Anonymizing Classification Data for Privacy Preservation

Benjamin C.M. Fung, Ke Wang, and Philip S. Yu, *Fellow*, *IEEE*

Abstract—Classification is a fundamental problem in data analysis. Training a classifier requires accessing a large collection of data. Releasing person-specific data, such as customer data or patient records, may pose a threat to an individual's privacy. Even after removing explicit identifying information such as Name and SSN, it is still possible to link released records back to their identities by matching some combination of nonidentifying attributes such as $\{Sex, Zip, Birthdate\}$. A useful approach to combat such linking attacks, called k-anonymization [1], is anonymizing the linking attributes so that at least k released records match each value combination of the linking attributes. Previous work attempted to find an optimal k-anonymization that minimizes some data distortion metric. We argue that minimizing the distortion to the training data is not relevant to the classification goal that requires extracting the *structure* of predication on the "future" data. In this paper, we propose a k-anonymization solution for classification. Our goal is to find a k-anonymization, not necessarily optimal in the sense of minimizing data distortion, which preserves the classification structure. We conducted intensive experiments to evaluate the impact of anonymization on the classification on future data. Experiments on real-life data show that the quality of classification can be preserved even for highly restrictive anonymity requirements.

Index Terms—Privacy protection, anonymity, security, integrity, data mining, classification, data sharing.

1 Introduction

Data sharing in today's globally networked systems poses a threat to individual privacy and organizational confidentiality. An example by Samarati [2] shows that linking medication records with a voter list can uniquely identify a person's name and medical information. New privacy acts and legislations are recently enforced in many countries. In 2001, Canada launched the *Personal Information Protection and Electronic Document Act* [3] to protect a wide spectrum of information, such as age, race, income, evaluations, and even intentions to acquire goods or services. This information spans a considerable portion of many databases. Government agencies and companies have to revise their systems and practices to fully comply with this act in three years.

Consider a table T about a patient's information on $Birthplace$, $Birthyear$, Sex, and $Diagnosis$. If a description on $\{Birthplace, Birthyear, Sex\}$ is so specific that not many people match it, releasing the table may lead to linking a unique record to an external record with explicit identity, thus identifying the medical condition and compromising the privacy rights of the individual [2]. Suppose that the attributes $Birthplace$, $Birthyear$, Sex, and $Diagnosis$ must be released (say, to some health research institute for research purposes). One way to prevent such linking is *masking* the detailed information of these attributes as follows:

1. If there is a taxonomical description for a categorical attribute (for example, $Birthplace$), we can *generalize* a specific value description into a less specific but semantically consistent description. For example, we can generalize the cities San Francisco, San Diego, and Berkeley into the corresponding state California.
2. If there is no taxonomical description for a categorical attribute, we can *suppress* a value description to a "null value" denoted $\perp$. For example, we can suppress San Francisco and San Diego to the null value $\perp$ while keeping Berkeley.
3. If the attribute is a continuous attribute (for example, Birthyear), we can *discretize* the range of the attribute into a small number of intervals. For example, we can replace specific Birthyear values from 1961 to 1965 with an interval [1961-1966).

By applying such masking operations, the information on $\{Birthplace, Birthyear, Sex\}$ is made less specific, and a person tends to match more records. For example, a male born in San Francisco in 1962 will match all records that have the values $\langle \mathrm{CA}, [1961-1966), \mathrm{M}\rangle$; clearly, not all matched records correspond to the person. Thus, the masking operation makes it more difficult to tell whether an individual actually has the diagnosis in the matched records.

Protecting privacy is one goal. Making the released data useful to data analysis is another goal. In this paper, we consider classification analysis [4]. The next example shows that if masking is performed "carefully," privacy can be protected while preserving the usefulness for classification.

Example 1 (The running example). Consider the data in Table 1 and the taxonomy trees in Fig. 1. The table has 34 records in total. Each row represents one or more records with the $Class$ column containing the class frequency of the records represented, Y for "income > 50 thousand" and N for "income $\leq$ 50 thousand." For

- B.C.M. Fung and K. Wang are with the School of Computer Science, Simon Fraser University, Burnaby, BC Canada V5A 1S6. E-mail: {bfung, wangk}@cs.sfu.ca.
- P.S. Yu is with the IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532. E-mail: psyu@us.ibm.com.

Manuscript received 7 May 2005; revised 22 Nov. 2006; accepted 11 Dec. 2006; published online 24 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0173-0505.
Digital Object Identifier no. 10.1109/TKDE.2007.1015.

TABLE 1
(Compressed) Table

Education	Sex	Work_Hrs	Class	# of Recs.
9th	M	30	0Y3N	3
10th	M	32	0Y4N	4
11th	M	35	2Y3N	5
12th	F	37	3Y1N	4
Bachelors	F	42	4Y2N	6
Bachelors	F	44	4Y0N	4
Masters	M	44	4Y0N	4
Masters	F	44	3Y0N	3
Doctorate	F	44	1Y0N	1
		Total:	21Y13N	34

example, the third row represents five records having $Education = 11th$, $Sex = Male$, and $Work\_Hrs = 35$. The value 2Y3N in the Class column conveys that two records have the class Y and three records have the class N. Semantically, this (compressed) table is equivalent to the table containing 34 rows with each row representing one record. There is only one record for "female doctor" (the last row), which makes the person represented uniquely distinguishable from others by Sex and $Education$. To make "female doctor" less unique, we can generalize $Masters$ and $Doctorate$ to $GradSchool$. As a result, "she" becomes less distinguishable by being one of the four females with a graduate school degree. As far as classification is concerned, no information is lost in this generalization because $Class$ does not depend on the distinction of $Masters$ and $Doctorate$.

In the classification problem, a classifier is built from the *training data* and is used to classify the *future data*. It is important that the classifier makes use of the *structure* that will repeat in the future data, not the *noises* that occur only in the training data. In Table 1, 19 out of 22 persons having $Work\_Hrs \geq 37$ are in the class Y, and only three persons having $Work\_Hrs \geq 37$ are in the class N. It is not likely that this difference is entirely due to sampling noises. In contrast, M and F of Sex seem to be arbitrarily associated with both classes, suggesting that sex cannot be used to predict his/her class.

In this paper, we consider the following *k-anonymization for classification*. The data provider wants to release a person-specific table for modeling classification of a specified class attribute in the table. Two types of information in the table are released. The first type is *sensitive information*, such as $Diagnosis$. The second type is the *quasi-identifier* (QID) [5], [1], which is a combination of attributes such as $\{Birthplace, Birthyear, Sex\}$. The QID does not identify individuals but can be used to link to a person if the combination is unique. The data provider wants to prevent linking the released records to an individual through the QID. This privacy requirement is specified by the k-anonymity [1]: If one record in the table has some value on the QID, then at least $k-1$ other records have that value. The k-anonymization for classification is to produce a masked table that satisfies the k-anonymity requirement and retains useful information for classification. A formal statement will be given in Section 2.

If classification is the goal, why does not the data provider build and publish a classifier (instead of publishing the data)? There are real-life scenarios where it is necessary to release the data. First of all, knowing that the data is used for classification does not imply that the data provider knows exactly *how* the recipient may analyze the data. The recipient often has application-specific bias towards building the classifier. For example, some recipient prefers accuracy, whereas the others prefer interpretability; or some prefers recall, whereas the others prefer precision, and so on. In other cases, the recipient may not know exactly what to do before seeing the data, such as visual data mining, where the human makes decisions based on certain distributions of data records at each step. Publishing the data provides the recipient a greater flexibility of data analysis.

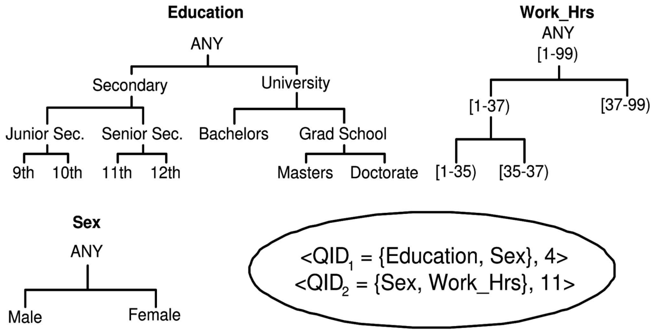


Fig. 1. Taxonomy trees and QIDs.

Our insight is as follows: Typically, the data contains overly specific "noises" that are harmful to classification. To construct a classifier, noises need to be generalized into patterns that are shared by more records in the same class. The data also contains "redundant structures." For example, if any of $Education$ and $Work\_Hrs$ is sufficient for determining the class and if one of them is distorted, the class can still be determined from the other attribute. Our approach exploits such rooms provided by noises and redundant structures to mask the data without compromising the quality of classification. To this end, we propose an information metric to focus masking operations on the noises and redundant structures. We conducted intensive experiments to evaluate the impact of anonymization on the classification of future data. Below are several useful features of our approach.

- *Information and privacy guided top-down refinement (TDR).* Instead of masking the most specific table bottom-up, we refine masked values top-down starting from the most masked table.
- *Handling different types of attributes.* We handle categorical attributes with taxonomy, categorical attributes without taxonomy, and continuous attributes.
- *Handling multiple QIDs.* Compared to the single QID that contains all the attributes, we enforce k-anonymity on only attribute sets that can be potentially used as a QID. This approach avoids unnecessary distortion to the data.
- *Scalable and anytime solution.* Our method has a linear time complexity in the table size. Moreover, the user can stop the TDR *any time* and have a table satisfying the anonymity requirement.

The notion of k-anonymity was first proposed in [1]. In general, a cost metric is used to measure the data distortion of anonymization. Two types of cost metric have been considered. The first type, based on the notion of minimal generalization [2], [6], is independent of the purpose of the data release. The second type factors in the purpose of the data release such as classification [7]. The goal is to find the optimal k-anonymization that minimizes this cost metric. In general, achieving optimal k-anonymization is NP-hard [8], [9]. Greedy methods were proposed in [10], [11], [12], [13], [14], and [15]. Scalable algorithms (with the exponential complexity, the worst case) for finding the optimal k-anonymization were studied in [2], [6], [7], and [16].

Our insight is that the optimal k-anonymization is not suitable to classification where masking structures and masking noises have different effects: the former deems to damage classification, whereas the latter helps classification. It is well known in data mining and machine learning that the unmodified data, which has the lowest possible cost according to any cost metric, often has a worse classification than some generalized (that is, masked) data. In a similar spirit, less masked data could have a worse classification than some more masked data. This observation was confirmed by our experiments. The optimal k-anonymization seeks to minimize the error on the training data, thus overfits the data, subject to the privacy constraint. Neither the overfitting nor the privacy constraint is relevant to the classification goal that seeks to minimize the error on future data.

Besides the standard setting, extensions of k-anonymity were also studied. LeFevre et al. [17] proposed the notion of multidimensional k-anonymity, where data generalization is over multidimension-at-a-time, and LeFevre et al. [18] extended multidimensional generalization to anonymize data for a specific task such as classification. Xu et al. [19] proposed some greedy methods to achieve k-anonymity with cell generalization and showed that the cell generalization generally causes less information loss than the multidimensional generalization. These masking operations allow the coexistence of a specific value and a general value, such as $Bachelor$ and $University$ in Table 1. Such masked data will suffer from "interpretation difficulty" in the data analysis phase. For example, the exact number of bachelors cannot be determined when only some, say, only three out of the 10 $Bachelor$s are generalized to $Universities$. If a classifier is built from such data, it is unclear which classification rule, $Bachelor \rightarrow Y$ or $University \rightarrow N$, should be used to classify bachelor.

Machanavajjhala [20] measured anonymity by the l-diversity that corresponds to some notion of uncertainty of linking a QID to a particular sensitive value. Wang et al. [21], [22] proposed to bound the confidence of inferring a particular sensitive value using one or more privacy templates specified by the data provider. Wong et al. [15] proposed some generalization methods to simultaneously achieve k-anonymity and bound the confidence. Xiao and Tao [23] limited the breach probability, which is similar to the notion of confidence, and allowed a flexible threshold for each individual. k-anonymization for data owned by multiple parties was considered in [24]. k-anonymization for sequential releases was studied in [25].

2 Problem Definition

A data provider wants to release a person-specific table $T(D_1, \ldots, D_m, Class)$ to the public for modeling the class label $Class$. Each D_i is either a categorical or a continuous attribute. A record has the form $\langle v_1, \ldots, v_m, cls \rangle$, where v_i is a domain value for D_i and cls is a class for $Class$. $att(v)$ denotes the attribute of a value v. The data provider also wants to protect against linking an individual to sensitive information either within or outside T through some *identifying* attributes, called QID. A sensitive linking occurs if some value of the QID identifies a "small" number of records in T. This requirement is formally defined below.

Definition 1 (Anonymity requirement). *Consider p QIDs $QID_1, \ldots, QID_p$ on T. $a(qid_i)$ denotes the number of data records in T that share the value qid_i on QID_i. The* anonymity *of QID_i, denoted $A(QID_i)$, is the smallest $a(qid_i)$ for any value qid_i on QID_i. A table T satisfies the* anonymity requirement *$\{\langle QID_1, k_1 \rangle, \ldots, \langle QID_p, k_p \rangle\}$ if $A(QID_i) \geq k_i$ for $1 \leq i \leq p$, where k_i is the* anonymity threshold *on QID_i specified by the data provider.*

It is not hard to see that if QID_j is a subset of QID_i, $A(QID_i) \leq A(QID_j)$. Therefore, if $k_j \leq k_i$, $A(QID_i) \geq k_i$ implies $A(QID_j) \geq k_j$, and $\langle QID_j, k_j \rangle$ can be removed in the presence of $\langle QID_i, k_i \rangle$. Following a similar argument, to prevent a linking through any QID, that is, *any* subset of attributes in $QID_1 \cup \cdots \cup QID_p$, the single QID requirement $\langle QID, k \rangle$, where $QID = QID_1 \cup \cdots \cup QID_p$ and $k = max\{k_j\}$, can be specified. However, a table satisfying $\{\langle QID_1, k_1 \rangle, \ldots, \langle QID_p, k_p \rangle\}$ does not have to satisfy $\langle QID, k \rangle$.

Example 2. Suppose that a data provider wants to release Table 1. To protect linking $\{Education, Sex\}$ to sensitive information, the data provider specifies $\langle QID_1 = \{Education, Sex\}, 4 \rangle$. This requirement is violated by $\langle 9th, M \rangle$, $\langle Masters, F \rangle$, and $\langle Doctorate, F \rangle$. To protect linking through $\{Sex, Work\_Hrs\}$ as well, the data provider can specify the two QIDs in Fig. 1. To prevent linking through any combination of the identifying attributes, the data provider can specify $QID = \{Education, Sex, Work\_Hrs\}$.

Definition 1 generalizes the classic notion of k-anonymity by allowing multiple QIDs (with possibly different thresholds). Suppose that the data provider wants to release a table $T(A, B, C, D, S)$, where S is the sensitive attribute and knows that the recipient has access to previously released tables $T1(A, B, X)$ and $T2(C, D, Y)$, where X and Y are attributes not in T. To prevent linking the records in T to X or Y, the data provider only has to specify the k-anonymity on $QID1 = \{A, B\}$ and $QID2 = \{C, D\}$. In this case, enforcing the k-anonymity on $QID = \{A, B, C, D\}$ will distort the data more than what is necessary. All previous works suffer from this problem because they handled multiple QIDs through the single QID made up of all attributes in the multiple QIDs.

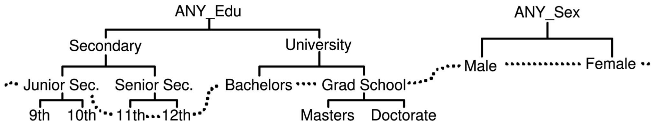


Fig. 2. A cut for $QID_1 = \{Education, Sex\}$.

To transform T to satisfy the anonymity requirement, we consider three types of masking operations on the attributes D_j in $\cup QID_i$.

Masking Operations:

1. *Generalize* D_j if D_j is a categorical attribute with a taxonomy tree. A leaf node represents a domain value and a parent node represents a less specific value. Fig. 2 shows a taxonomy tree for *Education*. A generalized D_j can be viewed as a "cut" through its taxonomy tree. A *cut* of a tree is a subset of values in the tree, denoted Cut_j, which contains exactly one value on each root-to-leaf path. This type of generalization does not suffer from the interpretation difficulty discussed in Section 1, and it was previously employed in [7], [12], [13], [14], [24], and [25].
2. *Suppress* D_j if D_j is a categorical attribute with no taxonomy tree. The *suppression* of a value on D_j means replacing *all* occurrences of the value with the special value $\perp_j$. All suppressed values on D_j are represented by the same $\perp_j$, which is treated as a new value in D_j by a classification algorithm. We use Sup_j to denote the set of values suppressed by $\perp_j$. This type of suppression is at the value level in that Sup_j is, in general, a subset of the values in the attribute D_j.
3. *Discretize* D_j if D_j is a continuous attribute. The discretization of a value v on D_j means replacing *all* occurrences of v with an interval containing the value. Our algorithm dynamically grows a taxonomy tree for intervals at runtime, where each node represents an interval, and each nonleaf node has two child nodes representing some "optimal" binary split of the parent interval. More details will be discussed in Section 3. A discretized D_j can be represented by the set of intervals, denoted Int_j, corresponding to the leaf nodes in the dynamically grown taxonomy tree of D_j.

Definition 2 (Anonymity for Classification). *Given a table T, an anonymity requirement $\{\langle QID_1, k_1\rangle, \ldots, \langle QID_p, k_p\rangle\}$ and an optional taxonomy tree for each categorical attribute contained in $\cup QID_i$ mask T on the attributes $\cup QID_i$ to satisfy the anonymity requirement while preserving the classification structure in the data (that is, the masked table remains useful for classifying the Class column).*

The cost metric for our anonymization should be measured by the classification error on the future data. It does not work to replace this cost metric by the classification error on the masked table because a perfect classifier for the masked table (say, a classifier based on a system-assigned record ID) can be inaccurate for the future data. For this reason, our problem does not have a closed-form cost metric, and an "optimal" solution to our problem is not necessarily an optimal k-anonymization based on a closed-form cost metric, and vice versa. Therefore, the previous optimal k-anonymization approaches [7], [16] based on a closed-form cost metric are not suitable. A more reasonable approach is minimally, not always optimally, masking the data, with focus on classification. We will present such an approach in Section 3.

It is impractical to enumerate all masked tables because the number of masked tables can be very large. For a categorical attribute with a taxonomy tree Y, the number of possible cuts, denoted $C(Y)$, is equal to $C(Y_1) \times \ldots \times C(Y_u) + 1$, where $Y_1, \ldots, Y_u$ are the subtrees rooted at the children of the root of Y, and 1 is for the trivial cut at the root of Y. $C(Y)$ increases very quickly as we unfold the definition for each subtree Y_i recursively. For a categorical attribute without a taxonomy tree and with q distinct values, there are 2^q possible suppressions because each distinct value can be either suppressed or not. For a continuous attribute, each existing value can be a potential split in the dynamically grown taxonomy tree. The number of possible masked tables is equal to the product of such numbers for all the attributes in $\cup QID_i$.

A masked table T can be represented by

$$\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle,$$

where Cut_j, Sup_j, and Int_j are defined as above. If the masked T satisfies the anonymity requirement, $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ is called a *solution set*.

3 Search Criteria

A table T can be masked by a sequence of refinements starting from the most masked state in which each attribute is either generalized to the topmost value, suppressed to the special value $\perp$, or represented by a single interval. Our method iteratively refines a masked value selected from the current set of cuts, suppressed values, and intervals, until violating the anonymity requirement. Each refinement increases the information and decreases the anonymity since records with specific values are more distinguishable. The key is selecting the "best" refinement at each step with both impacts considered.

3.1 Refinement

Below, we formally describe the notion of refinement on different types of attributes $D_j \in \cup QID_i$ and define a selection criterion for a single refinement.

3.1.1 Refinement for Generalization

Consider a categorical attribute D_j with a user-specified taxonomy tree. Let $child(v)$ be the set of child values of v in a user-specified taxonomy tree. A *refinement*, written $v \rightarrow child(v)$, replaces the parent value v with the child value in $child(v)$ that generalizes the domain value in each (generalized) record that contains v.

3.1.2 Refinement for Suppression

For a categorical attribute D_j without taxonomy tree, a refinement $\perp_j \rightarrow \{v, \perp_j\}$ refers to disclosing one value v from the set of suppressed values Sup_j. Let $R_{\perp_j}$ denote the set of suppressed records that currently contain $\perp_j$. Disclosing v means replacing $\perp_j$ with v in all records in $R_{\perp_j}$ that originally contain v.

3.1.3 Refinement for Discretization

For a continuous attribute, refinement is similar to that for generalization except that no prior taxonomy tree is given and the taxonomy tree has to be grown dynamically in the process of refinement. Initially, the interval that covers the full range of the attribute forms the root. The refinement on an interval v, which is written as $v \rightarrow child(v)$, refers to the optimal split of v into two child intervals $child(v)$ that maximizes the information gain. The anonymity is not used for finding a split good for classification. This is similar to defining a taxonomy tree where the main consideration is how the taxonomy best describes the application. Due to this extra step of identifying the optimal split of the parent interval, we treat continuous attributes separately from categorical attributes with taxonomy trees.

A refinement is *valid* (with respect to T) if T satisfies the anonymity requirement after the refinement. A refinement is *beneficial* (with respect to T) if more than one class is involved in the refined records. A refinement is performed only if it is both valid and beneficial. Therefore, a refinement guarantees that every newly generated *qid* has $a(qid) \geq k$.

Example 3. Continue with Example 2. Fig. 2 shows a cut, indicated by the dashed curve. This cut is the lowest (maximal) in the sense that any refinement on *Junior Sec.* or *Grad School* would violate the anonymity requirement, that is, invalid. Also, refinement on *Junior Sec.* or *Grad School* is nonbeneficial since none of them refines data records in different classes.

3.2 Selection Criterion

We propose a selection criterion for guiding our TDR process to heuristically maximize the classification goal. Consider a refinement $v \rightarrow child(v)$, where $v \in D_j$, and D_j is a categorical attribute with a user-specified taxonomy tree or D_j is a continuous attribute with a dynamically grown taxonomy tree. The refinement has two effects: it increases the information of the refined records with respect to classification, and it decreases the anonymity of the refined records with respect to privacy. These effects are measured by "information gain," denoted $InfoGain(v)$, and "anonymity loss," denoted $AnonyLoss(v)$. v is a good candidate for refinement if $InfoGain(v)$ is large and $AnonyLoss(v)$ is small. Our selection criterion is choosing the candidate v, for the next refinement, that has the maximum *information-gain/anonymity-loss trade-off*, which is defined as

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

To avoid division by zero, 1 is added to $AnonyLoss(v)$. Each choice of $InfoGain(v)$ and $AnonyLoss(v)$ gives a trade-off between classification and anonymization. It should be noted that $Score$ is not a goodness metric of k-anonymization. In fact, it is difficult to have a closed-form metric to capture the classification goal (on future data). We achieve this goal through this heuristic selection criterion.

For concreteness, we borrow Shannon's information theory to measure information gain [26]. Let R_v denote the set of records masked to the value v, and let R_c denote the set of records masked to a child value c in $child(v)$ after refining v. Let $|x|$ be the number of elements in a set x. $|R_v| = \sum_c |R_c|$, where $c \in child(v)$.

InfoGain(v): Defined as

$$InfoGain(v) = I(R_v) - \sum_c \frac{|R_c|}{|R_v|} I(R_c), \tag{2}$$

where $I(R_x)$ is the *entropy* of R_x [26]:

$$I(R_x) = -\sum_{cls} \frac{freq(R_x, cls)}{|R_x|} \times log_2 \frac{freq(R_x, cls)}{|R_x|}. \tag{3}$$

$freq(R_x, cls)$ is the number of data records in R_x having the class cls. Intuitively, $I(R_x)$ measures the entropy (or "impurity") of classes in R_x. The more dominating the majority class in R_x is, the smaller the $I(R_x)$ becomes (that is, less entropy in R_x). Therefore, $I(R_x)$ measures the error because nonmajority classes are considered as errors. $InfoGain(v)$ then measures the reduction of entropy after refining v. $InfoGain(v)$ is nonnegative. For more details on information gain and classification, see [27].

AnonyLoss(v): Defined as

$$AnonyLoss(v) = avg\{A(QID_j) - A_v(QID_j)\}, \tag{4}$$

where $A(QID_j)$ and $A_v(QID_j)$ represent the anonymity before and after refining v. $avg\{A(QID_j) - A_v(QID_j)\}$ is the average loss of anonymity for all QID_j that contain the attribute of v.

If D_j is a categorical attribute without taxonomy tree, the refinement $\perp_j \rightarrow \{v, \perp_j\}$ means refining $R_{\perp_j}$ into R_v and $R'_{\perp_j}$, where $R_{\perp_j}$ denotes the set of records containing $\perp_j$ before the refinement. R_v and $R'_{\perp_j}$ denote the set of records containing v and $\perp_j$ after the refinement, respectively. We employ the same $Score(v)$ function to measure the goodness of the refinement $\perp_j \rightarrow \{v, \perp_j\}$, except that $InfoGain(v)$ is now defined as:

$$InfoGain(v) = I(R_{\perp_j}) - \frac{|R_v|}{|R_{\perp_j}|} I(R_v) - \frac{|R'_{\perp_j}|}{|R_{\perp_j}|} I(R'_{\perp_j}). \tag{5}$$

Example 4. The refinement on ANY_Edu refines the 34 records into 16 records for $Secondary$ and 18 records for $University$. The calculation of $Score(ANY\_Edu)$ is

TABLE 2
Comparing $InfoGain$ and $Score$ for Example 5

(a)

Education	Sex	Work_Hrs	Class	# of Recs.
10th	M	40	20Y0N	20
10th	M	30	0Y4N	4
9th	M	30	0Y2N	2
9th	F	30	0Y4N	4
9th	F	40	0Y6N	6
8th	F	30	0Y2N	2
8th	F	40	0Y2N	2
		Total:	20Y20N	40

Candidate	InfoGain	AnonyLoss	Score
ANY_Edu	0.6100	$40 - 4 = 36$	$0.6100/(36 + 1) = 0.0165$
ANY_Sex	0.4934	$40 - 14 = 26$	$0.4934/(26 + 1) = 0.0183$
[1-99)	0.3958	$40 - 12 = 28$	$0.3958/(28 + 1) = 0.0136$

(b)

(c)

Education	Sex	Work_Hrs	Class	# of Recs.
10th	ANY_Sex	[1-99)	20Y4N	24
9th	ANY_Sex	[1-99)	0Y12N	12
8th	ANY_Sex	[1-99)	0Y4N	4

(d)

Education	Sex	Work_Hrs	Class	# of Recs.
ANY_Edu	M	[1-99)	20Y6N	26
ANY_Edu	F	[1-99)	0Y14N	14

(e)

Education	Sex	Work_Hrs	Class	# of Recs.
ANY_Edu	M	[40-99)	20Y0N	20
ANY_Edu	M	[1-40)	0Y6N	6
ANY_Edu	F	[40-99)	0Y8N	8
ANY_Edu	F	[1-40)	0Y6N	6

(a) (Compressed) table. (b) Statistics for the most masked table. (c) Final masked table by $InfoGain$. (d) Intermediate masked table by $Score$. (e) Final Masked Table by $Score$.

$$I(R_{ANY\_Edu}) = -\frac{21}{34} \times log_2 \frac{21}{34} - \frac{13}{34} \times log_2 \frac{13}{34} = 0.9597,$$

$$I(R_{Secondary}) = -\frac{5}{16} \times log_2 \frac{5}{16} - \frac{11}{16} \times log_2 \frac{11}{16} = 0.8960,$$

$$I(R_{University}) = -\frac{16}{18} \times log_2 \frac{16}{18} - \frac{2}{18} \times log_2 \frac{2}{18} = 0.5033,$$

$$InfoGain(ANY\_Edu) = I(R_{ANY\_Edu})$$

$$-\left(\frac{16}{34} \times I(R_{Secondary}) + \frac{18}{34} \times I(R_{University})\right) = 0.2716,$$

$$AnonyLoss(ANY\_Edu) = (34 - 16)/1 = 18,$$

$$Score(ANY\_Edu) = \frac{0.2716}{18 + 1} = 0.0143.$$

3.3 $InfoGain$ versus $Score$

An alternative to $Score$ is using $InfoGain$ alone, that is, maximizing the information gain produced by a refinement without considering the loss of anonymity. This alternative may pick a candidate that has a large reduction in anonymity, which may lead to a quick violation of the anonymity requirement, thereby, prohibiting refining the data to a lower granularity. The next example illustrates this point.

Example 5. Consider Table 2a, the anonymity requirement

$$\langle QID = \{Education, Sex, Work\_Hrs\}, 4\rangle,$$

the most masked table containing one row

$$\langle ANY\_Edu, ANY\_Sex, [1 - 99)\rangle$$

with the class frequency 20Y20N and three candidate refinements:

$$ANY\_Edu \rightarrow \{8th, 9th, 10th\}, \quad ANY\_Sex \rightarrow \{M, F\},$$
$$\text{and } [1 - 99) \rightarrow \{[1 - 40), [40 - 99)\}.$$

Table 2b shows the calculated $InfoGain$, $AnonyLoss$, and $Score$ of the three candidate refinements. According to the $InfoGain$ criterion, ANY_Edu will be first refined because it has the highest $InfoGain$. The result is shown in Table 2c with $A(QID) = 4$. After that, there is no further valid refinement because refining either ANY_Sex or [1-99) will result in a violation of 4-anonymity. Note that the first 24 records in the table fail to separate the 4N from the other 20Y.

In contrast, according to the $Score$ criterion, ANY_Sex will be first refined. The result is shown in Table 2d, and $A(QID) = 14$. Subsequently, further refinement on ANY_Edu is invalid because it will result in $a(\langle 9th, M, [1 - 99)\rangle) = 2 < k$, but the refinement on [1-99) is valid because it will result in $A(QID) = 6 \geq k$. The final masked table is shown in Table 2e where the information for separating the two classes is preserved. Thus, by considering the information/anonymity trade-off, the $Score$ criterion produces a more desirable sequence of refinements for classification.

4 TDR

4.1 The Algorithm

We present our algorithm *TDR*. In a preprocessing step, we compress the given table T by removing all attributes not in $\cup QID_i$ and collapsing duplicates into a single row with the $Class$ column storing the class frequency as in Table 1. The compressed table is typically much smaller than the original table. Below, the term "data records" refers to data records in this compressed form. There exists a masked table satisfying the anonymity requirement if and only if the most masked table does, that is, $|T| \geq k$. This condition is checked in the preprocessing step as well. To focus on main ideas, we assume that $|T| \geq k$ and the compressed table fits in the memory. In Section 4.5, we will discuss the modification needed if the compressed table does not fit in the memory.

Algorithm 1. Top-Down Refinement (TDR)

```
Initialize every value of $D_j$ to the topmost value,
   suppress every value of $D_j$ to $\perp_j$, or include every
   continuous value of $D_j$ into the full-range interval,
   where $D_j \in \cup QID_i$.
Initialize $Cut_j$ of $D_j$ to include the topmost value, $Sup_j$
   of $D_j$ to include all domain values of $D_j$, and $Int_j$ of $D_j$
   to include the full-range interval, where $D_j \in \cup QID_i$.
while some $x \in \langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ is valid and
   beneficial
     Find the $Best$ refinement from $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
     Perform $Best$ on $T$ and update
        $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
     Update $Score(x)$ and validity for
        $x \in \langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
end while
return Masked $T$ and $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
```

High-level description of our algorithm. Algorithm 1 summarizes the conceptual algorithm. Initially, Cut_j contains only the topmost value for a categorical attribute D_j with a taxonomy tree, Sup_j contains all domain values of a categorical attribute D_j without a taxonomy tree, and Int_j contains the full-range interval for a continuous attribute D_j. The valid beneficial refinements in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ form the set of *candidates*. At each iteration, we find the candidate of the highest $Score$, denoted $Best$ (Line 4), apply $Best$ to T, update $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ (Line 5), and update $Score$ and the validity of the candidates in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ (Line 6). The algorithm terminates when there is no more candidate in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$, in which case it returns the masked table together with the solution set $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.

Example 6. Consider the anonymity requirement:

$$\{\langle QID_1 = \{Education, Sex\}, 4\rangle,$$
$$\langle QID_2 = \{Sex, Work\_Hrs\}, 11\rangle\}.$$

Assume that the taxonomy trees in Fig. 1 are specified for $Education$ and Sex. Initially, all data records in Table 1 are masked and collapsed into a single row $\langle ANY\_Edu, ANY\_Sex, [1-99)\rangle$, with the class frequency 21Y13N and $\cup Cut_j = \{ANY\_Edu, ANY\_Sex\}$, $\cup Int_j = \{[1-99)\}$. All refinements in $\langle \cup Cut_j, \cup Int_j \rangle$ are candidates. To find the best refinement, we need to compute $Score(ANY\_Edu)$, $Score(ANY\_Sex)$, and $Score([1-99))$.

Our algorithm obtains the masked T by iteratively refining the table from the most masked state. An important property of TDR is that the anonymity requirement is *antimonotone* with respect to the TDR: If it is violated before a refinement, it remains violated after the refinement. This is because a refinement never equates distinct values; therefore, it never increases the count of duplicates $a(qid)$. Hence, the hierarchically organized search space with the most masked state at the top is separated by a border above which lie all satisfying states and below which lie all violating states. The TDR finds a state on the border, and this state is *maximally refined* in that any further refinement of it would cross the border and violate the anonymity requirement. Note that there may be more than one maximally refined state on the border. Our algorithm finds the one based on the heuristic selection criterion of maximizing $Score$ at each step. Samarati [2] presents some results related to antimonotonicity, but the results are based on a different masking model that generalizes all values in an attribute to the same level and suppresses data at the record level.

Theorem 1. *Algorithm 1 finds a maximally refined table that satisfies the given anonymity requirement.*

Algorithm 1 makes no claim on efficiency. In fact, in a straightforward implementation, Lines 4, 5, and 6 require scanning all data records and recomputing $Score$ for all candidates in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$. Obviously, this is not scalable. The key to the efficiency of our algorithm is *directly* accessing the data records to be refined and updating $Score$ based on some statistics maintained for candidates in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$. In the rest of this section, we explain a scalable implementation of Lines 4, 5, and 6.

4.2 Find the Best Refinement (Line 4)

This step makes use of computed $InfoGain(x)$ and $A_x(QID_i)$ for all candidates x in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ and computed $A(QID_i)$ for each QID_i. Before the first iteration, such information is computed in an initialization step for every topmost value, every suppressed value, and every full-range interval. For each subsequent iteration, such information comes from the update in the previous iteration (Line 6). Finding the best refinement $Best$ involves at most $|\cup Cut_j| + |\cup Sup_j| + |\cup Int_j|$ computations of $Score$ without accessing data records. Updating $InfoGain(x)$ and $A_x(QID_i)$ will be considered in Section 4.4.

4.3 Perform the Best Refinement (Line 5)

We consider two cases of performing the $Best$ refinement, corresponding to whether a taxonomy tree is available for the attribute D_j for $Best$.

Case 1: D_j **has a taxonomy tree**. Consider the refinement $Best \rightarrow child(Best)$, where $Best \in D_j$ and D_j is either a categorical attribute with a specified taxonomy tree or a continuous attribute with a dynamically grown taxonomy tree. First, we replace $Best$ with $child(Best)$ in $\langle \cup Cut_j, \cup Int_j \rangle$. Then, we need to retrieve R_{Best}, the set of data records masked to $Best$, to tell the child value in $child(Best)$ for each individual data record. We present a data structure *Taxonomy Indexed PartitionS (TIPS)* to facilitate this operation. This data structure is also crucial for updating $InfoGain(x)$ and $A_x(QID_i)$ for candidates x. The general idea is to group data records according to their masked records on $\cup QID_i$.

Definition 3 (TIPS). *TIPS is a tree structure with each node representing a masked record over $\cup QID_i$ and each child node representing a refinement of the parent node on exactly one attribute. Stored with each leaf node is the set of (compressed) data records having the same masked record, called a* leaf partition. *For each candidate refinement x, P_x denotes a leaf partition whose masked record contains x, and $Link_x$ denotes the link of all such P_x. The head of $Link_x$ is stored with x.*

The masked table is represented by the leaf partitions of TIPS. $Link_x$ provides a direct access to R_x, the set of (original) data records masked by the value x. Initially, TIPS has only one leaf partition containing all data records, masked by the topmost value or interval on every attribute in $\cup QID_i$. In each iteration, we perform the best refinement $Best$ by refining the leaf partitions on $Link_{Best}$.

Refine $Best$ **in TIPS.** We refine each leaf partition P_{Best} found on $Link_{Best}$ as follows: For each value c in $child(Best)$, a child partition P_c is created under P_{Best}, and data records in P_{Best} are split among the child partitions: P_c contains a data record in P_{Best} if a categorical value c generalizes the corresponding domain value in the record or if an interval c contains the corresponding domain value in the record, an empty P_c is removed. $Link_c$ is created to link up all P_cs for the same c. Also, link P_c to every $Link_x$ to which P_{Best} was previously linked, except for $Link_{Best}$.

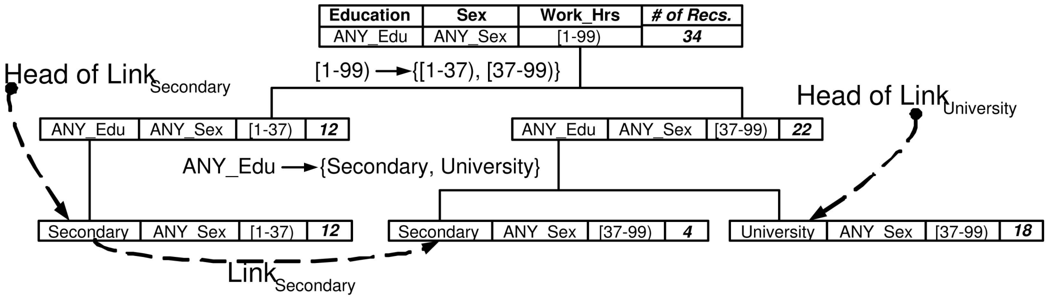

Education | Sex | Work_Hrs | # of Recs.
ANY_Edu | ANY_Sex | [1-99) | 34
[1-99) → {[1-37), [37-99)}
Head of Link Secondary
Head of Link University
ANY_Edu | ANY_Sex | [1-37) | 12
ANY_Edu | ANY_Sex | [37-99) | 22
ANY_Edu → {Secondary, University}
Secondary | ANY_Sex | [1-37) | 12
Secondary | ANY_Sex | [37-99) | 4
University | ANY_Sex | [37-99) | 18
Link Secondary

Fig. 3. The TIPS data structure.

Finally, mark c as "beneficial" if R_c has more than one class, where R_c denotes the set of data records masked to c.

This is the only operation that actually accesses data records in the whole algorithm. The overhead is maintaining $Link_x$. For each attribute in $\cup QID_i$ and each leaf partition on $Link_{Best}$, there are at most $|child(Best)|$ "relinkings." Therefore, there are at most $|\cup QID_i| \times |Link_{Best}| \times |child(Best)|$ "relinkings" for applying $Best$.

Example 7. Continue with Example 6. Initially, TIPS has only one leaf partition containing all data records and representing the masked record

$$\langle ANY\_Edu, ANY\_Sex, [1-99)\rangle.$$

Let the best refinement be $[1-99) \rightarrow \{[1-37), [37-99)\}$ on $Work\_Hrs$. We create two child partitions under the root partition as in Fig. 3 and split data records between them. Both child partitions are on $Link_{ANY\_Edu}$ and $Link_{ANY\_Sex}$. $\cup Int_j$ is updated into $\{[1-37), [37-99)\}$ and $\cup Cut_j$ remains unchanged. Suppose that the next best refinement is $ANY\_Edu \rightarrow \{Secondary, University\}$, which refines the two leaf partitions on $Link_{ANY\_Edu}$, resulting in the TIPS in Fig. 3.

Count statistics in TIPS. A scalable feature of our algorithm is maintaining some statistical information for each candidate x in $\langle \cup Cut_j, \cup Int_j\rangle$ for updating $Score(x)$ without accessing data records. For each value c in $child(Best)$ added to $\langle \cup Cut_j, \cup Int_j\rangle$ in the current iteration, we collect the following *count statistics* of c while scanning data records in P_{Best} for updating TIPS: 1) $|R_c|$, $|R_d|$, $freq(R_c, cls)$, and $freq(R_d, cls)$ for computing $InfoGain(c)$, where $d \in child(c)$ and cls is a class label. Refer to Section 3 for these notations. 2) $|P_d|$, where P_d is a child partition under P_c *as if* c is refined, kept together with the leaf node for P_c. This information will be used in Section 4.4.

TIPS has several useful properties: 1) All data records in the same leaf partition have the same masked record, although they may have different refined values. 2) Every data record appears in exactly one leaf partition. 3) Each leaf partition P_x has exactly one masked qid_j on QID_j and contributes the count $|P_x|$ towards $a(qid_j)$. Later, we use the last property to extract $a(qid_j)$ from TIPS.

Case 2: D_j has no taxonomy tree. Consider a refinement $\perp_j \rightarrow \{Best, \perp_j\}$, where $\perp_j \in D_j$, and D_j is a categorical attribute without a taxonomy tree. First, we remove $Best$ from Sup_j. Then, we replace $\perp_j$ with the disclosed value $Best$ in all suppressed records that currently contain $\perp_j$ and originally contain $Best$. The TIPS data structure in Definition 3 can also support the refinement operation in this case. The only difference is to add an extra $Link_{\perp_j}$ to link up all leaf partitions $P_{\perp_j}$ containing value $\perp_j$. The candidate set now includes $\cup Sup_j$, that is, $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$.

Disclose $Best$ in TIPS. We refine each leaf partition $P_{\perp_j}$ found on $Link_{\perp_j}$ as follows: Two child partitions P_{Best} and $P'_{\perp_j}$ are created under $P_{\perp_j}$. Data records in $P_{\perp_j}$ are split among the child partitions: P_{Best} contains a data record r in $P_{\perp_j}$ if $Best$ is the original domain value in r; otherwise, $P'_{\perp_j}$ contains r. Then, link P_{Best} to every $Link_x$ to which $P_{\perp_j}$ was previously linked, except for $Link_{\perp_j}$. Also, link $P'_{\perp_j}$ to every $Link_x$ to which $P_{\perp_j}$ was previously linked, except for $Link_{Best}$.

Count statistics in TIPS. Similar to Case 1, we collect the following count statistics of $x \in \cup Sup_j$ while scanning data records in $P_{\perp_j}$ for updating TIPS: 1) $|R'_{\perp_j}|$, $|R_x|$, $freq(R'_{\perp_j}, cls)$, and $freq(R_x, cls)$ for computing $InfoGain(x)$, where $x \in \cup Sup_j$ and cls is a class label. 2) $|P_y|$, where P_y is a child partition under P_x *as if* x is disclosed, kept together with the leaf node for P_x. This information will be used in Section 4.4.

4.4 Update Score and Validity (Line 6)

This step updates $Score(x)$ and validity for candidates x in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$ to reflect the impact of the $Best$ refinement. The key is computing $Score(x)$ from the count statistics maintained in Section 4.3 without accessing data records. We update $InfoGain(x)$ and $A_x(QID_i)$ separately. Note that the updated $A(QID_i)$ is obtained from $A_{Best}(QID_i)$.

4.4.1 Update $InfoGain(x)$

An observation is that $InfoGain(x)$ is not affected by $Best \rightarrow child(Best)$, except that we need to compute $InfoGain(c)$ for each newly added value c in $child(Best)$. $InfoGain(c)$ can be computed while collecting the count statistics for c in Case 1 of Section 4.3. In case the refined attribute has no taxonomy tree, $InfoGain(x)$ can be computed from the count statistics for x in Case 2 of Section 4.3.

4.4.2 Update $AnonyLoss(x)$

Again, we consider the two cases:

Case 1: D_j has a taxonomy tree. Unlike information gain, it is not enough to compute $A_c(QID_i)$ only for the new values c in $child(Best)$. Recall that $A_x(QID_i)$ is equal to the minimum $a(qid_i)$ after refining x. If both $att(x)$ and $att(Best)$ are contained in QID_i, the refinement on $Best$ may affect this minimum, hence, $A_x(QID_i)$. Below, we

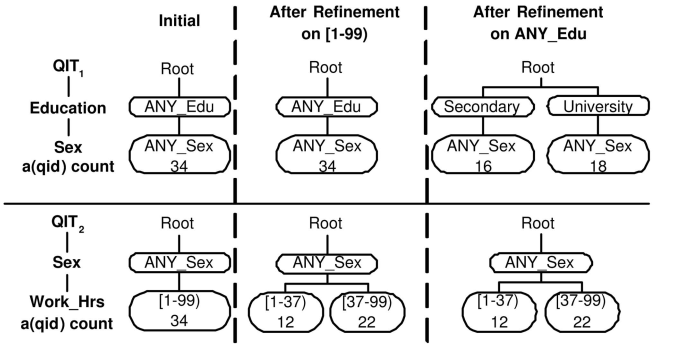


Fig. 4. The QITS data structure.

present the data structure *Quasi-Identifier TreeS* ($QITS$) to extract $a(qid_i)$ efficiently from TIPS for updating $A_x(QID_i)$.

Definition 4 (QITS). *QIT_i for $QID_i = \{D_1, \ldots, D_w\}$ is a tree of w levels. The level $p > 0$ represents the masked values for D_p. Each root-to-leaf path represents an existing qid_i on QID_i in the masked data, with $a(qid_i)$ stored at the leaf node. A branch is trimmed if its $a(qid_i) = 0$.*

$A(QID_i)$ is equal to the minimum $a(qid_i)$ in QIT_i. In other words, QIT_i provides an index of $a(qid_i)$ by qid_i. Unlike TIPS, QITS does not maintain data records. On applying $Best \rightarrow child(Best)$, we update every QIT_i such that QID_i contains the attribute $att(Best)$.

Update QIT_i. For each occurrence of $Best$ in QIT_i, create a separate branch for each c in $child(Best)$. The procedure in Algorithm 2 computes $a(qid_i)$ for the newly created qid_is on such branches. The general idea is to loop through each P_c on $Link_c$ in TIPS, increment $a(qid_i)$ by $|P_c|$. This step does not access data records because $|P_c|$ was part of the count statistics of $Best$. Let r be the number of QID_i containing $att(Best)$. The number of $a(qid_i)$ to be computed is at most $r \times |Link_{Best}| \times |child(Best)|$.

Algorithm 2. Computing $a(qid_i)$ for new qid_i

```
for each $P_c \in Link_c$ do
    for each $QID_i$ containing $att(Best)$ do
        $a(qid_i) = a(qid_i) + |P_c|$, where $qid_i$ is the masked
        value on $QID_i$ for $P_c$
    end for
end for
```

Example 8. In Fig. 4, the initial QIT_1 and QIT_2 (that is, left most) have a single path. After applying

$$[1-99) \rightarrow \{[1-37), [37-99)\},$$

the qid $\langle ANY\_Sex, [1-99)\rangle$ in QIT_2 is replaced with two new qids $\langle ANY\_Sex, [1-37)\rangle$ and $\langle ANY\_Sex, [37-99)\rangle$ and $A(QID_2) = 12$. Since QID_1 does not include $Work\_Hrs$, QIT_1 remains unchanged and $A(QID_1) = 34$.

After applying the second refinement

$$ANY\_Edu \rightarrow \{Secondary, University\},$$

QIT_2 remains unchanged, and $A(QID_2) = 12$. The qid $\langle ANY\_Edu, ANY\_Sex\rangle$ in QIT_1 is replaced with two new qids $\langle Secondary, ANY\_Sex\rangle$ and $\langle University, ANY\_Sex\rangle$. To compute $a(qid)$ for these new qids, we add $|P_{Secondary}|$ to $Link_{Secondary}$ and $|P_{University}|$ to $Link_{University}$ (see Fig. 3): $a(\langle Secondary, ANY\_Sex\rangle) = 0 + 12 + 4 = 16$, and

$$a(\langle University, ANY\_Sex\rangle) = 0 + 18 = 18.$$

So, $A_{ANY\_Edu}(QID_1) = 16$.

We now update $A_x(QID_i)$ for candidates x in

$$\langle \cup Cut_j, \cup Int_j \rangle$$

(in the impact of $Best \rightarrow child(Best)$). Doing this by refining x requires accessing data records, hence, it is not scalable. We compute $A_x(QID_i)$ using the count statistics maintained for x without accessing data records.

Update $A_x(QID_i)$. For a candidate x in $\langle \cup Cut_j, \cup Int_j\rangle$, computing $A_x(QID_i)$ is necessary in two cases: First, x is in $child(Best)$ because $A_x(QID_i)$ has not been computed for newly added candidates x. Second, $A_x(QID_i)$ might be affected by the refinement on $Best$, in which case, $att(x)$ and $att(Best)$ must be contained in QID_i. In both cases, we first compute $a(qid_i^x)$ for the new qid_i^xs created *as if* x is refined. The procedure is the same as in Algorithm 2 for refining $Best$, except that $Best$ is replaced with x and no actual update is performed on QIT_i and TIPS. Note that the count $|P_c|$, where c is in $child(x)$, used in the procedure is part of the count statistics maintained for x.

Next, we compare $a(qid_i^x)$ with $A(QID_i)$ to determine the minimum, that is, $A_x(QID_i)$. There are two subcases:

1. If no contributing qid of $A(QID_i)$ (that is, $a(qid) = A(QID_i)$) contains the value x, the contributing qids of $A(QID_i)$ will remain existing if x is refined. Hence, $A_x(QID_i)$ is the minimum of $A(QID_i)$ and $a(qid_i^x)$.
2. If some contributing qid of $A(QID_i)$ contains the value x, such qids become new qid_i^x if x is refined, so $A_x(QID_i)$ is the minimum of $a(qid_i^x)$.

Finally, if the new $A_x(QID_i) \geq k_i$, we keep it with x and mark x as "valid" in the cut.

Case 2: D_j has no taxonomy tree. Even the refined attribute has no taxonomy tree, the general operation of computing $AnonyLoss(x)$ is the same as Case 1. The difference is that the refined values of $\perp_j$ becomes $\{Best, \perp_j\}$, where $Best$ is the disclosed value and the updated $\perp_j$ represents the remaining suppressed values Sup_j. Also, the candidate set includes $\cup Sup_j$, that is, $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$. On disclosing $Best$, we update all QIT_i such that $att(Best)$ is in QIT_i to reflect the move of records from $Link_{\perp_j}$ to $Link_{Best}$.

Update QIT_i. For each occurrence of $\perp_j$ in QIT_i, create a separate branch for $Best$ and a separate branch for updated $\perp_j$. Follow the procedure in Algorithm 2 to compute $a(qid_i)$ for the newly created qid_is on such branches, except that P_cs become P_{Best} and $P'_{\perp_j}$. Refer to Case 2 in Section 4.3 for these notations.

4.5 Efficiency Analysis

Each iteration involves two types of work. The first type accesses data records in R_{Best} or $R_{\perp_j}$ for updating TIPS and count statistics in Section 4.3. If $Best$ is an interval, an extra step is required for determining the optimal split for each child interval c in $child(Best)$. This requires making a scan on records in R_c, which is a subset of R_{Best}. To determine a split, R_c has to be sorted, which can be an expensive operation. Fortunately, resorting R_c is unnecessary for each iteration because its superset R_{Best} are already sorted. Thus, this type of work involves one scan of the records being refined in each iteration. The second type of work computes $Score(x)$ for the candidates x in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ *without* accessing data records in Section 4.4. For a table with m attributes and each taxonomy tree with at most p nodes, the number of such x is at most $m \times p$. This computation makes use of the maintained count statistics and does not access data records. Let h be the maximum number of times that a value in a record will be refined. For an attribute with a taxonomy tree, h is bounded by the height of the taxonomy tree, and for an attribute without taxonomy tree, h is bounded by 1 (that is, a suppressed value is refined at most once). In the whole computation, each record will be refined at most $m \times h$ times, therefore accessed at most $m \times h$ times because only refined records are accessed. Since $m \times h$ is a small constant independent of the table size, our algorithm is linear in the table size.

In the special case that there is only a single QID, each root-to-leaf path in TIPS has represented a qid, and we can store $a(qid)$ directly at the leaf partitions in TIPS without QITS. A single QID was considered in [7], [10], [12], and [16] where the QID contains all potentially identifying attributes to be used for linking the table to an external source. Our algorithm is more efficient in this special case.

To focus on main ideas, our current implementation assumes that the compressed table fits in memory. Often, this assumption is valid because the compressed table can be much smaller than the original table. If the compressed table does not fit in the memory, we can store leaf partitions of TIPS on disk if necessary. Favorably, the memory is used to keep only leaf partitions that are smaller than the page size to avoid fragmentation of disk pages. A nice property of TDR is that leaf partitions that cannot be further refined (that is, on which there is no candidate refinement) can be discarded, and only some statistics for them needs to be kept. This likely applies to small partitions in memory, therefore, the memory demand is unlikely to build up.

Compared to iteratively masking the data bottom-up starting from domain values, the TDR is more natural and efficient for handling continuous attributes. To produce a small number of intervals for a continuous attribute, the top-down approach needs only a small number of interval splitting, whereas the bottom-up approach needs many interval merging starting from many domain values. In addition, the top-down approach can discard data records that cannot be further refined, whereas the bottom-up approach has to keep all data records until the end of computation.

5 Experimental Evaluation

Our goal in this section is to evaluate the proposed method, that is, TDR, in terms of preserving the usefulness for classification and the scalability on large data sets. For the usefulness evaluation, we compare the classifier built from the masked data with the classifier built from the unmodified data. This comparison makes sense because the anonymization is due to the privacy consideration and the data will be released without modification in the absence of such consideration. In addition, the unmodified data has the lowest possible cost, therefore, it serves the best possible candidate according to previous cost metrics [7], [12], [16]. Though some recent works such as that in [7] model the classification metric on the masked table, the optimality of such metrics does not translate into the optimality of classifiers, as pointed out in Section 1. To our knowledge, [12] is the only work that has evaluated the impact of anonymity on classification with single dimensional generalization. For these reasons, our evaluation uses the baseline of the unmodified data and the reported results in [12]. All experiments on TDR were conducted on an Intel Pentium IV 2.6-GHz PC with 1-Gbyte RAM.

5.1 Data Quality

Our first objective is to evaluate if the proposed TDR preserves the quality for classification while masking the data to satisfy various anonymity requirements. We used the C4.5 classifier [27] and Naive Bayesian classifier (from http://magix.fri.uni-lj.si/orange/) as classification models. We adopted three widely used benchmarks: *Adult*, *Japanese Credit Screening* (*CRX*), and *German Credit Data* (*German*) were obtained from the University of California, Irvine (UCI) repository [28]. Unless stated otherwise, all attributes were used for building classifiers.

In a typical real-life situation, the data provider releases all data records in a single file, leaving the split of training and testing sets to the data miner. Following this practice, we combined the training set and testing set into one set for masking and built a classifier using the masked training set and collected the error using the masked testing set. This error, called the *anonymity error*, denoted AE, was compared with the *baseline error*, denoted BE, for the unmodified training and testing sets. Note that AE depends on the anonymity requirement. $AE - BE$ measures the quality loss due to data masking.

5.1.1 Data Set: Adult

The *Adult* data set has six continuous attributes, eight categorical attributes, and a binary $Class$ column representing two income levels, $\leq$ 50K or $>$ 50K. Table 3a describes each attribute (*cont.* for continuous and *cat.* for categorical). After removing records with missing values from the presplit training and testing sets, we have 30,162 and 15,060 records for training and testing, respectively. This is exactly the same data set as used in [12].

For the same anonymity threshold k, a single QID is always more restrictive than breaking it into multiple QIDs. For this reason, we first consider the case of single QID. To ensure that masking is working on attributes that have an impact on classification, the QID contains the top N attributes ranked by the C4.5 classifier. The top rank attribute is the attribute at the top of the C4.5 decision tree. Then, we remove this attribute and repeat this process to determine the rank of other attributes. The top nine attributes are Cg, Ag, M, En, Re, H, S, E, and O in that order. We specified three anonymity requirements denoted Top5, Top7, and Top9, where the QID contains the top five,

TABLE 3
Attributes for Data Sets

Attribute	Type	Numerical Range	
		# of Leaves	# of Levels
Age (Ag)	cont.	17 - 90	
Capital-gain (Cg)	cont.	0 - 99999	
Capital-loss (Cl)	cont.	0 - 4356	
Education-num (En)	cont.	1 - 16	
Final-weight (Fw)	cont.	13492 - 1490400	
Hours-per-week (H)	cont.	1 - 99	
Education (E)	cat.	16	5
Martial-status (M)	cat.	7	4
Native-country (N)	cat.	40	5
Occupation (O)	cat.	14	3
Race (Ra)	cat.	5	3
Relationship (Re)	cat.	6	3
Sex (S)	cat.	2	2
Work-class (W)	cat.	8	5

(a)

Attribute	Type	Numerical Range
		# of Values
Duration (Du)	cont.	4 - 72
Credit (Cd)	cont.	250 - 18424
Installment-rate (Ir)	cont.	1 - 4
Residence-time (Rt)	cont.	1 - 4
Age (Ag)	cont.	19 - 75
Existing-credits (Ec)	cont.	1 - 4
Liable-people (Li)	cont.	1 - 2
Account-status (As)	cat.	4
Credit-history (Ch)	cat.	5
Loan-purpose (Lp)	cat.	11
Savings-account (Sa)	cat.	5
Employment (Em)	cat.	5
Personal-status (Ps)	cat.	5
Debtors (D)	cat.	3
Property (Pr)	cat.	4
Installments (I)	cat.	3
Housing (H)	cat.	3
Job (J)	cat.	4
Telephone (T)	cat.	2
Foreign (F)	cat.	2

(b)

(a) The Adult data set and (b) the German data set.

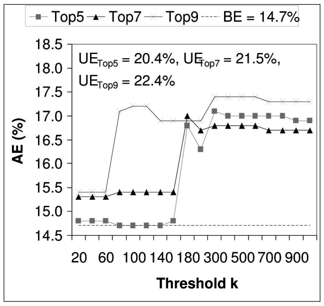


Fig. 5. Suppress and Discretize TopN in *Adult*.

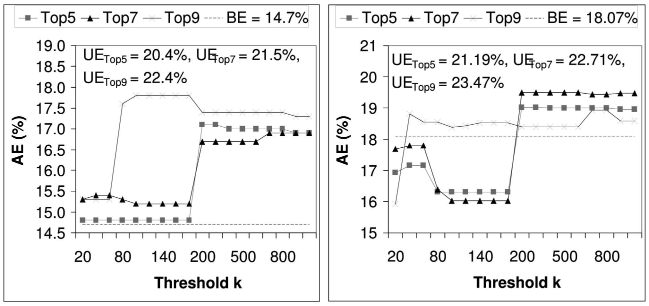
seven, and nine attributes, respectively. The *upper error*, denoted UE, refers to the error on the data with all the attributes in the QID removed (equivalent to generalizing them to the topmost ANY or suppressing them to $\perp$ or including them into a full-range interval). $UE - BE$ measures the impact of the QID on classification.

Fig. 5 displays AE for the C4.5 classifier with the anonymity threshold $20 \leq k \leq 1,000$ by applying discretization on the six continuous attributes and suppression on the eight categorical attributes without taxonomy trees. Note that k is not spaced linearly. We summarize the analysis for Top7 as follows: First, $AE - BE$, where $BE = 14.7$ percent, is less than 2.5 percent over the entire range of tested anonymity threshold, and AE is much lower than $UE = 21.5$ percent. This supports that accurate classification and privacy protection can coexist. Second, AE *generally* increases as the anonymity threshold k increases, but not monotonically. For example, the error slightly drops when k increases from 180 to 200. This is due to the variation between the training and testing sets, and the fact that a better structure may appear in a more masked state.

We further evaluate the effectiveness of generalization on categorical attributes with taxonomy trees. Although Iyengar [12] has specified taxonomy trees for categorical attributes, we do not agree with the author's groupings. For example, the author grouped *Native-country* according to continents, except Americas. We followed the grouping according to the *World Factbook* published by the Central Intelligence Agency (CIA) (http://www.cia.gov/cia/publications/factbook/).

Fig. 6a displays AE for the C4.5 classifier with the anonymity threshold $20 \leq k \leq 1,000$ by applying discretization on the six continuous attributes and generalization on the eight categorical attributes according to our specified taxonomy trees. We summarize the analysis for Top7 as follows: $AE - BE$, where $BE = 14.7$ percent, is less than 2 percent over the range of anonymity threshold $20 \leq k \leq 600$, and AE is much lower than $UE = 21.5$ percent. These results are similar to the results in Fig. 5 although the finally masked versions of data are very different. This suggests that there exist redundant "good" classification structures in the data.

A closer look at the masked data for Top7 with $k = 500$ reveals that among the seven top ranked attributes, three are masked to a different degree of granularity, and four, namely, Cg (ranked first), Ag (ranked second), Re (ranked fifth), and S (ranked seventh), are masked to the topmost value ANY. Even for this drastic masking, AE has only increased by 2 percent from $BE = 14.7$ percent, whereas the



(a) (b)

Fig. 6. Generalize and discretize topN in *Adult*. (a) C4.5. (b) Naive Bayesian.

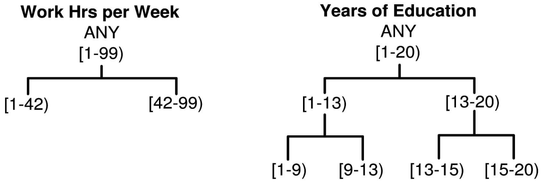


Fig. 7. Generated taxonomy trees of *Hours-per-week* and *Education-num*.

worst case can be $UE = 21.5$ percent. With the masking, classification now is performed by the remaining three attributes in the QID and the unmodified but lower ranked attributes. Clearly, this is a different classification structure from what would be found from the unmodified data. As a result, though masking may eliminate some structures, new structures emerge to help.

Fig. 6b displays AE for the Naive Bayesian classifier. Compared to the C4.5 classifier, though BE and UE are higher (which has to do with the classification method, not the masking), the quality loss due to masking, $AE - BE$ (note $BE = 18.07$ percent), is smaller, which is no more than 1.5 percent for the range of anonymity threshold $20 \leq k \leq 1,000$. This suggests that the information-based masking is also useful to other classification methods such as the Naive Bayesian that do not use the information gain. Another observation is that AE is even lower than BE for the anonymity threshold $k \leq 180$ for Top5 and Top7. This confirms again that the optimal k-anonymization is not relevant to the classification goal due to possibility of "overfitting." The unmodified data certainly has the least distortion by any cost metric. However, this experiment shows that the least distortion does not translate into the accuracy of classifier. $AE < BE$ also occurs in the experiment on the *CRX* data set in Fig. 9a. Our approach is bias toward masking the noises in order to help classification.

Fig. 7 shows the generated taxonomy trees for continuous attributes *Hours-per-week* and *Education-num* with Top7 and $k = 60$. The splits are very reasonable. For example, in the taxonomy tree of *Education-num*, the split point at 13 distinguishes whether the person has post secondary education. If the user does not like these trees, she may modify them or specify her own and, subsequently, treat continuous attributes as categorical attributes with specified taxonomy trees.

Our method took at most 10 seconds for all previous experiments. Out of the 10 seconds, approximately 8 seconds were spent on reading data records from disk and writing the masked data to disk. The actual processing time for generalizing the data is relatively short.

In an effort to study the effectiveness of multiple QIDs, we compared AE between a multiple QIDs requirement and the *corresponding* single united QID requirement. We randomly generated 30 multiple QID requirements as follows: For each requirement, we first determined the number of QIDs using the uniform distribution $U[3,7]$ (that is, randomly drawn a number between 3 and 7) and the length of QIDs using $U[2,9]$. For simplicity, all QIDs in the same requirement have the same length and same threshold $k = 100$. For each QID, we randomly selected some attributes according to the QID length from the 14 attributes.

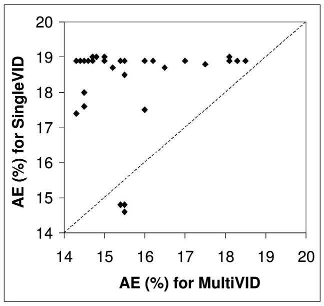


Fig. 8. SingleQID versus MultiQID ($k = 100$).

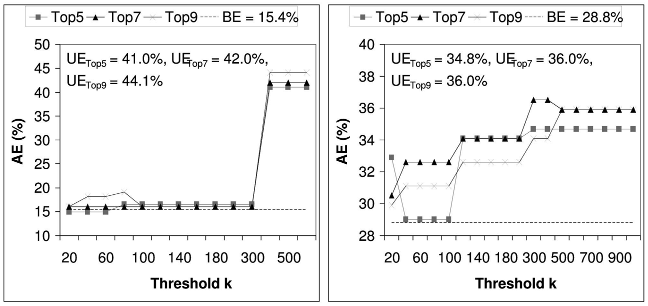
A repeating QID was discarded. For example, a requirement of three QIDs and length two is $\{\langle\{Ag, En\}, k\rangle, \langle\{Ag, Ra\}, k\rangle, \langle\{S, H\}, k\rangle\}$, and the corresponding single QID requirement is $\{\langle\{Ag, En, Ra, S, H\}, k\rangle\}$.

In Fig. 8, each data point represents the AE of a multiple QID requirement, denoted MultiQID, and the AE of the corresponding single QID requirement, denoted SingleQID. The C4.5 classifier was used. Most data points appear at the upper left corner of the diagonal, suggesting that MultiQID generally yields lower AE than its corresponding SingleQID. This verifies the effectiveness of multiple QIDs to avoid unnecessary masking and improve data quality.

5.1.2 Data Set: CRX

The *CRX* data set is based on a credit card application. There are six continuous attributes, 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 presplit training and testing, respectively. In the UCI repository, all values and attribute names in *CRX* have been changed to meaningless symbols such as $A_1 \dots A_{15}$. No taxonomy tree is given in advance. The Top9 attributes in *CRX* are A_9, A_{11}, A_{10}, A_8, A_{15}, A_7, A_{14}, A_6, and A_5 in that order.

Fig. 9a displays AE for the C4.5 classifier with the anonymity threshold $20 \leq k \leq 600$ by applying discretization on the six continuous attributes and suppression on the eight categorical attributes without taxonomy trees. Different anonymity requirements Top5, Top7, and Top9 yield similar AE's and similar patterns, indicating more restrictive requirement may not have much impact on classification



(a) (b)

Fig. 9. Suppress and discretize TopN in *CRX* and *German*. (a) *CRX*. (b) *German*.

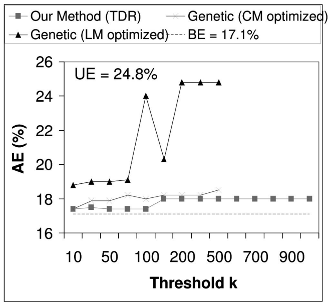


Fig. 10. Comparing with genetic algorithm.

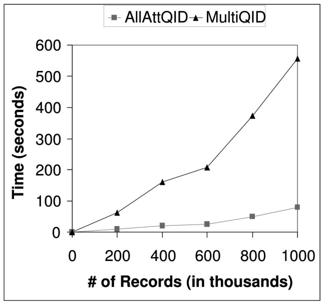


Fig. 11. Scalability versus number of records $(k = 50)$.

quality. This largely depends on the availability of alternative "good" classification structures in the data set.

We summarize the analysis for Top7 as follows: First, $AE - BE$, where $BE = 15.4$ percent, is less than 4 percent over the range of anonymity threshold $20 \leq k \leq 300$, and AE is much lower than $UE = 42$ percent. This supports that accurate classification and privacy protection can coexist. AE drastically increases when $k > 300$ since CRX only has 653 records.

5.1.3 Data Set: German

German has seven continuous attributes, 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 presplit training and testing, respectively. Table 3b describes each attribute. The Top9 attributes in *German* are Cd, As, Du, Ch, Sa, I, Lp, D, and Pr in that order.

Fig. 9b displays AE for the C4.5 classifier with the anonymity threshold $20 \leq k \leq 1,000$ by applying discretization on the seven continuous attributes and suppression on the 13 categorical attributes without taxonomy trees. $AE - BE$, where $BE = 28.8$ percent, is less than 4 percent over the range of anonymity threshold $20 \leq k \leq 100$ for the anonymity requirement Top7. Although AE is mildly lower than $UE = 36$ percent, the benefit of masking $UE - AE$ is not as significant as in other data sets. If the data provider requires higher degree of anonymity, then she may consider simply removing the TopN attributes from the data set rather than masking them.

5.2 Comparing with Other Algorithms

Iyengar [12] presented a genetic algorithm solution. This experiment was customized to conduct a fair comparison with the results in [12]. We used the same *Adult* data set, same attributes, and same anonymity requirement as specified in [12]:

$$\text{GA} = \langle \{Ag, W, E, M, O, Ra, S, N\}, k \rangle.$$

We obtained the taxonomy trees from the author for generalization, except for the continuous attribute Ag, which we used discretization. Following the procedure in [12], all attributes not in GA were removed and were not used to produce BE, AE, and UE in this experiment, and all errors were based on the 10-fold cross validation and the C4.5 classifier. For each fold, we first masked the training data and then applied the masking to the testing data.

Fig. 10 compares AE of TDR with the errors reported for two methods in [12], Loss Metric (LM) and Classification Metric (CM), for $10 \leq k \leq 500$. TDR outperformed LM, especially for $k \geq 100$ but performed only slightly better than CM. TDR continued to perform well from $k = 500$ to $k = 1,000$, for which no result was reported for LM and CM in [12]. This analysis shows that our method is at least comparable to genetic algorithm [12] in terms of accuracy. However, our method took only 7 seconds to mask the data, including reading data records from disk and writing the masked data to disk. Iyengar [12] reported that his method requires 18 hours to transform this data, which has about only 30K data records. Clearly, the genetic algorithm is not scalable.

Recently, [18] made a comparison with our previous version of TDR in [14] in terms of data quality on some other data sets. Their experiments suggested that the classification quality on the masked data can be further improved by using a more flexible masking operation, multidimensional generalization; however, this type of generalization suffers from the interpretation difficulty as discussed in Section 1. Xu et al. [19] reported that the multidimensional generalization algorithm took about 10 seconds to mask the *Adult* data set. We compared TDR with some recently developed greedy anonymization algorithms that also conducted experiments on the *Adult* data set. The efficiency of the bottom-up cell generalization algorithm in [15] is comparable to TDR when $k = 2$, 10, but they did not report the efficiency for larger k. A cell generalization algorithm in [19] took about 60 seconds to mask the data. In general, multidimensional and cell generalization algorithms are less efficient than our method due to the larger number of possible masked tables.

5.3 Efficiency and Scalability

This experiment evaluates the scalability of TDR by blowing up the size of the *Adult* data set. First, we combined the training and testing sets, giving 45,222 records. For each original record r in the combined set, we created $\alpha - 1$ "variations" of r, where $\alpha > 1$ is the blowup scale. For each variation of r, we randomly selected q attributes from $\cup QID_j$, where q has the uniform distribution $U[1, |\cup QID_j|]$, that is, randomly drawn between 1 and the number of attributes in QIDs and replaced the values on the selected attributes with values randomly drawn from the domain of the attributes. Together with all original records, the enlarged data set has $\alpha \times 45,222$ records. To provide a precise evaluation, the runtime reported excludes the time for loading data records from disk and the time for writing the masked data to disk.

Fig. 11 depicts the runtime of TDR using generalization and discretization for 200,000 to 1,000,000 data records and the anonymity threshold $k = 50$ based on two types of anonymity requirements. AllAttQID refers to the single QID having all 14 attributes. This is one of the most time-consuming settings because of the largest number of candidate refinements to consider at each iteration. For TDR, the small anonymity threshold of $k = 50$ requires more iterations to reach a solution, hence, more runtime, than a larger threshold. TDR takes approximately 80 seconds to transform 1,000,000 records.

In Fig. 11, MultiQID refers to the average runtime over the 30 random multiple QID requirements in Section 5.1 with $k = 50$. Compared to AllAttQID, TDR becomes less efficient for handling multiple QIDs for two reasons. First, an anonymity requirement on multiple QIDs is a less restrictive constraint than the single QID anonymity requirement containing all attributes; therefore, TDR has to perform more refinements before violating the anonymity requirement. Moreover, TDR needs to create one QIT for each QID and maintains $a(qid)$ in QITS. The increase is roughly by a factor proportional to the number of QIDs in an anonymity requirement. The runtime of suppression and discretization on this expanded data set is roughly the same as shown in Fig. 11.

5.4 Summary

Our experiments verified several claims about the proposed TDR method. First, TDR masks a given table to satisfy a broad range of anonymity requirements without sacrificing significantly the usefulness to classification. Second, while producing a comparable accuracy, TDR is much more efficient than previously reported approaches, particularly, the genetic algorithm in [12]. Third, the previous optimal k-anonymization [7], [16] does not necessarily translate into the optimality of classification. The proposed TDR finds a better anonymization solution for classification. Fourth, the proposed TDR scales well with large data sets and complex anonymity requirements. These performances together with the features discussed in Section 1 make TDR a practical technique for privacy protection while sharing information.

6 Conclusions

We considered the problem of ensuring an individual's anonymity while releasing person-specific data for classification analysis. We pointed out that the previous optimal k-anonymization based on a closed-form cost metric does not address the classification requirement. Our approach is based on two observations specific to classification: Information specific to individuals tends to be overfitting, thus of little utility, to classification; even if a masking operation eliminates some useful classification structures, alternative structures in the data emerge to help. Therefore, not all data items are equally useful for classification and less useful data items provide the room for anonymizing the data without compromising the utility. With these observations, we presented a top-down approach to iteratively refine the data from a general state into a special state, guided by maximizing the trade-off between information and anonymity. This top-down approach serves a natural and efficient structure for handling categorical and continuous attributes and multiple anonymity requirements. Experiments showed that our approach effectively preserves both information utility and individual's privacy and scales well for large data sets.

Acknowledgments

Research was supported in part by a research grant and a PGS scholarship from the Natural Sciences and Engineering Research Council of Canada.

References

[1] P. Samarati and L. Sweeney, "Generalizing Data to Provide Anonymity When Disclosing Information," *Proc. 17th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems (PODS '98)*, p. 188, June 1998.

[2] P. Samarati, "Protecting Respondents' Identities in Microdata Release," *IEEE Trans. Knowledge Eng.*, vol. 13, no. 6, pp. 1010-1027, Nov./Dec. 2001.

[3] The House of Commons in Canada, "The Personal Information Protection and Electronic Documents Act," 1991, http://www.privcom.gc.ca/.

[4] S.M. Weiss and C.A. Kulikowski, *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Machine Learning, and Expert Systems.* Morgan Kaufmann, 1991.

[5] T. Dalenius, "Finding a Needle in a Haystack—or Identifying Anonymous Census Record," *J. Official Statistics*, vol. 2, no. 3, pp. 329-336, 1986.

[6] L. Sweeney, "Achieving k-Anonymity Privacy Protection Using Generalization and Suppression," *Int'l J. Uncertainty, Fuzziness, and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571-588, 2002.

[7] R.J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization," *Proc. 21st Int'l Conf. Data Eng. (ICDE '05)*, pp. 217-228, Apr. 2005.

[8] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigraphy, D. Thomas, and A. Zhu, "Approximation Algorithms for k-Anonymity," *J. Privacy Technology*, no. 20051120001, Nov. 2005.

[9] A. Meyerson and R. Williams, "On the Complexity of Optimal k-Anonymity," *Proc. 23rd ACM Symp. Principles of Database Systems (PODS '04)*, pp. 223-228, 2004.

[10] L. Sweeney, "Datafly: A System for Providing Anonymity in Medical Data," *Proc. Int'l Conf. Database Security*, pp. 356-381, 1998.

[11] A. Hundepool and L. Willenborg, "μ- and τ-Argus: Software for Statistical Disclosure Control," *Proc. Third Int'l Seminar on Statistical Confidentiality*, 1996.

[12] V.S. Iyengar, "Transforming Data to Satisfy Privacy Constraints," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 279-288, July 2002.

[13] K. Wang, P. Yu, and S. Chakraborty, "Bottom-Up Generalization: A Data Mining Solution to Privacy Protection," *Proc. Fourth IEEE Int'l Conf. Data Mining (ICDM '04)*, Nov. 2004.

[14] B.C.M. Fung, K. Wang, and P.S. Yu, "Top-Down Specialization for Information and Privacy Preservation," *Proc. 21st Int'l Conf. Data Eng. (ICDE '05)*, pp. 205-216, Apr. 2005.

[15] R.C.W. Wong, J. Li, A.W.C. Fu, and K. Wang, "(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06)*, Aug. 2006.

[16] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain k-Anonymity," *Proc. 2005 ACM SIGMOD Int'l Conf. Management of Data*, pp. 49-60, 2005.

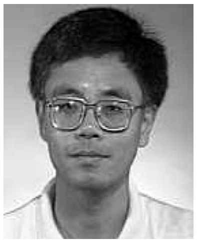
[17] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," *Proc. 22nd IEEE Int'l Conf. Data Eng. (ICDE '06)*, 2006.

[18] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Workload-Aware Anonymization," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06)*, Aug. 2006.

[19] J. Xu, W. Wang, X. Pei, B. Wang, B. Xie, and A.W.C. Fu, "Utility-Based Anonymization Using Local Recoding," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06)*, Aug. 2006.

[20] A. Machanavajjhala, J. Gehrke, and D. Kifer, "l-Diversity: Privacy beyond k-Anonymity," *Proc. 22nd Int'l Conf. Data Eng. (ICDE '06),* 2006.
[21] K. Wang, B.C.M. Fung, and P.S. Yu, "Template-Based Privacy Preservation in Classification Problems," *Proc. Fifth IEEE Int'l Conf. Data Mining (ICDM '05),* pp. 466-473, Nov. 2005.
[22] K. Wang, B.C.M. Fung, and P.S. Yu, "Handicapping Attacker's Confidence: An Alternative to k-Anonymization," *Knowledge and Information Systems: An Int'l J. (KAIS),* 2006.
[23] X. Xiao and Y. Tao, "Personalized Privacy Preservation," *Proc. 2006 ACM SIGMOD Int'l Conf. Management of Data,* June 2006.
[24] K. Wang, B.C.M. Fung, and G. Dong, "Integrating Private Databases for Data Analysis," *Proc. 2005 IEEE Int'l Conf. Intelligence and Security Informatics (ISI '05),* pp. 171-182, May 2005.
[25] K. Wang and B.C.M. Fung, "Anonymizing Sequential Releases," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06),* pp. 414-423, Aug. 2006.
[26] C.E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical J.,* vol. 27, pp. 379-623, 1948.
[27] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
[28] D.J. Newman, "UCI Repository of Machine Learning Databases," http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

Benjamin C.M. Fung received the BSc and MSc degrees in computing science from Simon Fraser University. He is currently a PhD candidate at Simon Fraser with the postgraduate scholarship doctoral award from the Natural Sciences and Engineering Research Council of Canada (NSERC). His recent research interests include privacy-preserving data mining/publishing, secure distributed computing, and text mining. Before pursuing his PhD, 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. He has published in data mining and security conferences, journals, and books. He has served on program committees for international conferences.



Ke Wang received the PhD degree from the Georgia Institute of Technology. He is currently a professor at the School of Computing Science, Simon Fraser University. Before joining Simon Fraser, he was an associate professor at the National University of Singapore. He has taught in the areas of database and data mining. His 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, biomining, 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, and machine learning and optimization to provide actionable solutions to real-life problems. He is an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and has served on program committees for international conferences.

Philip S. Yu received the BS degree in electrical engineering from the National Taiwan University, the MS and PhD degrees in electrical engineering from Stanford University, and the MBA degree from New York University. He is with the IBM T.J. Watson Research Center and is currently the manager of the Software Tools and Techniques Group. He has published more than 450 papers in refereed journals and conferences. He holds or has applied for more than 250 US patents. He 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. He is an IBM Master Inventor.