

In this paper, we present a template-based privacy preservation to protect against the threats caused by data mining abilities. The problem has dual goals: preserve the information for a wanted classification analysis and limit the usefulness of unwanted sensitive inferences that may be derived from the data. Sensitive inferences are specified by a set of “privacy templates”. Each template specifies the sensitive information to be protected, a set of identifying attributes, and the maximum association between the two. We show that suppressing the domain values is an effective way to eliminate sensitive inferences. For a large data set, finding an optimal suppression is hard, since it requires optimization over all suppressions. We present an approximate but scalable solution. We demonstrate the effectiveness of this approach on real life data sets.

1. Introduction

Knowledge Discovery in Databases (KDD) or data mining aims at finding out new knowledge about an application domain using collected data on the domain, typically data on individual entities like persons, companies, transactions. Naturally, the general concerns over data security and individual privacy are relevant for data mining. The first concern relates to the *input* of data mining methods due to data access. Many techniques have been proposed [4, 6, 8, 10, 12, 15, 18] to address this problem while preserving the benefits of data mining. The second concern is related to the *output* of data mining methods. Although the output of data mining methods are aggregate patterns, not intended to identify single individuals, they can be used to infer sensitive information about individuals. In this paper, we consider the privacy threats caused by such “data mining abilities”. Let us first consider an example.

*Research was supported in part by a research grant from Emerging Opportunity Fund of IRIS, and a research grant from the Natural Sciences and Engineering Research Council of Canada.

Job	Country	Child	Bankruptcy	Rating	#Rec
Cook	US	No	Current	0G/4B	4
Artist	France	No	Current	1G/3B	4
Doctor	US	Yes	Never	4G/2B	6
Trader	UK	No	Discharged	4G/0B	4
Trader	UK	No	Never	1G/0B	1
Trader	Canada	No	Never	1G/0B	1
Clerk	Canada	No	Never	3G/0B	3
Clerk	Canada	No	Discharged	1G/0B	1
Total:					24

Table 1. The initial table

Job	Country	Child	Bankruptcy	Rating	#Rec
Cook	US	No	Current	0G/4B	4
Artist	France	No	Current	1G/3B	4
Doctor	US	Yes	Never	4G/2B	6
\perp_{Job}	\perp_{Country}	No	Never	5G/0B	5
\perp_{Job}	\perp_{Country}	No	Discharged	5G/0B	5
Total:					24

Table 2. The suppressed table

Example 1 (Running Example). Table 1 contains records about bank customers. After removing irrelevant attributes, each row represents the duplicate records and the count. The class attribute Rating contains the class frequency of credit rating. For example, 0G/4B represents 0 Good and 4 Bad. Suppose that the bank (the data owner) wants to release the data to a data mining firm for classification analysis on Rating, but does not want the data mining firm to infer the bankruptcy state *Discharged* using the attributes Job and Country. For example, out of the 5 individuals with Job = *Trader* and Country = *UK*, 4 has the *Discharged* status. Therefore, the rule $\{Trader, UK\} \rightarrow Discharged$ has support 5 and confidence 80%. If the data owner tolerates no more than 75% confidence for this inference, the data is not safe for release. In general, currently bankrupted customers have a bad rating and simply removing the Bankruptcy column loses too much information for the classification analysis. ■

The private information illustrated in this example has the form “if x then y ”, where x identifies a group of individuals and y is a sensitive value. We consider this inference “private” if its confidence is high, in which case an individual in the group identified by x tends to be linked to y . The higher the confidence, the stronger the linking. In the context of data mining, association or classification rules [1, 14] are used to capture *general patterns of large populations* for summarization and prediction, where a low support means the lack of statistical significance. In the context of privacy protection, however, inference rules are used to infer *sensitive information about the existing individuals*, and it is important to eliminate sensitive inferences of any support, large or small. In fact, a sensitive inference in a small group could present even more threats than in a large group because individuals in a small group are more identifiable [16].

The problem considered in this paper can be described as follow. The data owner wants to release a version of data in the format

$$T(M_1, \dots, M_m, \Pi_1, \dots, \Pi_n, \Theta)$$

to achieve two goals. The **classification goal** is to preserve as much information as possible for modeling the *class attribute* Θ . The **privacy goal** is to limit the ability of data mining tools to derive inferences about *sensitive attributes* Π_1, \dots, Π_n . This requirement is specified using one or more templates of the form, $\langle IC \rightarrow \pi, h \rangle$, where π is a value from some Π_i , *inference channel* IC is a set of attributes not containing Π_i , and h is a threshold on confidence. The data satisfies $\langle IC \rightarrow \pi, h \rangle$ if every matching inference has a confidence no more than h . The privacy goal can be achieved by suppressing some values on *masking attributes* M_1, \dots, M_m . We are interested in a suppression of values for M_1, \dots, M_m that achieves both goals.

In Example 1, the inference $\{Trader, UK\} \rightarrow Discharged$ violates the template

$$\langle \{Job, Country\} \rightarrow Discharged, 75\% \rangle.$$

To eliminate this inference, we can suppress *Trader* and *Clerk* to a special value \perp_{Job} , and suppress *UK* and *Canada* to a special value $\perp_{Country}$, see Table 2. Now, the new inference $\{\perp_{Job}, \perp_{Country}\} \rightarrow Discharged$ has confidence 50%, less than the specified 75%. No information is lost since Rating does not depend on the distinction of the suppressed values.

The use of templates provides several flexibilities for specifying the notion of privacy: selectively protecting certain values π while not protecting other values; specifying a different threshold h for a different template $IC \rightarrow \pi$; specifying multiple inference channels IC (even for the same π); specifying templates for multiple sensitive attributes Π . These flexibilities provide not only a powerful representation of privacy requirements, but also a way to focus on the

problem area in the data to minimize unnecessary information loss.

Given that the classification task is known in advance, one may ask why not releasing a classifier, which likely contains less information, instead of the data. This could be an option if the data owner knows exactly how the data recipient may analyze the data. Often, this information is unknown. For example, in visual data mining, the data miner has to visualize data records in a certain way to guide the search, and in this case releasing data records is essential. Some classifiers such as the k-nearest neighbor are actually the data itself, and some are better in accuracy whereas others are better in interpretability. The data owner may not have the expertise to make such decisions because sophisticated data analysis is not part of its normal business.

The contributions of this work can be summarized as follows. First, we formulate a template-based privacy preservation problem. Second, we show that suppression is an effective way to eliminate sensitive inferences. However, finding an optimal suppression is a hard problem since it requires optimization over all possible suppressions. For a table with a total of q distinct values on masking attributes, there are 2^q possible suppressed tables. We present an approximate solution based on a search that iteratively improves the solution and prunes the search whenever no better solution is possible. We evaluate this method on real life data sets.

2. Related Work

Most works on privacy preservation address the concern related to the input of data mining tools where private information is revealed directly by inspection of the data without sophisticated analysis [4, 6, 8, 10, 12, 15, 18]. Our work is more related to the concern over the output of data mining methods, where the threats are caused by what data mining tools can discover. We focus on this group of works.

Kloesgen [13] pointed out the problem of group discrimination where the discovered group behavior is attached to all members in a group, which is a form of inferences. Clifton [3] suggested to eliminate sensitive inferences by limiting the data size. Recently, Kantarcioglu et al. [11] defined an evaluation method to measure the loss of privacy due to releasing data mining results. However, they did not propose a solution to prevent the adversary from getting data mining results that violate privacy.

Verykios et al. [17] proposed several algorithms for hiding association rules in a transaction database with minimal modification to the data. The general idea is to hide one rule at a time by either decreasing its support or its confidence; this is achieved by removing items from transactions. They need to assume that frequent itemsets of rules are disjoint in order to avoid high time complexity. We consider the

use of the data for classification analysis. Instead of minimizing pure syntax changes to the data, we minimize the information lose for this analysis and eliminate *all* sensitive inferences including those with a low support. We can efficiently handle overlapping inference rules.

Suppression of domain values was employed in [2, 10, 15] for achieving k -anonymity. The k -anonymity requirement states that no identifying attributes identify a group of size smaller than k . Therefore, if such identifying attributes are used to link the data to an external table, all the records in each group (at least k) will behave in the same way and are difficult to be distinguished. However, this notion does not address sensitive inference on the identified groups.

In database security, Farkas et al. [7] conducted a survey on inference control. In multilevel secure databases, the focus is detecting and removing inference channels by combining meta-data with data. For example, it is possible for a user to use a series of unsuspecting queries to infer sensitive data in the database. [19] proposed a method to detect such queries using functional dependencies. This type of inferences is different from ours.

In statistical databases, the focus is limiting the ability of inferring confidential information by correlating different statistics. For example, Cox [5] proposed the $k\%$ -dominance rule which suppresses a sensitive cell if the attribute values of two or three entities in the cell contribute more than $k\%$ of the corresponding SUM statistic. Such “cell suppression” suppresses the count or other statistics stored in a cell of a statistical table, which is very different from the “value suppression” considered in our work.

3. The Problem

Consider a table $T(M_1, \dots, M_m, \Pi_1, \dots, \Pi_n, \Theta)$. M_j are called *masking attributes*. Π_i are called *sensitive attributes*. Θ is called the *class attribute*. All attributes have a categorical domain. For each M_j , we add the special value \perp_j to its domain. For a domain value v , $att(v)$ denotes the attribute of v . M_j and Π_i are disjoint.

The data owner specifies sensitive inferences using templates. A *template* has the form $\langle IC \rightarrow \pi, h \rangle$. π is *sensitive value* from some Π_i . IC , called an *inference channel*, is some set of attributes not containing Π_i . h is a confidence threshold. An *inference* for $\langle IC \rightarrow \pi, h \rangle$ has the form $ic \rightarrow \pi$, where ic contains values from the attributes in IC . The *confidence* of $ic \rightarrow \pi$, written $conf(ic \rightarrow \pi)$, is the percentage of the records that contain π among those that contain the values in ic , that is, $s(ic, \pi)/s(ic)$, where $s(V)$ denotes the number of records containing the values in V . $Conf(IC \rightarrow \pi)$ denotes the maximum $conf(ic \rightarrow \pi)$ for all ic over IC .

Definition 3.1 (Privacy Templates). T satisfies a template

$\langle IC \rightarrow \pi, h \rangle$ if $Conf(IC \rightarrow \pi) \leq h$. T satisfies a set of templates if T satisfies every template in the set. ■

Some template may be “redundant” once we have some other template. The next lemma considers one such case, which can be used to remove “redundant” templates.

Lemma 3.1. Consider two templates $\langle IC \rightarrow \pi, h \rangle$ and $\langle IC' \rightarrow \pi', h' \rangle$. If $\pi = \pi'$, $h \geq h'$, and $IC \subseteq IC'$, then (1) $Conf(IC' \rightarrow \pi') \geq Conf(IC \rightarrow \pi)$, and (2) if T satisfies $\langle IC' \rightarrow \pi', h' \rangle$, T satisfies $\langle IC \rightarrow \pi, h \rangle$. ■

To see (1), consider the partition $\{P_1, \dots, P_k\}$ of the records matching ic according to the values x_1, \dots, x_k on $X = IC' - IC$. If the partitioning decreases the confidence of $ic \rightarrow \pi$ in some P_i , it must increase the confidence in some other P_j . Therefore, the partitioning does not decrease the maximum confidence in P_i 's. We omit the detailed proof. (2) follows immediately from (1).

If T violates a set of templates, (under certain conditions) we can suppress some values on masking attributes M_j to make it satisfy the templates. The *suppression* of a value on M_j means replacing *all* occurrences of the value with the special value \perp_j . Thus, all suppressed values on M_j are represented by the same \perp_j . One question is what makes us believe that suppression can reduce the confidence of sensitive inference. Indeed, if suppression actually increases the confidence, we are not getting any closer to the privacy goal but losing information for the classification goal. Below, we show that suppression *never* increases $Conf(IC \rightarrow \pi)$.

Consider suppressing a value v in M_j to \perp_j . The suppression affects only the records that contain v or \perp_j before the suppression. Let \perp_j and \perp'_j denote \perp_j before and after the suppression. The difference is that \perp'_j covers v but \perp_j does not. After the suppression, two inferences $\{ic, v\} \rightarrow \pi$ and $\{ic, \perp_j\} \rightarrow \pi$ become one inference $\{ic, \perp'_j\} \rightarrow \pi$. We have the next lemma, which has a similar proof as that of Lemma 3.1.

Lemma 3.2. $max\{conf(ic, v \rightarrow \pi), conf(ic, \perp_j \rightarrow \pi)\} \geq conf(ic, \perp'_j \rightarrow \pi)$. ■

The lemma says that, by suppressing a value, $Conf(IC \rightarrow \pi)$ does not go up. This property provides the basis for employing suppression to reduce $Conf(IC \rightarrow \pi)$.

Corollary 3.1. $Conf(IC \rightarrow \pi)$ is non-increasing with respect to suppression. ■

From Corollary 3.1, the *most suppressed* T , where all values for M_j are suppressed to \perp_j for every M_j in $\cup IC$, has the minimum $Conf(IC \rightarrow \pi)$. Therefore, if this table does not satisfy the templates, no suppressed T does.

Lemma 3.3. Given a set of templates, there exists a suppressed table T that satisfies the templates if and only if the most suppressed T satisfies the templates. ■

Definition 3.2 (Inference Problem). Given a table T and a set of templates, the *inference problem* is to (1) decide whether there exists a suppressed T that satisfies the set of templates, and if yes, (2) produce a satisfying suppressed T that preserves as much information as possible for modeling the class attribute. ■

4. The Algorithm

Given a table T and a set of templates $\{ \langle IC \rightarrow \pi, h \rangle \}$, our algorithm iteratively “discloses” the domain values starting from the most suppressed T in which each masking attribute M_j in $\cup IC$ contains only \perp_j , i.e., the set of suppressed values, Sup_j , contains all domain values in M_j . At any time, we have a set of *suppressed records*, with duplicates being collapsed into a single record with a count. In each iteration, we disclose one value from some Sup_j . To *disclose* a value v from Sup_j , we do exactly the opposite of suppressing v , i.e., replace \perp_j with v in all suppressed records that *currently* contain \perp_j and *originally* contain v before suppression. This process repeats until no disclosure is possible without violating the set of templates. From Corollary 3.1, any further disclosure leads to no solution.

Algorithm 1 Progressive Disclosure Algorithm (PDA)

- 1: suppress every value of M_j to \perp_j where $M_j \in \cup IC$;
 - 2: every Sup_j contains all domain values of $M_j \in \cup IC$;
 - 3: **while** there is a valid/beneficial candidate in $\cup Sup_j$ **do**
 - 4: find the winner w of highest $Score(w)$ from $\cup Sup_j$;
 - 5: disclose w on T and remove w from $\cup Sup_j$;
 - 6: update $Score(x)$ and the valid/beneficial status for x in $\cup Sup_j$;
 - 7: **end while**
 - 8: output the suppressed T and $\cup Sup_j$;
-

The above algorithm, called *Progressive Disclosure Algorithm (PDA)*, is presented in Algorithm 1. At each iteration, if some Sup_j contains a “valid” and “beneficial” candidate for disclosure, the algorithm chooses the winner candidate w that maximizes the score function denoted $Score$. A disclosure is *valid* if it leads to a table satisfying the set of templates. A disclosure from Sup_j is *beneficial* if more than one class is involved in the records containing \perp_j . Next, the algorithm discloses w , and updates the $Score$ and status of every affected candidate. We focus on the three key steps (Lines 4 to 6) in the rest of this section.

Example 2. Consider the templates:

$$\begin{aligned} \{ \{ \text{Job, Country} \} \} &\rightarrow \text{Discharged, 50\%}, \\ \{ \{ \text{Job, Child} \} \} &\rightarrow \text{Discharged, 50\%}. \end{aligned}$$

Initially, the values of Job, Country and Child in Table 1 are suppressed to \perp_{Job} , \perp_{Country} and \perp_{Child} , and $\cup Sup_j$ contains all domain values in Job, Country, and Child. This is

the most suppressed, or the least disclosed, state. ■

4.1. Find the Winner (Line 4)

This step finds the winner w , i.e., the valid and beneficial candidate from $\cup Sup_j$ that has the highest $Score$. Since disclosing a value v gains information and loses privacy, $Score(v)$ measures the *information gain* [14] per unit of privacy loss, defined as

$$Score(v) = \frac{InfoGain(v)}{PrivLoss(v) + 1}. \quad (1)$$

Consider the set of suppressed records that currently contain \perp_j , denoted $T[\perp_j]$. Disclosing v from Sup_j means replacing \perp_j with v in all records in $T[\perp_j]$ that originally contain v . Let T_v denote the set of such records, and let $T[v]$ denote T_v after replacing \perp_j with v . The disclosure of v is to replace $T[\perp_j]$ with $T[v]$ and $T[\perp_j] - T_v$. $InfoGain(v)$ is the information gain of this replacement. $InfoGain(v)$ depends only on the class frequency and count statistics on the single attribute $att(v)$ in $T[\perp_j]$, $T[v]$ and $T[\perp_j] - T_v$.

$PrivLoss(v)$ measures the privacy loss, defined as the average increase of $Conf(IC \rightarrow \pi)$ over all affected $IC \rightarrow \pi$, i.e., those IC such that $att(v)$ is contained in IC ,

$$avg\{Conf_v(IC \rightarrow \pi) - Conf(IC \rightarrow \pi) \mid att(v) \in IC\},$$

where $Conf$ and $Conf_v$ represent the confidence before and after disclosing v .

Computing $Conf_v$ efficiently is a challenge since it may involve count statistics on a combination of several attributes. It is inefficient to perform the disclosure of v in order to compute $Conf_v$. The key to the scalability of our algorithm is incrementally updating $Score(v)$ in each iteration for valid/beneficial candidates v in $\cup Sup_j$. We will present this update algorithm in Section 4.3.

4.2. Disclose the Winner (Line 5)

This step discloses the winner w and replaces \perp_j with w in the suppressed records in $T[\perp_j]$ that originally contain w . It requires accessing the raw records of these suppressed records. The following data structure facilitates the direct access to such raw records. The idea is to partition raw records according to their suppressed records on $\cup IC$.

Definition 4.1 (VIP). *Value Indexed Partitions (VIP)* contains the set of suppressed records over $\cup IC$. Each suppressed record represents the set of raw records from which it comes, called a *partition*. Each raw record is in exactly one partition. For each disclosed value x (including \perp) on an attribute in $\cup IC$, $P[x]$ denotes a partition represented by a suppressed record containing x . $Link[x]$ links up all $P[x]$'s, with the head stored with the value x . ■

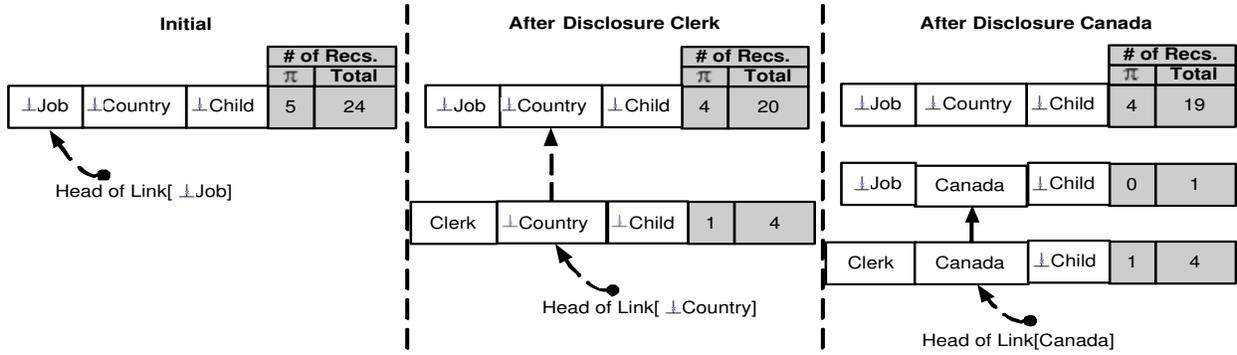


Figure 1. Evolution of VIP ($\pi = Discharged$)

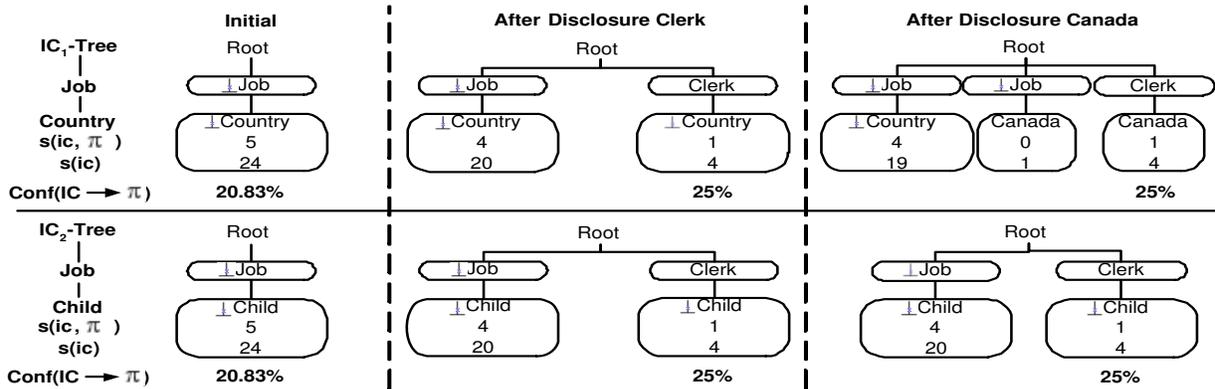


Figure 2. Evolution of IC-trees

To disclose the winner w , we follow the link $Link[\perp_w]$ and find all suppressed records that contain \perp_w , and through these suppressed records, access the represented raw records. Let \perp_w denote the special \perp value for the attribute of w . The following example illustrates the procedure of disclosing w in VIP.

Example 3. Consider the templates in Example 2. In Figure 1, the left-most VIP has the most suppressed record $\langle \perp_{Job}, \perp_{Country}, \perp_{Child} \rangle$ on three links: $Link[\perp_{Job}]$, $Link[\perp_{Country}]$, $Link[\perp_{Child}]$. The shaded fields “Total” and “ π ” contain the number of raw records suppressed (i.e., $|P|$) and the number of those records containing *Discharged*.

Suppose the winner is *Clerk*. We create a new suppressed record $\langle Clerk, \perp_{Country}, \perp_{Child} \rangle$, as shown in the middle VIP, to represent 4 raw records. We add this new suppressed record to $Link[\perp_{Country}]$, $Link[\perp_{Child}]$, and to the new $Link[Clerk]$. Finally, we remove *Clerk* from Sup_j . The next winner, *Canada*, refines the two partitions on $Link[\perp_{Country}]$, resulting in the right-most VIP. The overhead of maintaining these links is proportional to the length of $Link[\perp_w]$ and is negligible. ■

For the purpose of updating $Score(x)$ efficiently, we

also maintain the following *count statistics* for each partition P in the VIP: for every class θ and sensitive value π , (1) $|P|$, $s(\theta)$ and $s(\pi)$, (2) for each masking attribute M_j on which P has the value \perp_j , for every value v in Sup_j , $s(v)$, $s(\{v, \theta\})$ and $s(\{v, \pi\})$. All $s()$ refer to the partition P . These count statistics are stored together with the partition P and, on disclosing w , are updated as we scan the partitions on $Link[\perp_w]$.

We should mention that this step (Line 5) is the only time that raw records are accessed in our algorithm.

4.3. Update Score and Status (Line 6)

This step updates $Score(x)$ and the valid/beneficial status for x in $\cup Sup_j$. $InfoGain(x)$ is affected only if x and w are from the same attribute. $InfoGain(x)$ can be updated using the count statistics stored at the partitions on $Link[\perp_w]$, in the same scan as maintaining the count statistics in the previous step. Mark x as beneficial if there is more than one class in these partitions.

To update $PrivLoss(x)$, for every $IC \rightarrow \pi$, we first update $Conf(IC \rightarrow \pi)$ using $Conf_w(IC \rightarrow \pi)$ that was computed in the previous iteration. Next, we update $Conf_x(IC \rightarrow \pi)$ for x in $\cup Sup_j$. Observe that if $att(x)$

is not in IC , $Conf_x(IC \rightarrow \pi) = Conf(IC \rightarrow \pi)$; if $att(w)$ is not in IC , $Conf_x(IC \rightarrow \pi)$ is not affected by the disclosure of w . Therefore, we need to update $Conf_x(IC \rightarrow \pi)$ only if both $att(x)$ and $att(w)$ are contained in IC . We propose the following IC -tree structure to maintain $Conf(IC \rightarrow \pi)$.

Definition 4.2 (IC-trees). For each $IC = \{A_1, \dots, A_u\}$, the IC -tree is a tree of u levels, where level $i > 0$ represents the values for A_j . A root-to-leaf path represents an existing ic on IC in the suppressed T , with $s(ic)$ and $s(ic, \pi)$ stored at the leaf node. ■

Recall that $conf(ic \rightarrow \pi) = s(ic, \pi)/s(ic)$. $Conf(IC \rightarrow \pi)$ is given by $\max\{conf(ic \rightarrow \pi)\}$ for all ic in the IC -tree. All templates $\langle IC \rightarrow \pi, h \rangle$ with the same IC can share a single IC -tree by keeping $s(ic, \pi)$ separately for different π . We update IC -trees on disclosing w . Here is an example.

Example 4. Figure 2 shows the initial IC_1 -tree and IC_2 -tree on the left, where $IC_1 = \{\text{Job}, \text{Country}\}$ and $IC_2 = \{\text{Job}, \text{Child}\}$. On disclosing *Clerk*, $\{\text{Clerk}, \perp_{\text{Country}}\}$ and $\{\text{Clerk}, \perp_{\text{Child}}\}$ are created in IC_1 -tree and IC_2 -tree. Next, on disclosing *Canada*, $\{\text{Clerk}, \perp_{\text{Country}}\}$ is refined into $\{\text{Clerk}, \text{Canada}\}$ in IC_1 -tree, and a new $\{\perp_{\text{Job}}, \text{Canada}\}$ is split from $\{\perp_{\text{Job}}, \perp_{\text{Country}}\}$. To compute $s(ic)$ and $s(ic, \pi)$ for these ic 's, we access all partitions $P[\text{Canada}]$ in one scan of $Link[\text{Canada}]$ in the VIP:

$$\begin{aligned} s(\perp_{\text{Job}}, \text{Canada}) &= 1, \\ s(\perp_{\text{Job}}, \text{Canada}, \pi) &= 0, \\ s(\perp_{\text{Job}}, \perp_{\text{Country}}) &= 20 - 1 = 19, \\ s(\perp_{\text{Job}}, \perp_{\text{Country}}, \pi) &= 4 - 0 = 4. \blacksquare \end{aligned}$$

As discussed above, for $x \in \cup Sup_j$, if both $att(x)$ and $att(w)$ are in IC , we need to update $Conf_x(IC \rightarrow \pi)$. Recall that $Conf_x(IC \rightarrow \pi)$ is the maximum $conf(ic \rightarrow \pi)$ after disclosing x . Therefore, we can treat x as if it were disclosed, and computing $s(ic, x)$, $s(ic, x, \pi)$, $s(ic, \perp_x)$ and $s(ic, \perp_x, \pi)$ as we did for w . The only difference is that we perform these computations on a *copy* because we do not actually update the VIP and IC -trees for x . $Conf_x(IC \rightarrow \pi)$ is the new maximum $conf(ic \rightarrow \pi)$ in the IC -tree. If $Conf_x(IC \rightarrow \pi) \leq h$, mark x as valid.

4.4. The Cost Analysis

The cost at each iteration can be summarized as two operations. The first operation scans the partitions on $Link[\perp_w]$ for disclosing the winner w in VIP and maintaining some count statistics. The second operation simply makes use of the count statistics to update the score and status of every affected candidate without accessing data records. Thus, each iteration accesses only the records suppressed to \perp_w . The number of iterations is bounded by the number of distinct values in the masking attributes.

5. Experimental Evaluation

We evaluated how well the proposed method can preserve the usefulness for classification for some highly restrictive limiting requirements. We also evaluated the efficiency of this method. We adopted two widely used benchmarks from the UCI repository [9]: *Japanese Credit Screening* and *Adult*. We removed all continuous attributes since our method focuses on only categorical attributes. We used the C4.5 classifier [14] for classification modeling. All experiments were conducted on an Intel Pentium IV 3GHz PC with 1GB RAM.

Templates. We chose the best N attributes for the classification analysis, denoted $\text{Top}N$, as the sensitive attributes Π_1, \dots, Π_N . Simply removing such sensitive attributes will compromise the classification goal. The top most attribute is the attribute at the top of the C4.5 decision tree. Then we removed this attribute and repeated this process to determine the rank of other attributes. The remaining attributes were chosen as the masking attributes M_1, \dots, M_m . For each Π_i , we choose the 50% least frequent values as sensitive values. The rationale is that less frequent values are more vulnerable to inference attacks. Let $\{\pi_1, \dots, \pi_k\}$ denote the union of such values for all Π_i . The template is $\langle IC \rightarrow \{\pi_1, \dots, \pi_k\}, h \rangle$, where IC contains all masking attributes. From Lemma 3.1, this template is more restrictive than a set of multiple templates with each being a subset of IC (for the same threshold h).

Errors to measure. The *base error (BE)* refers to the error for the original data without suppression. The *suppression error (SE)* refers to the error for the data suppressed by our method. The suppression was performed before splitting the data into the training set and the testing set. $SE - BE$ measures the quality loss due to suppression, the smaller the better. We also compared with the error caused by simply removing all sensitive attributes, which is denoted by *removal error (RE)*. $RE - SE$ measures the benefit of suppression over this simple method, and the larger the better. Finally, $RE - BE$ measures the importance of sensitive attributes on classification. All errors are collected on the testing set.

5.1. Japanese Credit Screening

The *Japanese Credit Screening* data set, also known as *CRX*, is based on credit card application. There are 9 categorical attributes and a binary class attribute representing the application status *succeeded* or *failed*. After removing records with missing values, there are 465 and 188 records for the pre-split training and testing respectively. In the UCI repository, all values and attribute names in *CRX* have been changed to meaningless symbols, e.g., $A_1 \dots A_{15}$. We consider the four template requirements: $\text{Top}1$, $\text{Top}2$, $\text{Top}3$

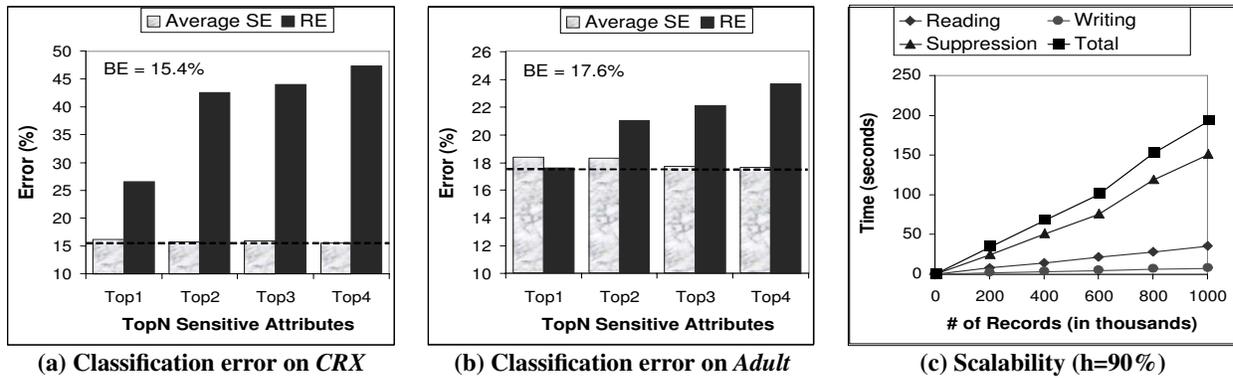


Figure 3. Experimental results

and Top4. $BE = 15.4\%$. Table 3 shows the number of inferences above different confidence thresholds h in the original data. For example, the number of inferences that have a confidence larger than 90% is 6 in *CRX* for Top4.

Threshold h	10%	30%	50%	70%	90%
CRX (Top4)	40	27	15	8	6
Adult (Top4)	1333	786	365	324	318

Table 3. Number of inferences above h

Figure 3a depicts SE and RE for TopN averaged over $h = 50\%, 70\%, 90\%$. The dashed line represents BE . We summarize the results as follows:

1. SE spans narrowly between 15.4% and 16.5% across different TopN. $SE - BE$ is less than 1.1% for all sets of templates considered. These results support that inference limiting and accurate classification can coexist. For example, from Table 3, 15 inferences with a confidence higher than 50% were eliminated for Top4. Often, different ic's share some common values, and suppressing a few common values simultaneously eliminates multiple inferences.
2. The minimum $RE - SE$ is 10.1% for Top1, and the maximum $RE - SE$ is 31.3% for Top4. These large gaps show a significant benefit of suppression over the removal of sensitive attributes.
3. For all templates tested, the variance of SE is less than 0.6%, suggesting that suppression is robust. It also suggests that protecting more sensitive attributes (i.e., a larger N in TopN) or having a lower threshold h does not necessarily compromise the classification quality. In fact, as N increases, more sensitive attributes can help the classification.

4. Having more sensitive attributes (i.e., a larger N in TopN) implies that the removal of these attributes has a larger impact to classification. This is reflected by the increasing RE in Figure 3a.

5. The algorithm took less than 2 seconds, including disk I/O operations, for all the above experiments.

Let us take a closer look at the suppressed data for Top4 with $h = 70\%$. Some values of attributes A_4 and A_5 are suppressed, and the entire A_{13} is suppressed. Despite such vigorous suppression, $SE = 15.4\%$ is equal to BE . In fact, there exist multiple classification structures in the data. When suppression eliminates some of them, other structures emerge to take over the classification. Our method makes use of such "rooms" to eliminate sensitive inferences while preserving the quality of classification.

5.2. Adult

The *Adult* data set is a census data previously used in [2, 8, 10, 18]. There are 8 categorical attributes and a binary class attribute representing the income levels $\leq 50K$ or $> 50K$. There are 30,162 and 15,060 records without missing values for the pre-split training and testing respectively. Table 4 describes each categorical attribute. Top4 attributes are M, Re, E, S in that order. $BE = 17.6\%$.

Attribute	# of Values	Attribute	# of Values
Education (E)	16	Marital-status (M)	7
Occupation (O)	14	Native-country (N)	40
Race (Ra)	5	Relationship (Re)	6
Sex (S)	2	Work-class (W)	8

Table 4. Attributes for the *Adult* data set

Figure 3b shows the errors for TopN, averaged over $h = 10\%, 30\%, 50\%, 70\%, 90\%$. We summarize the results as follows: **(1)** $SE - BE$ is less than 0.8% in all cases. This is amazing considering that hundreds of inferences were eliminated according to Table 3. **(2)** The largest $RE - SE$ is approximately 6% for Top4. **(3)** The difference between maximum and minimum SE is less than 1%. **(4)** For Top1, RE is slightly lower than SE , implying that removing the top attribute does not affect the classification. However, as more sensitive attributes were removed (i.e., Top2, Top3, Top4), RE picked up. **(5)** The algorithm spent at most 14 seconds for all experiments on *Adult*, of which approximately 10 seconds were spent on suppressing the 45,222 data records.

5.3. Scalability

The purpose of this experiment is to see how scalable our method is for large data sets. We evaluated the scalability on an expanded version of *Adult*. We first combined the training and testing sets, giving 45,222 records. Then for each original record r in the combined set, we created $\alpha - 1$ “variations” of r , where $\alpha > 1$ is the *expansion scale*. For each variation of r , we randomly and uniformly selected y attributes from UIC , selected some random values for these y attributes, and inherited the values of r on the remaining attributes, including the class and sensitive attributes. Together with original records, the expanded data set has $\alpha \times 45,222$ records.

Figure 3c depicts the runtime of our suppression method for 200K to 1M data records based on the templates $\langle IC \rightarrow \{\pi^1, \dots, \pi^k\}, 90\% \rangle$, where $\{\pi^1, \dots, \pi^k\}$ is the set of 50% least frequent values in the Top1 attribute M , and IC contains the other 7 attributes. This is one of the most time consuming settings because of the largest number of disclosure candidates to consider at each iteration, and a larger h requires more iterations to reach a solution. Our method spent 192 seconds to suppress 1M records, of which 150 seconds were spent on suppression, and the rest was spent on disk I/O operations.

6. Conclusions

We studied the problem of eliminating the sensitive inferences that are made possible by data mining tools, while preserving the classification value of the data. A sensitive inference has a high confidence in linking a group of individuals to sensitive values. We eliminated sensitive inferences by letting the user specify the templates and maximum confidence for such inferences. We used suppression of domain values as a way to achieve this goal. We presented a progressive disclosure algorithm that iteratively searches for a better suppression and prunes the search whenever no better alternative is possible. Experiments on

real life data sets showed that the proposed approach preserves the information for classification modeling even for very restrictive privacy requirements.

References

- [1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large datasets. In *Proc. of the 1993 ACM SIGMOD*, pages 207–216, 1993.
- [2] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proc. of the 21st IEEE ICDE*, pages 217–228, 2005.
- [3] C. Clifton. Using sample size to limit exposure to data mining. *Journal of Computer Security*, 8(4):281–307, 2000.
- [4] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving data mining. *SIGKDD Explorations*, 4(2), 2002.
- [5] L. H. Cox. Suppression methodology and statistical disclosure control. *Journal of the American Statistics Association, Theory and Method Section*, 75:377–385, 1980.
- [6] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of the 8th ACM SIGKDD*, pages 217–228, 2002.
- [7] C. Farkas and S. Jajodia. The inference problem: A survey. *SIGKDD Explorations*, 4(2):6–11, 2003.
- [8] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of the 21st IEEE ICDE*, pages 205–216, Tokyo, Japan, 2005.
- [9] S. Hettich and S. D. Bay. The UCI KDD Archive, 1999. <http://kdd.ics.uci.edu>.
- [10] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proc. of the 8th ACM SIGKDD*, 2002.
- [11] M. Kantarcioglu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proc. of the 2004 ACM SIGKDD*, pages 599–604, 2004.
- [12] J. Kim and W. Winkler. Masking microdata files. In *ASA Proc. of the Section on Survey Research Methods*, 1995.
- [13] W. Kloesgen. Knowledge discovery in databases and data privacy. In *IEEE Expert Symposium: Knowledge Discovery in Databases*, 1995.
- [14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [15] L. Sweeney. Datafly: A system for providing anonymity in medical data. In *Proc. of the 11th International Conference on Database Security*, pages 356–381, 1998.
- [16] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5):571–588, 2002.
- [17] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. *IEEE TKDE*, 16(4):434–447, 2004.
- [18] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-up generalization: a data mining solution to privacy protection. In *Proc. of the 4th IEEE ICDM*, 2004.
- [19] R. W. Yip and K. N. Levitt. The design and implementation of a data level database inference detection system. In *Proc. of the 12th International Working Conference on Database Security XII*, pages 253–266, 1999.