Knowl Inf Syst (2007) 11(3): 345–368
DOI 10.1007/s10115-006-0035-5

Knowledge and Information Systems

REGULAR PAPER

**Ke Wang · Benjamin C. M. Fung ·
Philip S. Yu**

Handicapping attacker's confidence: an alternative to *k*-anonymization

Received: 30 November 2005 / Revised: 20 December 2005 / Accepted: 20 February 2006 /
Published online: 3 October 2006

Abstract We present an approach of limiting the confidence of inferring sensitive properties to protect against the threats caused by data mining abilities. The problem has dual goals: preserve the information for a wanted data analysis request and limit the usefulness of unwanted sensitive inferences that may be derived from the release of data. Sensitive inferences are specified by a set of "privacy templates". Each template specifies the sensitive property to be protected, the attributes identifying a group of individuals, and a maximum threshold for the confidence of inferring the sensitive property given the identifying attributes. We show that suppressing the domain values monotonically decreases the maximum confidence of such sensitive inferences. Hence, we propose a data transformation that minimally suppresses the domain values in the data to satisfy the set of privacy templates. The transformed data is free of sensitive inferences even in the presence of data mining algorithms. The prior *k*-anonymizationfocuses on personal identities. This work focuses on the association between personal identities and sensitive properties.

Keywords Privacy protection · *k*-anonymity · Sensitive inference · Data mining · Classification · Data sharing

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, typ-

K. Wang (✉) · B. C. M. Fung
School of Computer Science, Simon Fraser University, BC, Canada V5A 1S6
E-mail: wangk@cs.sfu.ca

P. S. Yu
IBM T. J. Watson Research Center, Hawthorne, NY 10532, USA

Table 1 The initial table

Job	Country	Child	Bankruptcy	Rating	No. of records
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

ically 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, 9, 11, 17, 21] 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 properties about individuals. In this paper, we consider the privacy threats caused by such "data mining abilities". Let us first consider an example.

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 five individuals with Job $=$ *Trader* and Country $= UK$, four 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 property. 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, 15] 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 properties 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

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

to achieve two goals. The **privacy goal** is to limit the ability of data mining tools to derive inferences about *sensitive attributes* $\Pi_1, \ldots, \Pi_n$. This requirement is specified using one or more templates of the form, $\langle QID \rightarrow \pi, h\rangle$, where π is a "value" or "property" from some Π_i, *quasi-identifier QID* is a set of "attributes" not containing Π_i, and h is a threshold on confidence. Each value over *QID* identifies a group of individuals. The data satisfies $\langle QID \rightarrow \pi, h\rangle$ if every inference matching the template has a confidence no more than h. The privacy goal is achieved by suppressing some values on *masking attributes* $M_1, \ldots, M_m$. The **data analysis goal** is to preserve as much information as possible for a specified data analysis task. To measure the "information" in a concrete way, we primarily consider the task of modeling some *class attribute* Θ in the data. Other notions of information utility can be captured by replacing the information component of our metric, therefore, require little modification to our approach. We assume that attributes $\Pi_1, \ldots, \Pi_n$ and $M_1, \ldots, M_m$ are important, thus, simply removing them fails to address the classification goal. We are interested in a suppression of values for $M_1, \ldots, M_m$ that achieves both goals.

Example 2 In Example 1, the inference $\{Trader, UK\} \rightarrow Discharged$ violates the template

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

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

Several points are worth noting.

First, the use of privacy templates is a flexibility, not a restriction. The data owner can selectively protect certain sensitive properties π while not protecting other properties, specify a different threshold h for a different template $QID \rightarrow \pi$, specify multiple quasi-identifiers *QID* (even for the same π), specify 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. In the case that the

Table 2 The suppressed table

Job	Country	Child	Bankruptcy	Rating	No. of records
Cook	US	No	Current	0G/4B	4
Artist	France	No	Current	1G/3B	4
Doctor	US	Yes	Never	4G/2B	6
$\perp_{\mathsf{Job}}$	$\perp_{\mathsf{Country}}$	No	Never	5G/0B	5
$\perp_{\mathsf{Job}}$	$\perp_{\mathsf{Country}}$	No	Discharged	5G/0B	5
Total					24

inferences through all *QID*s are to be limited, the data owner only needs to specify the "most restrictive" *QID* containing all the attributes that occur in any *QID* (more details in Sect. 3.1).

Second, this work differs from the prior work on k-anonymity [16] in a major way. The k-anonymity prevents linking personally identifying attributes to sensitive properties by requiring that at least k records share each description of the identifying attributes. The focus is on anonymizing the identifying attributes that define groups. However, if all or most individuals in a group are associated with the same sensitive property, the sensitive property for the group can be inferred with little uncertainty. Machanavajjhala et al. [13] address this problem by requiring "diversity" of the sensitive property in each group. In particular, their "entropy l-diversity", which ensures that sensitive properties are "well-represented" in a group, could be used to limit the confidence of attacks. A larger entropy means a more uniform distribution of sensitive properties in a group, therefore, less association with a particular sensitive property. For example, for a group of 100 records associated with two different diseases, if 90 records are associated with HIV and the other 10 records are associated with Flu, then this group is entropy 1.4-diverse. A major limitation of this approach is that entropy is not a "user-intuitive" measure of risk. In particular, the entropy 1.4-diverse does not convey that inferring HIV has 90% probability of success. Therefore, the data holder may find it difficult to specify her risk tolerance in terms of the confidence of attacks. In the case that HIV occurs less frequently but is more sensitive, their method allows the user to incorporate "background knowledge" to specify different protection for HIV and Flu. Our approach incorporates the background knowledge by allowing the data holder to specify different maximum confidence for different sensitive properties, based on prior knowledge such as the sensitivity and frequency of such properties.

Third, releasing a classifier, instead of the data, could be an option if the data owner knows exactly how the data recipient may analyze the data. Often, however, this information, even the data recipient in such cases as web publishing, is unknown. For example, in visual data mining the data recipient needs to visualize data records in order to produce a classifier that makes sense, and in the k-nearest neighbor classification the data itself is the classifier; in such cases releasing data records is essential. In other cases, some classifiers are preferred for accuracy, some for precision/recall, some for interpretability, and yet some for certain domain-specific properties. The data owner (such as a hospital) does not have the expertise to make such decisions for the data recipient (such as biomedical researchers) due to the lack of domain knowledge and sophisticated data mining techniques. For this reason, we consider the data release for the classification problem, not for individual classifiers or algorithms.

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. In particular, we iteratively disclose domain values in a top-down manner by first suppressing all domain values. In each iteration, we disclose the suppressed domain value to maximize some criterion taking into account both information gained and privacy lost. We evaluate this method on real-life data sets. Several features make this approach practically useful:

- *No taxonomy required*. Suppression replaces a domain value with ⊥ without requiring a taxonomy of values. This is a useful feature because most data do not have an associated taxonomy, though taxonomies may exist in certain specialized domains.
- *Preserving the truthfulness of values*. The special value ⊥ represents the "union", a less precise but truthful representation, of suppressed domain values. This truthfulness is useful for reasoning and explaining the classification model.
- *Subjective notion of privacy*. The data owner has the flexibility to define her own notion of privacy using templates for sensitive inferences.
- *Efficient computation*. It operates on simple but effective data structures to reduce the need for accessing raw data records.
- *Anytime solution*. At any time, the user can terminate the computation and have a table satisfying the privacy goal.
- *Extendibility*. Though we focus on categorical attributes and classification analysis, this work can be easily extended to continuous attributes and other information utility criteria. This extension will be elaborated in Sect. 6.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 defines the inference limiting problem. Section 4 presents our suppression approach. Section 5 evaluates the effectiveness of the proposed approach. Section 6 discusses several extensions. Section 7 concludes the paper.

2 Related work

Most works on privacy preservation address the concern related to the input of data mining where sensitive properties are revealed directly by inspection of the data without sophisticated analysis [4, 6, 8, 9, 11, 17, 21]. Our work is more related to the concern over the output of data mining in terms of what data mining tools can discover. We focus on this group of works.

Kloesgen [12] 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. [10] 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 attacker from getting data mining results that violate privacy.

Verykios et al. [19] 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, 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 eliminate *all* sensitive inferences including those with a low support. We can efficiently handle overlapping inference rules. Our approach handles the information lose for classification analysis as well as the general notion of data distortion in a uniform manner.

Suppression and generalization of domain values were employed in [2, 8, 9, 16, 21] for achieving k-anonymity. In a k-anonymized database, if one record is linked to some external sensitive property, so are at least $k-1$ other records. In other words, at least k records are indistinguishable to the linking algorithm. However, if all or most of these records are associated with similar sensitive property, this indistinguishability becomes irrelevant in that the attacker can reliably infer the sensitive property. In the preliminary work [20], we proposed the confidence of inference as a way to measure this threat. Recently, Machanavajjhala et al. [13] proposed the diversity of sensitive property as a way to make inferring a particular sensitive property uncertain. However, as explained earlier, it is more natural and intuitive for the data holder to measure the risk in terms of the probability of success of attacks.

In database security, Farkas and Jajodia [7] conducted a survey on inference control. In multilevel secure databases, the focus is detecting and removing quasi-identifiers by combining meta-data with data. Many of these methods operate at the schema-level and consider only precise inferences that always hold. If there is a security problem, the database is redesigned. Yip and Levitt [22] extended the work to the data-level by monitoring queries using functional dependencies. For example, it is possible for a user to use a series of unsuspicious queries to infer sensitive properties in the database. Yip and Levitt [22] 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

Let v be a single value, V be a set of values, and R be a set of records. $att(v)$ denotes the attribute of a value v. $|R|$ denotes the number of records in R. $R[v]$ denotes the set of records in R that contain v. $s(V)$ denotes the number of records containing the values in V. $f(R, V)$ denotes the number of records in R that contain the values in V. Sometimes, we simply list the values in V, i.e, $s(v_1, \ldots, v_k)$ and $f(R, v_1, \ldots, v_k)$, where v_j is either a single value or a set of values.

Consider a table $T(M_1, \ldots, M_m, \Pi_1, \ldots, \Pi_n, \Theta)$. 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. M_j and Π_i are disjoint.

Suppose that the data owner wants to release the table T to the public for modeling the class attribute Θ, but wants to limit the ability of making inference about sensitive attributes Π_i. An inference about sensitive property y has the form of "if x then y". Such inferences are "probabilistic", not "precise", and are easily obtained from the released data by applying today's data mining tools. If an inference is highly confident (i.e., accurate), there is little difficulty to infer sensitive property y about an individual matching the description x. One way to eliminate such threats is to limit the confidence of inferences. Below, we formalize this notion of privacy requirement into a set of privacy templates.

3.1 Privacy templates

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

Definition 3.1 (Privacy Templates) T satisfies *a template* $\langle QID \rightarrow \pi, h\rangle$ *if* $Conf(QID \rightarrow \pi) \leq h$. T satisfies *a set of templates if* T *satisfies every template in the set.*

A privacy template places an upper limit on the confidence of the specified inferences, including those involving $\perp_j$. For convenience, all templates $\langle QID \rightarrow \pi^i, h\rangle$ that only differ in π^i can be abbreviated as $\langle QID \rightarrow \{\pi^1, \ldots, \pi^k\}, h\rangle$. This is only a notational abbreviation, not a new kind of inferences.

Some template may be "redundant" once we have some other template. Theorem 3.1 considers one such case, which can be used to remove "redundant" templates.

Theorem 3.1 *Consider two templates*

$$\langle QID \rightarrow \pi, h\rangle \textit{ and } \langle QID' \rightarrow \pi', h'\rangle.$$

If $\pi = \pi'$, $h \geq h'$, *and* $QID \subseteq QID'$, *then*

1. $Conf(QID' \rightarrow \pi') \geq Conf(QID \rightarrow \pi)$, *and*
2. *If* T *satisfies* $\langle QID' \rightarrow \pi', h'\rangle$, T *satisfies* $\langle QID \rightarrow \pi, h\rangle$, *and*
3. $\langle QID \rightarrow \pi, h\rangle$ *can be removed in the presence of* $\langle QID' \rightarrow \pi', h'\rangle$.

Proof (1) Let $X = QID' - QID$. Assume that $X \neq \emptyset$. Consider an inference $qid \rightarrow \pi$ for $QID \rightarrow \pi$. Let $\{qid, x_1\} \rightarrow \pi, \ldots, \{qid, x_k\} \rightarrow \pi$ be the inferences for $QID' \rightarrow \pi$ involving qid. $s(qid) = \sum_{i=1}^{k} s(qid, x_i)$ and $s(qid, \pi) = \sum_{i=1}^{k} s(qid, x_i, \pi)$. Without loss of generality, we assume, for $2 \leq i \leq k$,

$$conf(qid, x_1 \rightarrow \pi) \geq conf(qid, x_i \rightarrow \pi).$$

We prove that $conf(qid, x_1 \rightarrow \pi) \geq conf(qid \rightarrow \pi)$; it then follows that $Conf(QID' \rightarrow \pi') \geq Conf(QID \rightarrow \pi)$ because $\pi' = \pi$. The intuition of the proof is similar to that of $max\{avg(M), avg(F)\} \geq avg(G)$, where a group G of people is divided into the male group M and the female group F, and $avg(x)$ computes the average age of a group x.

First, we rewrite $conf(qid, x_1 \rightarrow \pi) \geq conf(qid, x_i \rightarrow \pi)$ into

$$s(qid, x_1, \pi)s(qid, x_i) \geq s(qid, x_i, \pi)s(qid, x_1).$$

Recall that $s(qid) = \sum_{i=1}^{k} s(qid, x_i)$ and $s(qid, \pi) = \sum_{i=1}^{k} s(qid, x_i, \pi)$. Then, we have the following rewriting

$$\begin{aligned}
conf(qid, x_1 \rightarrow \pi) &= \frac{s(qid, x_1, \pi)}{s(qid, x_1)} \\
&= \frac{s(qid, x_1, \pi) \sum_{i=1}^{k} s(qid, x_i)}{s(qid, x_1)s(qid)} \\
&= \frac{s(qid, x_1, \pi)}{s(qid)} + \sum_{i=2}^{k} \frac{s(qid, x_1, \pi)s(qid, x_i)}{s(qid, x_1)s(qid)} \\
&\geq \frac{s(qid, x_1, \pi)}{s(qid)} + \sum_{i=2}^{k} \frac{s(qid, x_i, \pi)s(qid, x_1)}{s(qid, x_1)s(qid)} \\
&= \frac{s(qid, x_1, \pi)}{s(qid)} + \sum_{i=2}^{k} \frac{s(qid, x_i, \pi)}{s(qid)} \\
&= \frac{s(qid, \pi)}{s(qid)} = conf(qid \rightarrow \pi)
\end{aligned}$$

(2) follows from (1) and $h \geq h'$.
(3) follows from (2). □

The following corollary follows from Theorem 3.1. It states that only the "maximal" templates need to be specified among those having the same sensitive property π and confidence threshold h.

Corollary 3.1 *T satisfies $\langle QID \rightarrow \pi, h\rangle$ if and only if T satisfies $\{\langle QID' \rightarrow \pi, h\rangle \mid QID' \subseteq QID\}$.*

3.2 Suppression

If T violates the set of templates, we can suppress some values on masking attributes M_j to make it satisfy the templates (under certain conditions). *Suppression* of a value on M_j means replacing *all* occurrences of the value with the special value $\perp_j$. In the classification modeling, $\perp_j$ is treated as a new domain value in M_j.

An interesting question is what makes us believe that suppression of values can reduce the confidence of sensitive inference. Indeed, if suppression could increase the confidence, we are not getting any closer to the privacy goal but losing information. Below, we show that suppression *never* increases $Conf(QID \rightarrow \pi)$.

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 $\{qid, v\} \rightarrow \pi$ and $\{qid, \perp_j\} \rightarrow \pi$ become one inference $\{qid, \perp'_j\} \rightarrow \pi$.

Theorem 3.2 $max\{conf(qid, v \rightarrow \pi), conf(qid, \perp_j \rightarrow \pi)\} \geq conf(qid, \perp'_j \rightarrow \pi)$.

The proof is similar to Theorem 3.1, except that $\{qid, x_1\} \rightarrow \pi, \ldots, \{qid, x_k\} \rightarrow \pi$ are replaced with $\{qid, v\} \rightarrow \pi$ and $\{qid, \perp_j\} \rightarrow \pi$, and $qid \rightarrow \pi$ is replaced with $\{qid, \perp'_j\} \rightarrow \pi$. In words, Theorem 3.2 says that, by suppressing a value, $Conf(QID \rightarrow \pi)$ does not go up. This property provides the basis for employing suppression to reduce $Conf(QID \rightarrow \pi)$.

Corollary 3.2 *$Conf(QID \rightarrow \pi)$ is non-increasing with respect to suppression.*

3.3 The problem statement

Given a table T and a set of privacy templates $\{\langle QID^1 \rightarrow \pi^1, h^1\rangle, \ldots, \langle QID^k \rightarrow \pi^k, h^k\rangle\}$, we are interested in finding a suppressed table T that satisfies the set of templates and is useful for modeling the class attribute. The first question is whether it is always possible to satisfy the set of templates by suppressing T. The answer is no if for some $\langle QID^i \rightarrow \pi^i, h^i\rangle$, the minimum $Conf(QID^i \rightarrow \pi^i)$ among all suppressed T is above h^i. From Corollary 3.2, the *most suppressed* T, where all values for M_j are suppressed to $\perp_j$ for every M_j in $\cup QID^i$, has the minimum $Conf(QID^i \rightarrow \pi^i)$. If this table does not satisfy the templates, no suppressed T does.

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

In Table 1, $Conf(\{$Job, Country$\} \rightarrow Discharged)$ for the most suppressed T is $5/24$. Therefore, for any $h < 5/24$, this template is not satisfiable by suppressing T.

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.*

We can first apply Theorem 3.3 to determine if the set of privacy templates is satisfiable by suppressing T. If not, we inform the data owner and provide the actual $Conf(QID \rightarrow \pi)$ where $\langle QID \rightarrow \pi, h\rangle$ is violated. With this information, the data owner could adjust the templates, such as reconsidering whether the threshold h is reasonable. In the subsequent sections, we assume that the given set of privacy templates is satisfiable by suppressing T.

4 The algorithm

Given a table T (in which all values are disclosed) and a set of templates $\{\langle QID \rightarrow \pi, h\rangle\}$, there are two approaches to suppress T. One is iteratively suppressing domain values in M_j in $\cup QID$, called *bottom-up suppression*, and the other is first suppressing all domain values in M_j in $\cup QID$ and then iteratively disclosing the suppressed domain values, called *top-down disclosure*. We take the second approach. At any time in the top-down disclosure, we have a set of *suppressed values*, denoted Sup_j for M_j, and a set of *suppressed records*, with duplicates being collapsed into a single record with a count. In each iteration, we disclose one value v chosen from some Sup_j by doing exactly the opposite of suppressing v, i.e., replacing $\perp_j$ with v in all suppressed records that *currently* contain $\perp_j$ and *originally* contain v in the input table. This process repeats until no disclosure is possible without violating the set of templates.

The top-down disclosure approach has several nice features. First, any table produced by a sequence of disclosures can be produced by a sequence of suppressions. In fact, Sup_j on the termination of the algorithm tells exactly the suppressions on M_j needed to produce the suppressed table. Second, $Conf(QID \to \pi)$ is nondecreasing with respect to disclosures (Corollary 3.2). Therefore, any further disclosure beyond the termination leads to no solution. Third, compared to the bottom-up suppression starting from domain values, the top-down disclosure can handle restrictive privacy templates with a smaller number of iterations starting from the most suppressed table. In fact, by walking from a more suppressed table towards a less suppressed table, we always deal with a small number of satisfying inferences and never examine the large number of violating inferences in a less suppressed table. Finally, the user can terminate the disclosure process at any time and have a table satisfying the privacy templates.

Algorithm 4.1. Top-Down Disclosure (TDD)
Input: a table $T(M_1, \ldots, M_m, \Pi_1, \ldots, \Pi_n, \Theta)$ and a set of privacy templates
Output: a suppressed table satisfying the given privacy templates

```
1: suppress every value of $M_j$ to $\perp_j$ where $M_j \in \cup QID$;
2: every $Sup_j$ contains all domain values of $M_j \in \cup QID$;
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 every $x$ in $\cup Sup_j$;
7: end while
8: output the suppressed $T$ and $\cup Sup_j$;
```

Our algorithm, called *top-down disclosure (TDD)*, is presented in Algorithm 4.1. 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. Below, we focus on the three key steps (Lines 4–6):

Line 4: Find the winner candidate w. This step finds the valid and beneficial candidate w from $\cup Sup_j$ that has the highest *Score*. We discuss the computation of *Score* in Sect. 4.1.

Line 5: Disclose the winner candidate w. This step discloses w in T and removes w from Sup_j. We discuss an efficient method for performing a disclosure in Sect. 4.2.
Line 6: Update the score and status for candidates. This step updates $Score(x)$ and valid/beneficial status for the candidates x in $\cup Sup_j$ to reflect the impact of w. We discuss an efficient update in Sect. 4.3.

Example 3 Consider the templates:
$\langle\{\mathsf{Job}, \mathsf{Country}\} \rightarrow \mathit{Discharged}, 50\%\rangle$,
$\langle\{\mathsf{Job}, \mathsf{Child}\} \rightarrow \mathit{Discharged}, 50\%\rangle$.
Initially, the values of Job, Country and Child in Table 1 are suppressed to $\perp_{\mathsf{Job}}$, $\perp_{\mathsf{Country}}$ and $\perp_{\mathsf{Child}}$, 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)

The winner w is a 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*, denoted $InfoGain(v)$, per unit of privacy loss, denoted $PrivLoss(v)$, due to the disclosure of v:

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

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 $T[\perp_j] - T_v$.

$InfoGain(v)$: One way to measure $InfoGain(v)$ for the classification of the specified class attribute is the standard entropy-based information gain [15],

$$E(T[\perp_j]) - \frac{|T[v]|}{|T[\perp_j]|}E(T[v]) - \frac{|T[\perp_j] - T_v|}{|T[\perp_j]|}E(T[\perp_j] - T_v). \tag{2}$$

$E(R)$ measures the entropy or impurity of a set of records R wrt the specified class attribute and $InfoGain(v)$ measures the reduction of entropy after the disclosure of v. (See Quinlan [15] for the definition of $E(R)$.) The important point is that $InfoGain(v)$ depends only on the class frequency and count statistics of the single attribute $att(v)$ in $T[\perp_j]$, $T[v]$ and $T[\perp_j] - T_v$.

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

$$avg\{Conf_v(QID \rightarrow \pi) - Conf(QID \rightarrow \pi) \mid att(v) \in QID\}, \tag{3}$$

where $Conf$ and $Conf_v$ represent the confidence before and after disclosing v. 1 is added to $PrivLoss(v)$ to avoid division by zero.

Computing $Conf_v$ efficiently is a challenge because it involves count statistics on combinations of attributes. It is inefficient to actually perform the disclosure of v just to compute $Conf_v$ because performing disclosures involves record scans. The key to the scalability of our algorithm is incrementally updating $Score(v)$ in each iteration using the statistics collected during performing the winner disclosure w. We will present this update algorithm in Sect. 4.3.

4.2 Disclose the winner (Line 5)

To disclose the winner w, we replace $\perp_j$ with w in the suppressed records in $T[\perp_j]$ that originally contain w. So, we need to access the raw records that originally contain w. The following data structure facilitates the direct access to all the raw records affected by this disclosure. The general idea is to partition raw records according to their suppressed records on the set of attributes $\cup QID$.

Definition 4.1 (VIP) Value Indexed Partitions (VIP) *contains the set of suppressed records over* $\cup QID$. *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* $\cup QID$, $P[x]$ *denotes a partition represented by a suppressed record containing* x. $Link[x]$ *links up all partitions* $P[x]$*s, with the head stored with the value* x.

$Link[x]$ provides a direct access to all raw records that those suppressed records contain the value x. Let $\perp_w$ denote the special value $\perp$ for the attribute of the winner w. To disclose w, we follow $Link[\perp_w]$ and find all suppressed records that contain $\perp_w$, and through these suppressed records, access the represented raw records. So, we do not have to scan unaffected data records.

Disclose w in VIP: For each partition $P[\perp_w]$ on $Link[\perp_w]$ and its suppressed record r, create a new suppressed record r' as a copy of r except that $\perp_w$ is replaced with w, create the partition $P[w]$ for r' to contains all raw records in $P[\perp_w]$ that contain w, and remove such records from $P[\perp_w]$. Link all new $P[w]$s by the new $Link[w]$, and relink them to the links to which $P[\perp_w]$ is currently linked, except for $Link[\perp_w]$. Finally, remove w from Sup_j.

Since one "relinking" operation is required for each masking attribute M_j and each new partition, there are at most $m \times |Link[\perp_w]|$ "relinking" operations in total for disclosing w, where m is the number of masking attributes and $|Link[\perp_w]|$ is the length of $Link[\perp_w]$. This overhead of maintaining $Link[x]$ is negligible. The following example illustrates the procedure of disclosing w in VIP.

Example 4 Consider the templates in Example 3. In Fig. 1, the left-most VIP has the most suppressed record $\langle \perp_{\text{Job}}, \perp_{\text{Country}}, \perp_{\text{Child}} \rangle$ on three links:

$$Link[\perp_{\text{Job}}],\ Link[\perp_{\text{Country}}],\ Link[\perp_{\text{Child}}].$$

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

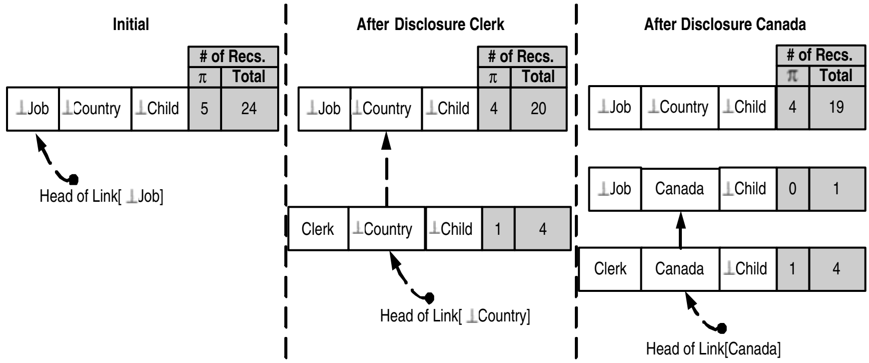


Fig. 1 Evolution of VIP ($\pi = Discharged$)

Count statistics in VIP: To update $Score(x)$ efficiently, we maintain the following *count statistics* for each partition P in the VIP: for every class θ and sensitive property π, (1) $|P|$, $f(P, \theta)$ and $f(P, \pi)$, (2) for each masking attribute M_j on which P has the value $\perp_j$, for every suppressed value x in Sup_j, $f(P, x)$, $f(P, \{x, \theta\})$ and $f(P, \{x, \pi\})$. 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. Subsequently, updating $Score(x)$ makes use of the count statistics in the VIP without accessing raw records.

4.3 Update score and status (Line 6)

This step updates $Score(x)$ and the valid/beneficial status for candidates x in $\cup Sup_j$. $Score(x)$ is defined by $InfoGain(x)$ and $PrivLoss(x)$. $InfoGain(x)$ is affected only if x and w are from the same attribute, in other words, $x \in Sup_w$, where Sup_w denotes Sup_j for the attribute of w. To update $InfoGain(x)$, we compute

$$s(x) = \sum f(P, x),$$

$$s(x, \theta) = \sum f(P, \{x, \theta\}),$$

$$s(\perp_w) = \sum |P|,$$

$$s(\perp_w, \theta) = \sum f(P, \theta),$$

over the partitions P on $Link[\perp_w]$. These information can be computed in the same scan as collecting 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 $QID \to \pi$, we first update $Conf(QID \to \pi)$ using $Conf_w(QID \to \pi)$ that was computed in the previous iteration. Next, we update $Conf_x(QID \to \pi)$ for x in $\cup Sup_j$. We need to update $Conf_x(QID \to \pi)$

only if both $att(x)$ and $att(w)$ are contained in *QID*. We propose the following *QID*-tree structure to maintain $Conf(QID \rightarrow \pi)$.

Definition 4.2 (*QID*-**trees**) *For each* $QID = \{A_1, \ldots, A_u\}$, *the QID-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 qid on QID in the suppressed* T, *with* $s(qid)$ *and* $s(qid, \pi)$ *stored at the leaf node.*

Recall that $conf(qid \rightarrow \pi) = s(qid, \pi)/s(qid)$ and that $Conf(QID \rightarrow \pi)$ is $max\{conf(qid \rightarrow \pi)\}$ for all *qid* in the *QID*-tree. If several templates $\langle QID \rightarrow \pi, h\rangle$ have the same *QID*, they can share a single *QID*-tree by keeping $s(qid, \pi)$ separately for different π.

Update *QID*-**trees:** On disclosing w, we update all the *QID*-trees such that $att(w) \in QID$ to reflect the move of records from $Link[\perp_w]$ to $Link[w]$. First, for each leaf node representing $\{qid, \perp_w\}$, we create a new root-to-leaf node representing the new $\{qid, w\}$. Then, for each partition P on $Link[w]$, if $\{qid, w\}$ is the value on *QID*, update the *QID*-tree as follows:

$$\begin{aligned} s(qid, w) &= s(qid, w) + |P| \\ s(qid, \perp_w) &= s(qid, \perp_w) - |P| \\ s(qid, w, \pi) &= s(qid, w, \pi) + f(P, \pi) \\ s(qid, \perp_w, \pi) &= s(qid, \perp_w, \pi) - f(P, \pi). \end{aligned}$$

This involves one scan of the link $Link[w]$ because $|P|$ and $f(P, \pi)$ are kept with the Ps on this link. Here is an example.

Example 5 Figure 2 shows the initial QID_1-tree and QID_2-tree on the left, where $QID_1 = \{\text{Job}, \text{Country}\}$ and $QID_2 = \{\text{Job}, \text{Child}\}$. On disclosing *Clerk*, $\{Clerk, \perp_{\text{Country}}\}$ and $\{Clerk, \perp_{\text{Child}}\}$ are created in QID_1-tree and QID_2-tree. Next, on disclosing *Canada*, $\{Clerk, \perp_{\text{Country}}\}$ is refined into $\{Clerk, Canada\}$ in QID_1-tree, and a new $\{\perp_{\text{Job}}, Canada\}$ is split from $\{\perp_{\text{Job}}, \perp_{\text{Country}}\}$. For example, to compute $s(qid)$ and $s(qid, \pi)$ for the new $qid = (\perp_{\text{Job}}, Canada)$, we

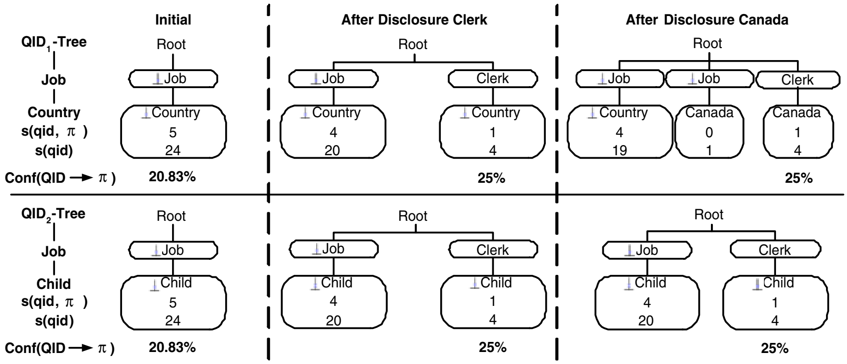


Fig. 2 Evolution of *QID*-trees

access all partitions $P[Canada]$ in one scan of $Link[Canada]$:

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

The resulting counts are shown on the right most *QID*-trees.

Update $Conf_x(QID \to \pi)$**:** On disclosing w, for $x \in \cup Sup_j$, we update $Conf_x(QID \to \pi)$ only if both $att(x)$ and $att(w)$ are in *QID*. Recall that $Conf_x(QID \to \pi)$ is the maximum $conf(qid \to \pi)$ *after* disclosing x. Therefore, we can treat x *as if* it *were* disclosed, and computing $s(qid, x)$, $s(qid, x, \pi)$, $s(qid, \perp_x)$ and $s(qid, \perp_x, \pi)$ as we did for w. We now follow $Link[\perp_x]$ instead of $Link[w]$. Since we just compute $Conf_x(QID \to \pi)$, not performing the disclosure of x, we do not update the VIP for x, but just make use of the count statistics in category (2) to compute $s(qid, x)$, $s(qid, x, \pi)$, $s(qid, \perp_x)$ and $s(qid, \perp_x, \pi)$. The computation is on a *copy* of the *QID*-trees because we do not actually disclose x on the *QID*-trees. $Conf_x(QID \to \pi)$ is the new maximum $conf(qid \to \pi)$ in the copy *QID*-tree. If $Conf_x(QID \to \pi) \leq h$, mark x as "valid".

4.4 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 privacy templates. We also evaluated the efficiency of this method. We adopted three widely used benchmarks: *Japanese Credit Screening* and *Adult* were obtained from the UCI repository [14]. *German Credit Data* was obtained from Silicon Graphics, Inc.[1] We removed all continuous attributes since our current implementation focuses on categorical attributes. We used the C4.5 classifier [15] for classification modeling. Other classifiers, such as SVM [18], may produce lower classification error than the C4.5 does; however, our focus is not on comparing different classifiers. All experiments were conducted on an Intel Pentium IV 3GHz PC with 1GB RAM.

Templates. For each data set, we conducted two sets of experiments, which differ in the choice of sensitive attributes $\Pi_1, \ldots, \Pi_N$ and masking attributes $M_1, \ldots, M_m$.

[1] http://www.sgi.com/tech/mlc/db/

- TopN: We chose the "best" N attributes, denoted TopN, as sensitive attributes $\Pi_1, \ldots, \Pi_N$. 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. Simply removing $\Pi_1, \ldots, \Pi_N$ will compromise the classification. The remaining attributes were chosen as the masking attributes $M_1, \ldots, M_m$. For each Π_i, we choose the 50% least frequent values as sensitive properties. The rationale is that less frequent properties are more vulnerable to inference attacks. Let $\{\pi_1, \ldots, \pi_k\}$ denote the union of such properties for all Π_i. The set of templates is $\{\langle QID \rightarrow \pi_i, h\rangle \mid 1 \leq i \leq k\}$, or written simply as $\langle QID \rightarrow \pi_1, \ldots, \pi_k, h\rangle$, where QID contains all masking attributes. From Theorem 3.1, this set of templates is more restrictive than a set of templates with each being a subset of QID (for the same threshold h).
- RanN: In this experiment, we randomly selected N attributes, denoted RanN, as sensitive attributes $\Pi_1, \ldots, \Pi_N$, and selected all remaining attributes as masking attributes. Once $\Pi_1, \ldots, \Pi_N$ are selected, the template $\langle QID \rightarrow \{\pi_1, \ldots, \pi_k\}, h\rangle$ is constructed as explained above. We report the average result for 30 privacy templates generated this way.

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. *SE* and *RE* depend on the privacy template, whereas *BE* does not. 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 nine 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., $A_1 \cdots A_{15}$. We used all the nine categorical attributes.

We consider the four template requirements: Top1, Top2, Top3 and Top4. Top4 attributes are A_9, A_{10}, A_7, A_6 in that order. $BE = 15.4\%$. Table 3 shows the

Table 3 Number of inferences above h

Threshold h	10%	30%	50%	70%	90%
CRX (Top4)	40	27	15	8	6
Adult (Top4)	1333	786	365	324	318
German (Top6)	496	337	174	162	161

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.

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. *Small* $SE - BE$. *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 *QID*s share some common values, and suppressing a few common values simultaneously eliminates multiple inferences. Our method biases to suppress such common values because *PrivLoss* in *Score* function minimizes the *average* increase of *Conf* on *all* templates.
2. *Large* $RE - SE$. 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. *Small variance of SE*. 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 suppression is performed on the masking attributes, but at the same time, more sensitive attributes can be used for classification.
4. *Larger benefits for larger* N. 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 Fig. 3a.

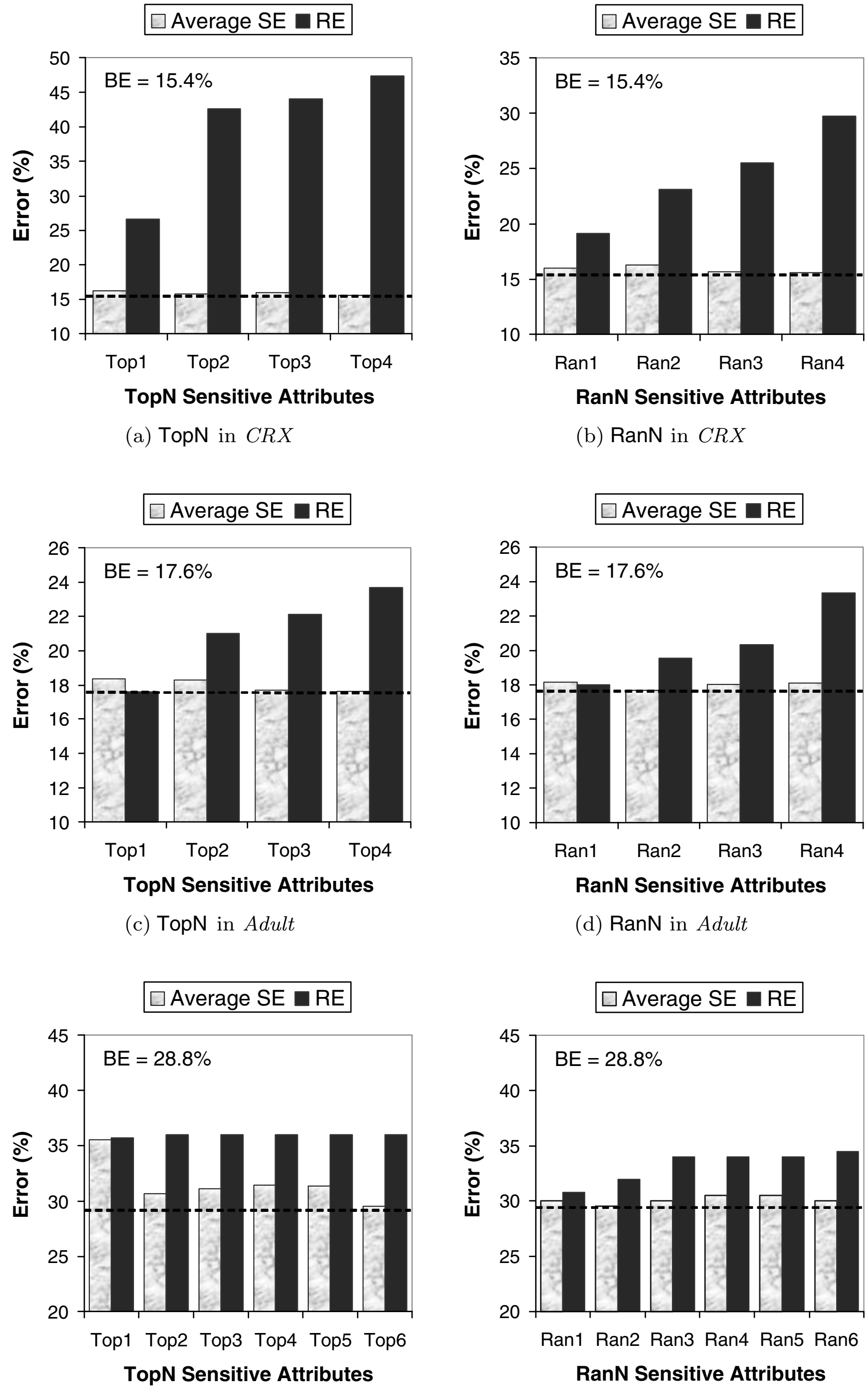
Let us take a closer look at the suppressed data for Top4 with $h = 70\%$. Some values of attributes A_4 and A_5 are suppressed, and the entire A_{13} is suppressed. Despite such vigorous suppression, $SE = 15.4\%$ is equal to *BE*. In fact, there exist multiple classification structures in the data. 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.

Figure 3b depicts *SE* on 30 sets of RanN, averaged over the same h as in the previous experiment. Again, *SE* spans narrowly between 15.4% and 16.5%, i.e., no more than 1.1% above *BE*. *RE* for RanN is lower than *RE* for TopN because some randomly selected sensitive attributes are not important and their removal has less impact on classification.

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

5.2 Adult

The *Adult* data set is a census data previously used in [2, 8, 9, 21]. There are eight categorical attributes and a binary class attribute representing the income levels ≤50 K or >50 K. 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. $BE = 17.6\%$.



(a) TopN in *CRX*

(b) RanN in *CRX*

(c) TopN in *Adult*

(d) RanN in *Adult*

(e) TopN in *German*

(f) RanN in *German*

Fig. 3 Classification error

Table 4 Attributes for the *Adult* data set

Attribute	No. of values	Attribute	No. of values
Education (E)	16	Marital-status (M)	7
Occupation (O)	14	Native-country (Nc)	40
Race (Ra)	5	Relationship (Re)	6
Sex (S)	2	Work-class (W)	8

Figure 3c 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 and Top4), RE picked up.

Figure 3d depicts a similar result for the 30 sets of RanN, but with lower REs. The experiments on both TopN and RanN strongly suggest that the suppression approach preserves the quality of classification consistently for various privacy templates. Our algorithm spent at most 14 s for all experiments on *Adult*, of which approximately 10 s were spent on suppressing the 45,222 data records.

5.3 German credit data

The *German Credit Data*, or simply *German*, has 13 categorical attributes and a binary class attribute representing the *good* or *bad* credit risks. There are 666 and 334 records, without missing values, for the pre-split training and testing respectively. Table 5 describes each categorical attribute. The Top6 attributes in *German* are A, Ch, Sa, I, Lp, D in that order. $BE = 28.8\%$. Like the *Adult* data, *German* also has many sensitive inferences as shown in Table 3.

Figure 3e shows the SE and RE averaged over $h = 30\%, 50\%, 70\%, 90\%$. The benefit $RE - SE$ is approximately 4.3% on average. Interestingly, RE almost stays flat at 36% for Top1 to Top6. To explain this, we looked into the data set and found that the Top2 attributes, i.e., A and Ch, play a dominant role in modeling

Table 5 Attributes for the *German* data set

Attribute	No. of values	Attribute	No. of values
Account-status (A)	4	Property (Pr)	4
Credit-history (Ch)	5	Installments (I)	3
Loan-purpose (Lp)	11	Housing (H)	3
Savings-account (Sa)	5	Job (J)	4
Employment (Em)	5	Telephone (T)	2
Personal-status (Ps)	5	Foreign (F)	2
Debtors (D)	3		

the class attribute. Removing any one (or both) of them increases the error by approximately 7% comparing with *BE*. Thus, after removing the top one attribute, removing the next top five attributes does not degrade the classification quality much.

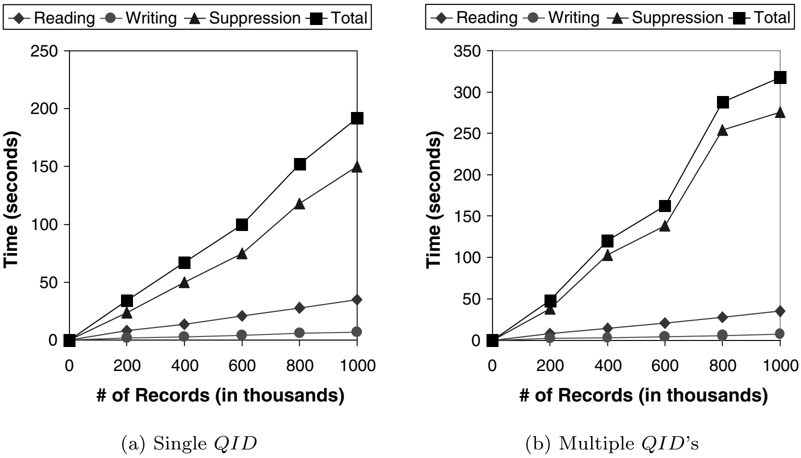
SE stays close to *RE* for Top1 and then drops to approximately 31% for Top2 to Top6. This 5% drop of *SE* from Top1 to Top2 is due to the fact that many values of the second top attribute Ch are suppressed in Top1, but the top two attributes A and Ch are not suppressed in Top2.

Figure 3f depicts the results for 30 sets of RanN. Unlike the TopN case, *RE* increases gradually with respect to the number N of sensitive attributes. This is because the importance of A and Ch has been averaged out in these 30 randomly constructed templates. Our algorithm spent less than 3 s for all experiments conducted on *German*.

5.4 Scalability

The key to scalability of our method is maintaining count statistics instead of scanning raw data records. 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 $\cup QID$, 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 4a depicts the runtime of our suppression method for 200 K to 1 M data records based on the templates $\langle QID \rightarrow \{\pi^1, \ldots, \pi^k\}, 90\%\rangle$, where the set of sensitive properties $\{\pi^1, \ldots, \pi^k\}$ is the set of 50% least frequent values



(a) Single QID (b) Multiple QID's

Fig. 4 Scalability ($h = 90\%$)

in the Top1 attribute M, and QID contains the other seven attributes. This is one of the most time consuming settings in the case of single QID 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 s to suppress 1 M records, of which 150 s were spent on suppression, and the rest was spent on disk I/O operations. We also tried $h = 100\%$. Our method took a total of 296 s to disclose all values due to the increased number of partitions and number of QIDs. However, this is not a typical case because typically we want to eliminate inferences with a confidence higher than some h that is below 100%.

We further extended the scalability experiment to privacy templates that have multiple QIDs. The number of different QIDs determines the number of QID-trees, and more QIDs means more maintenance cost of QID-trees. We determined the number of QIDs by uniformly and randomly drawing a number between 3 and 6, and the length of QID between 2 and 5. For each QID, we randomly selected the attributes from the seven remaining attributes, and discarded the repeating ones. All QIDs in the same set of privacy templates have the same length and same threshold $h = 90\%$. For example, a set of privacy templates having three QIDs of length 2 is

$$\begin{aligned}\{\langle\{E, Nc\} \rightarrow \{\pi^1, \ldots, \pi^k\}, 90\%\rangle,\\ \langle\{E, O\} \rightarrow \{\pi^1, \ldots, \pi^k\}, 90\%\rangle,\\ \langle\{Ra, W\} \rightarrow \{\pi^1, \ldots, \pi^k\}, 90\%\rangle\}.\end{aligned}$$

$\{\pi^1, \ldots, \pi^k\}$ is the same as above.

Figure 4b depicts the average runtime over 30 sets of privacy templates generated as described above. Our method spent 318 s to suppress 1 M records. Out of the 318 s, 276 s were spent on suppression. With $h = 100\%$, our method spent 412 s on suppression. Compared to the case of a single QID, more time was required for a requirement with multiple QIDs because it has to maintain one QID-tree for each distinct QID.

6 Extensions

To bring out the main ideas, our current implementation has assumed that the table fits in memory. Often, this assumption is valid because the table can be first compressed by removing irrelevant attributes and collapsing duplicates (as in Table 1). If the table does not fit in memory, we can keep the VIP in the memory but store the data partitions on disk. We can also use the memory to keep those partitions smaller than the page size to avoid page fragmentation. In addition, partitions that cannot be further refined can be discarded and only some statistics for them need to be kept. This likely applies to the small partitions kept in memory, therefore, the memory demand is unlikely to build up.

So far, we have considered only categorical attributes. Our approach can be extended to suppress continuous values by the means of *discretization*. For example, we can replace specific age values from 51 to 55 with a less specific interval

[51–55]. This method does not require a priori discretized taxonomy for a continuous attribute, but dynamically obtains one in the top-down disclosure process. Initially, all domain values of a continuous attribute are represented by a single interval v covering the whole range, and Sup_j contains v. At each iteration, a disclosure for an interval v refers to splitting the interval into two sub-intervals v_1 and v_2, with the splitting point being chosen to maximize *InfoGain*. Next, v is replaced by v_1 and v_2 in Sup_j forming a new set of candidates for the next disclosure. The criterion for choosing the interval for splitting is exactly same as that for choosing a suppressed value for disclosure. This process repeats until no disclosure is possible without violating the set of privacy templates. To extend Theorem 3.2 (therefore, Corollary 3.2) to cover $QID \rightarrow \pi$ in which QID contains continuous attributes as well, we can replace the disclosure $\perp'_j \rightarrow \{\perp_j, v\}$ with $v \rightarrow \{v_1, v_2\}$ in the proof, and the rest requires little changes.

We have considered classification as the use of the released data where the information gain wrt the class attribute is used as the information utility *InfoGain*. Our approach can be extended to other information utility by substituting *InfoGain* with a proper measure. For example, if the goal is to minimize the "syntax distortion" to the data [16], we can regard each suppression of a domain value v in a record as one unit of distortion and define $InfoGain(v)$ to be the number of records that contain v. The rest of the algorithm requires little changes.

7 Conclusions

We studied the problem of eliminating the sensitive inferences that are made possible by data mining abilities, while preserving the classification value of the data. A sensitive inference has a high confidence in linking a group of individuals to sensitive properties. 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 top-down 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.

Acknowledgements 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.

References

1. Agrawal R, Imielinski T, Swami A (1993) Mining associations between sets of items in massive databases. In: Proceedings of the ACM SIGMOD international conference on management of data, Washington, DC, pp 207–216
2. Bayardo R, Agrawal R (2005) Data privacy through optimal *k*-anonymization. In: Proceedings of the 21st IEEE internaional conference on data engineering (ICDE'05), Tokyo, Japan, pp 217–228
3. Clifton C (2000) Using sample size to limit exposure to data mining. J Comput Secur 8(4):281–307

4. Clifton C, Kantarcioglu M, Vaidya J, Lin X, Zhu MY (2002) Tools for privacy preserving data mining. SIGKDD Explorat 4(2):28–34
5. Cox LH (1980) Suppression methodology and statistical disclosure control. J Am Stat Assoc, Theory Method Sect 75:377–385
6. Evfimievski A, Srikant R, Agrawal R, Gehrke J (2002) Privacy preserving mining of association rules. In: Proceedings of the 8th ACM SIGKDD, Edmonton, Alberta, Canada, pp 217–228
7. Farkas C, Jajodia S (2003) The inference problem: a survey. SIGKDD Explorat 4(2):6–11
8. Fung BCM, Wang K, Yu PS (2005) Top-down specialization for information and privacy preservation. In: Proceedings of the 21st IEEE internaional conference on data engineering (ICDE'05), Tokyo, Japan, pp 205–216
9. Iyengar VS (2002) Transforming data to satisfy privacy constraints. In: Proceedings of the 8th ACM SIGKDD, Edmonton, Alberta, Canada, pp 279–288
10. Kantarcioglu M, Jin J, Clifton C (2004) When do data mining results violate privacy? In: Proceedings of the 10th ACM SIGKDD, Seattle, WA, USA, pp 599–604
11. Kim J, Winkler W (1995) Masking microdata files. In: ASA proceedings of the section on survey research methods
12. Kloesgen W (1995) Knowledge discovery in databases and data privacy. In: Proceedings of the IEEE expert symposium on knowledge discovery in databases
13. Machanavajjhala A, Gehrke J, Kifer D (2006) l-diversity: privacy beyond k-anonymity. In: Proceedings of the 22nd IEEE internaional conference on data engineering (ICDE'06), Atlanta, GA, USA
14. Newman DJ, Hettich, S, Blake CL, Merz CJ (1998) UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA, http://www.ics.uci.edu/∼mlearn/MLRepository.html
15. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann
16. Samarati P, Sweeney L (1998) Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report SRI-CSL-98-04, SRI Computer Science Laboratory
17. Sweeney L (2006) Datafly: a system for providing anonymity in medical data. In: Proceedings of the 11th international conference on database security, pp 356–381
18. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
19. Verykios VS, Elmagarmid AK, Bertino E, Saygin Y, Dasseni E (2004) Association rule hiding. IEEE Trans Knowledge Data Eng 16(4):434–447
20. Wang K, Fung BCM, Yu PS (2005) Template-based privacy preservation in classification problems. In: Proceedings of the 5th IEEE international conference on data mining (ICDM'05), Houston, TX, USA, pp 466–473
21. Wang K, Yu PS, Chakraborty S (2004) Bottom-up generalization: a data mining solution to privacy protection. In: Proceedings of the 4th IEEE international conference on data mining (ICDM'04), Brighton, UK, pp 249–256
22. Yip RW, Levitt KN (1999) Bottom-up generalization: a data mining solution to privacy protection. In: Proceedings of the 12th international working conference on database security XII, pp 253–266

Author Biographies

Ke Wang received Ph.D. from Georgia Institute of Technology. He is currently a professor at School of Computing Science, Simon Fraser University. Before joining Simon Fraser, he was an associate professor at National University of Singapore. He has taught in the areas of database and data mining. Dr. Wang's research interests include database technology, data mining and knowledge discovery, machine learning, and emerging applications, with recent interests focusing on the end use of data mining. This includes explicitly modeling the business goal (such as profit mining, bio-mining and web mining) and exploiting user prior knowledge (such as extracting unexpected patterns and actionable knowledge). He is interested in combining the strengths of various fields such as database, statistics, machine learning and optimization to provide actionable solutions to real-life problems. He is an associate editor of the IEEE TKDE journal and has served program committees for international conferences.

Benjamin C. M. Fung received B.Sc. and M.Sc. degrees in computing science from Simon Fraser University. Received the postgraduate scholarship doctoral award from the Natural Sciences and Engineering Research Council of Canada (NSERC), Mr. Fung is currently a Ph.D. candidate at Simon Fraser. His recent research interests include privacy-preserving data mining, secure distributed computing, and text mining. Before pursuing his Ph.D., he worked in the R&D Department at Business Objects and designed reporting systems for various Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems, including BaaN, Siebel, and PeopleSoft. Mr. Fung has published in data engineering, data mining, and security conferences, journals, and books, including IEEE ICDE, IEEE ICDM, IEEE ISI, SDM, KAIS, and the Encyclopedia of Data Warehousing and Mining.

Philip S. Yu received B.S. degree in E.E. from National Taiwan University, M.S. and Ph.D. degrees in E.E. from Stanford University, and M.B.A. degree from New York University. He is with IBM T.J. Watson Research Center and currently manager of the Software Tools and Techniques group. Dr. Yu has published more than 450 papers in refereed journals and conferences. He holds or has applied for more than 250 US patents. Dr. Yu is a Fellow of the ACM and the IEEE. He has received several IBM honors including two IBM Outstanding Innovation Awards, an Outstanding Technical Achievement Award, two Research Division Awards and the 85th plateau of Invention Achievement Awards. He received a Research Contributions Award from IEEE International Conference on Data Mining in 2003 and also an IEEE Region 1 Award for "promoting and perpetuating numerous new electrical engineering concepts" in 1999. Dr. Yu is an IBM Master Inventor.