Bottom-Up Generalization: A Data Mining Solution to Privacy Protection *

Ke Wang
Simon Fraser University
wangk@cs.sfu.ca

Philip S. Yu
IBM T. J. Watson Research Center
psyu@us.ibm.com

Sourav Chakraborty
Simon Fraser University
chakrabo@cs.sfu.ca

Abstract

The well-known privacy-preserved data mining *modifies existing data mining techniques to randomized data. In this paper, we investigate data mining as a technique for masking data, therefore, termed* data mining based privacy protection. *This approach incorporates partially the requirement of a targeted data mining task into the process of masking data so that essential structure is preserved in the masked data. The idea is simple but novel: we explore the data generalization concept from data mining as a way to hide detailed information, rather than discover trends and patterns. Once the data is masked, standard data mining techniques can be applied without modification. Our work demonstrated another positive use of data mining technology: not only can it discover useful patterns, but also mask private information.*

We consider the following privacy problem: a data holder wants to release a version of data for building classification models, but wants to protect against linking the released data to an external source for inferring sensitive information. We adapt an iterative bottom-up generalization from data mining to generalize the data. The generalized data remains useful to classification but becomes difficult to link to other sources. The generalization space is specified by a hierarchical structure of generalizations. A key is identifying the best generalization to climb up the hierarchy at each iteration. Enumerating all candidate generalizations is impractical. We present a scalable solution that examines at most one generalization in each iteration for each attribute involved in the linking.

1 Introduction

The increasing ability to accumulate, store, retrieve, cross-reference, mine and link vast number of electronic records brings substantial benefits to millions of people. For example, cross-mining personal records on chemical exposure and death records could help identify cancer-causing substances. These advances also raise responsibility and privacy concerns because of the potential of creating new information. An example given in [11] is that a *sensitive* medical record was uniquely linked to a *named* voter record in a publicly available voter list through the shared attributes of Zip, Birth date, Sex. Indeed, since "the whole is greater than the sum of the parts", protection of individual sources does not guarantee protection when sources are cross-examined. A relevant research topic is finding ways to safeguard against inferring private information through record linkage while continuing to allow benefits of information sharing and data mining.

1.1 Our contribution

Information becomes sensitive when they are specific to a small number of individuals. Data mining, on the other hand, typically makes use of information shared by some minimum number of individuals to ensure a required statistical significance of patterns. As such, sensitive information are to be discarded for reliable data mining. This observation motivates us to apply the requirement of an intended data mining task to identify useful information to be released, therefore, sensitive information to be masked. This approach, called *data mining based privacy protection*, turns data mining from a threat into a solution to privacy protection.

We consider the following *anonymity problem* [10]. A data holder wants to release a person-specific data R, but wants to prevent from linking the released data to an external source E through shared attributes $R \cap E$, called the *virtual identifier*. One approach is to generalize specific values into less specific but semantically consistent values to create K*-anonymity*: if one record r in R is linked to some external information, at least $K-1$ other records are similarly linked by having the same virtual identifier value as r. The idea is to make the inference ambiguous by creating extraneous linkages. An example is generalizing "birth date" to "birth year" so that every body born in the same year are

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

linked to a medical record with that birth year, but most of these linkages are non-existing in the real life.

We focus on the use of data for building a classifier. We propose a data mining approach, an iterative bottom-up generalization, to achieve the required K anonymity while preserving the usefulness of the generalized data to classification. The generalization space is specified by a taxonomical hierarchy per attribute in the virtual identifier. The key is identifying the "best" generalization to climb up the hierarchy at each iteration. Evaluating all possible candidates at each iteration is not scalable because each evaluation involves examining data records. We present a scalable solution that examines *at most one generalization per attribute in the virtual identifier in each iteration*, where the work for examining one generalization is proportional to the number of (distinct) virtual identifier values that are actually generalized. We evaluate both quality and scalability of this approach.

1.2 Related work

A well-studied technique for masking sensitive information, primarily studied in statistics, is *randomizing* sensitive attributes by adding random error to values [2, 3, 4, 8]. Recently, this technique was studied in data mining [1]. In these works, privacy was quantified by how closely the original values of a randomized attribute can be estimated. This is very different from the K-anonymity that quantifies how likely an individual can be linked to an external source. The *privacy-preserving data mining* in [1] extends traditional data mining techniques to handle randomized data. We investigate data mining itself as a technique for masking data. The masked data does not require modification of data mining techniques in subsequent data analysis.

Instead of randomizing data, *generalizing* data makes information less precise. Grouping continuous values and suppressing values are examples of this approach. Compared to randomization, generalization has several advantages. First, it preserves the "truthfulness" of information, making the released data meaningful at the record level. This feature is desirable in exploratory and visual data mining where decisions often are made based on examining records. In contrast, randomized data are useful only at the aggregated level such as average and frequency. Second, preferences can be incorporated through the taxonomical hierarchies and the data recipient can be told what was done to the data so that the result can be properly interpreted.

Generalization was used to achieve anonymity in Datafly system [10] and μ-Argus system [6]. Their works did not consider classification or a specific use of released data. In fact, data distortion was simply measured by the number of hierarchy levels climbed up [6]. Each iteration selected the attribute having most number of distinct values in the Datafly system or values not having K occurrences in the μ-Argus system to generalize or suppress. Such selection did not address the quality for classification where there is a different impact between generalization within a class and that across classes.

To our knowledge, [7] is the only work that has considered the anonymity problem for classification, and presented a genetic algorithm to search for the optimal generalization of the data. As noted in [7], their solution took 18 hours to generalize 30K records. We uses an entirely different approach, the iterative bottom-up generalization, and we focused on the scalability issue. We used an information/privacy trade-off to *select* a generalization, whereas [7] used the privacy requirement to filter a *selected* generalization. Furthermore, the sequence of generalizations produced by our approach can be used to determine a desired trade-off point between privacy and quality. The genetic evolution of random nature does not serve this purpose, neither does the final generalized data because the same final state can be reached by many generalization sequences.

Bottom-up generalization was previously used for extracting patterns, see [5] for example. The new lights in our work are the consideration on privacy protection, quality preservation, and the related scalability issue.

2 The Problem

Consider that a data holder wants to release a person-specific data $R(D_1, \cdots, D_n, C)$ to the public. A record has the form $< v_1, \cdots, v_n, cls >$, where v_i is a domain value from the attribute D_i and cls is a class in C. Suppose that R shares some attributes with an external source E, denoted $R \cap E$. If a value on $R \cap E$ is so specific that the probability of having this value by chance is negligible, each linking from a record in R to some information in E through this value has a good chance of identifying a real life fact. The data holder protects against such linkages by requiring a minimum number of records linkable through each value on $R \cap E$.

Definition 1 (Anonymity) The *virtual identifier*, denoted VID, is the set of attributes shared by R and E. $a(vid)$ denotes the number of records in R with the value vid on VID. The *anonymity* of VID, denoted $A(VID)$, is the minimum $a(vid)$ for any value vid on VID. If $a(vid) = A(VID)$, vid is called an *anonymity vid*. R satisfies the anonymity requirement $< VID, K >$ if $A(VID) \geq K$, where K is specified by the data holder. ■

We transform R to satisfy the anonymity requirement by generalizing specific values on VID into less specific but semantically consistent values. The generalization increases the probability of having a given value on VID by

chance, therefore, decreases the probability that a linking through this value represents a real life fact. The generalization space is specified through a taxonomical hierarchy per attribute in VID, provided by either the data holder or the data recipient. A hierarchy is a tree with leaf nodes representing domain values and parent nodes representing less specific values. R is generalized by a sequence of generalizations, where each generalization replaces all child values c with their parent value p in a hierarchy. Before a value c is generalized, all values below c should be generalized to c first.

Definition 2 (Generalization) A *generalization*, written $\{c\} \rightarrow p$, replaces all child values $\{c\}$ with the parent value p. A generalization is *valid* if all values below c have been generalized to c. A vid is *generalized* by $\{c\} \rightarrow p$ if the vid contains some value in $\{c\}$. ■

Relationship	Race	Workclass	$a(vid)$	C
c_1	b_2	a_3	4	0Y4N
c_1	b_2	c_3	4	0Y4N
c_1	b_2	d_3	3	0Y3N
c_1	c_2	a_3	3	2Y1N
c_1	c_2	b_3	4	2Y2N
d_1	c_2	b_3	4	4Y0N
d_1	c_2	e_3	2	2Y0N
d_1	d_2	b_3	3	2Y1N
d_1	d_2	e_3	2	2Y0N

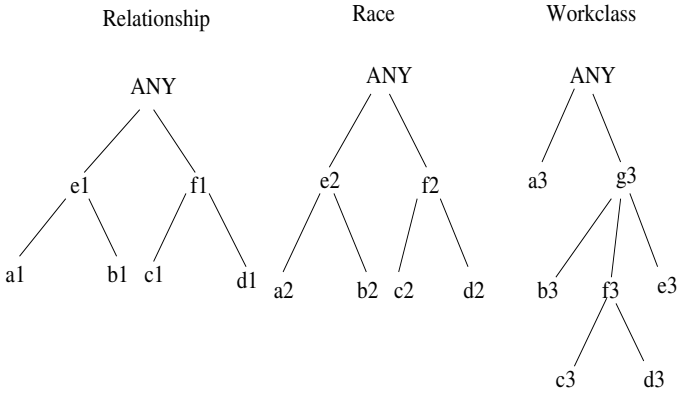

Figure 1. Data and hierarchies for VID

Example 1 Consider

$$VID = \{Relationship, Race, Workclass\},$$

and the hierarchies and vids in Figure 1. We have compressed all records having the same value on VID into a single row with the distribution of the Y/N class label and the count $a(vid)$. Initially, the generalizations at e_1, f_1, e_2, f_2, f_3 are valid, $A(VID) = 2$, and $d_1c_2e_3$ and $d_1d_2e_3$ are anonymity vids. The requirement of $K = 3$ can be satisfied by applying $\{c_2, d_2\} \rightarrow f_2$, which generalizes the vids $d_1c_2e_3$ and $d_1d_2e_3$ into a single vid $d_1f_2e_3$ with $a(d_1f_2e_3) = 4$. ■

Definition 3 (Anonymity for Classification) Given a relation R, an anonymity requirement $< VID, K >$, and a hierarchy for each attribute in VID, generalize R, by a sequence of generalizations, to satisfy the requirement and contain as much information as possible for classification. ■

The anonymity requirement can be satisfied in more than one way of generalizing R, and some lose more information than others with regard to classification. One question is how to select a sequence of generalizations so that information loss is minimized. Another question is how to find this sequence of generalizations efficiently for a large data set. We like to answer these questions in the rest of the paper.

3 Metrics for generalization

We consider a metric for a single generalization, which is used to guide the search of a sequence of generalizations in the next section. A "good" generalization should preserve information for classification *and* focus on the goal of achieving the K-anonymity. Let us formalize this criterion.

Consider a generalization $G : \{c\} \rightarrow p$. Let R_c denote the set of records containing c, and let R_p denote the set of records containing p after applying G. $|R_p| = \Sigma_c |R_c|$, where $|x|$ is the number of elements in a bag x. The effect of G is summarized by the "information loss" and "anonymity gain" after replacing R_c's with R_p.

We adapt the entropy based information loss, which can be substituted by other information measures:

$$I(G) = Info(R_p) - \Sigma_c \frac{|R_c|}{|R_p|} Info(R_c),$$

where $Info(R_x)$ is the *impurity* or *entropy* of R_x [9]:

$$Info(R_x) = -\Sigma_{cls} \frac{freq(R_x, cls)}{|R_x|} \times log_2 \frac{freq(R_x, cls)}{|R_x|}.$$

$freq(R_x, cls)$ is the number of records in R_x with the class label cls.

The anonymity gain is $A_G(VID) - A(VID)$, where $A(VID)$ and $A_G(VID)$ denote the anonymity before and after applying G, respectively. $A_G(VID) \geq A(VID)$. In the case of $A_G(VID) > K$, $A_G(VID) - K$ is the "surplus" of anonymity. While more anonymity is always preferred for privacy protection, it comes at the expense of losing more information. If such a "surplus" really outweighs the information concern, the data holder should specify a

larger K in the first place. This consideration leads to the modified anonymity gain:

$$P(G) = \ x - A(VID)$$

where $x = A_G(VID)$ if $A_G(VID) \leq K$, and $x = K$ otherwis.

Information-Privacy Metric. To minimize the information loss for achieving a given K-anonymity, our criterion is to favor the generalization having the minimum information loss for each unit of anonymity gain:

$$Minimize : IP(G) = I(G)/P(G).$$

$IP(G)$ is ∞ if $P(G) = 0$. If $P(G) = 0$ for all (valid) generalizations G, we compare them based on $I(G)$. This metric also maximizes the anonymity gain for each unit of information loss. We use $I(G)/P(G)$ instead of $I(G) - P(G)$ because differentiating semantically different quantities makes little sense.

Unlike the "penalty" metric in [7] that focuses on information distortion alone, $IP(G)$ takes into account both information and anonymity. The anonymity consideration helps focus the search on the privacy goal, therefore, has a look-ahead effect. However, this presents a new challenge to scalability because the effect on anonymity is only available after applying a generalization. We will examine this issue in subsequent sections.

4 Bottom-Up Generalization

Algorithm 1 describes our bottom-up generalization process. In the ith iteration, we generalize R by the "best" generalization G_{best} according to the IP metric. This algorithm makes no claim on efficiency because Line 2 and 3 requires computing $IP(G)$ for all candidate generalizations G. Let us look at this computation in more details.

Consider a candidate generalization $G : \{c\} \rightarrow p$ in an iteration. $|R_c|$ and $freq(R_c, cls)$ can be maintained after each iteration. $|R_p|$ and $freq(R_p, cls)$ can be obtained by aggregating $|R_c|$ and $freq(R_c, cls)$. Therefore, $I(G)$ can be easily computed, i.e., without accessing vids. In fact, any metric on a single attribute (plus the class label) can be computed this way. $A(VID)$ is available as a result of applying the previous generalization. Computing $A_G(VID)$, however, depends on the "effect" of G, which is only available after applying G, and requires accessing vids. This is a new challenge to scalability.

Our insight is that most generalizations G do not affect $A(VID)$, therefore, $A_G(VID) = A(VID)$. In fact, if a generalization G fails to generalize *all* anonymity vids, G will not affect $A(VID)$. For such G, $P(G) = 0$ and $IP(G) = \infty$, and our metric does not need $A_G(VID)$. Therefore, we can focus on "critical generalizations" as defined below.

Algorithm 1 The bottom-up generalization

1: **while** R does not satisfy the anonymity requirement **do**
2: **for all** generalization G **do**
3: compute $IP(G)$;
4: **end for**;
5: find the best generalization G_{best};
6: generalize R by G_{best};
7: **end while**;
8: output R;

Definition 4 (Critical generalization) G is *critical* if $A_G(VID) > A(VID)$. ■

A critical generalization G has a non-zero $P(G)$ and a finite $IP(G)$, whereas a non-critical generalization G has a zero $P(G)$ and infinite $IP(G)$. Therefore, so long as one generalization is critical, all non-critical generalizations will be ignored by the IP metric. If all generalizations are non-critical, the IP metric will select the one with minimum $I(G)$. In both cases, $A_G(VID)$ is not needed for a non-critical generalization G. Based on this observation, we optimize Algorithm 1 by replacing Line 2 and 3 with

2: **for all** critical generalization G **do**
3: compute $A_G(VID)$;

Three questions remain: how to identify all critical generalizations without actually computing $A_G(VID)$ for all generalizations; how many generalizations are critical, therefore, need to compute $A_G(VID)$; and how to apply a generalization without scanning all vids. We answer these questions in the next section.

5 Pruning Strategies

A key issue in our approach is how to identify critical generalizations without computing $A_G(VID)$ for all candidate G. First, we present an efficient structure for applying a given generalization.

5.1 The data structure

We store all distinct vids in a tree structure, called *Taxonomy Encoded Anonymity* (TEA) index. Each level of the tree represents the current generalization of a particular attribute, and each path represents a particular vid with $a(vid)$ stored in the leaf node. In addition, the TEA index links up the vids according to the generalizations that generalize them. Each time a generalization is applied, the TEA index is updated by adjusting the vids linked to this generalization. The purpose of this index is to prune the number of candidate generalizations to no more than $|VID|$ at each iteration, where $|VID|$ is the number of attributes in VID.

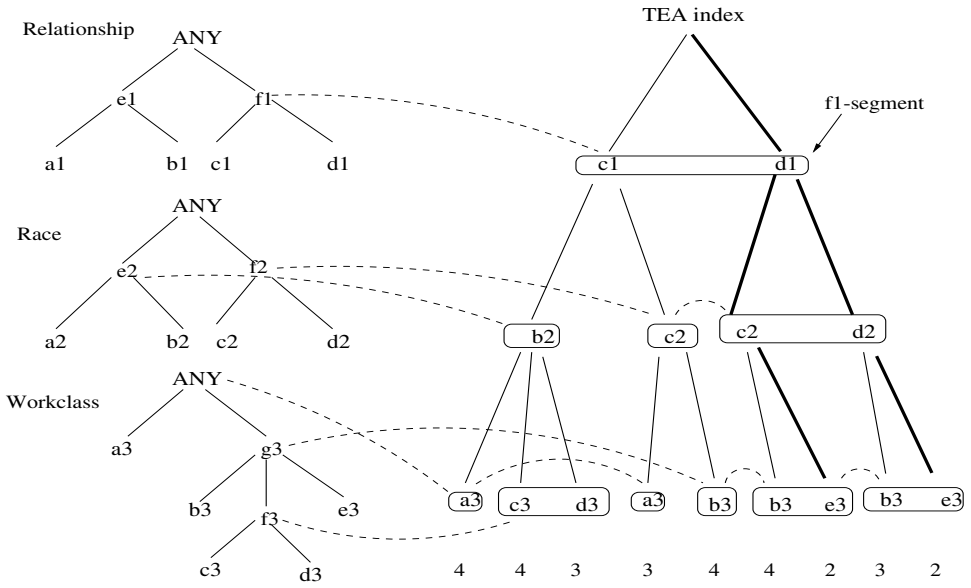

Figure 2. The TEA index for VID

Definition 5 (TEA index) The *Taxonomy Encoded Anonymity* (TEA) index for $VID = \{D_1, \cdots, D_k\}$ is a tree of k levels. The level $i > 0$ represents the current values for D_i. Each root-to-leaf path represents an existing vid in the data, with $a(vid)$ stored at the leaf node. For a generalization $G : \{c\} \rightarrow p$, a *segment* of G is a maximal set of sibling nodes, $\{s_1, \cdots, s_t\}$, such that $\{s_1, \cdots, s_t\} \subseteq \{c\}$, where t is the size of the segment. All segments of G are linked up. A vid is *generalized* by a segment if the vid contains a value in the segment. ■

Intuitively, a segment of G represents a set of sibling nodes in the TEA index that will be merged by applying G. To apply G, we follow the link of the segments of G and merge the nodes in each segment of G. The merging of sibling nodes implies inserting the new node into a proper segment and recursively merging the child nodes having the same value if their parents are merged. The merging of leaf nodes implies summing up $a(vid)$ stored at such leaf nodes. The cost is proportional to the number of vids generalized by G.

Example 2 Figure 2 shows the TEA index for the vids in Example 1. A rectangle represents a segment, and a a dashed line links up the segments of the same generalization. For example, the left-most path represents the vid $c_1b_2a_3$, and $a(c_1b_2a_3) = 4$. $\{c_1, d_1\}$ at level 1 is a segment of f_1 because it forms a maximal set of siblings that will be merged by f_1. $\{c_1c_2\}$ and $\{d_1c_2, d_1d_2\}$ at level 2 are two segments of f_2. $\{c_1b_2c_3, c_1b_2d_3\}$ at level 3 is a segment of f_3. $d_1d_2e_3$ and $d_1c_2e_3$, in bold face, are the anonymity vids.

Consider applying $\{c_2, d_2\} \rightarrow f_2$. The first segment of f_2 contains only one sibling node $\{c_1c_2\}$, we simply relabel the sibling by f_2. This creates new vids $c_1f_2a_3$ and $c_1f_2b_3$. The second segment of f_2 contains two sibling nodes $\{d_1c_2, d_1d_2\}$. We merge them into a new node labeled by f_2, and merge their child nodes having the same label. This creates new vids $d_1f_2b_3$ and $d_1f_2e_3$, with $a(d_1f_2b_3) = 7$ and $a(d_1f_2e_3) = 4$. ■

Observation 1. G is critical *only if* every anonymity vid is generalized by some size-k segment of G, $k > 1$. In Figure 2, no anonymity vid is generalized by the (only) size-2 segment of f_3, so f_3 is not critical. The two anonymity vids are generalized by the (only) segment size-2 of f_1, but f_1 is still not critical.

Observation 2. At each level of the TEA index, each vid is generalized by *at most* one segment. This observation implies that the "only if" condition in Observation 1 holds for at most one generalization at each level of the TEA index.

Theorem 1 G is critical only if every anonymity vid is generalized by some size-k segment of G, where $k > 1$. At most $|VID|$ generalizations satisfy this "only if" condition, where $|VID|$ denotes the number of attributes in VID. ■

By checking the "only if" condition in Theorem 1, we can prune the computation of $A_G(VID)$ for all but at most $|VID|$ generalizations, and are still guaranteed to find all critical generalizations. Note that $|VID|$ is a very small constant, for example, 3 in Example 1. We implement this pruning strategy in three steps.

5.2 Step 1: pruning generalizations

This step finds all generalizations satisfying the "only if" condition in Theorem 1, denoted $Cand$. We start at the leaf nodes for the anonymity vids in the TEA index, walk up their paths synchronously one level at a time. At each level, we check if every anonymity vid is generalized by some size-k segment of the *same* generalization G, $k > 1$. If not, no critical generalization exists at the current level. If yes, we add G to $Cand$. We then move up to the next level in the TEA index.

5.3 Step 2: finding the best generalization

This step finds the best generalization by computing $IP(G)$ for *every* (valid) generalization G. $A(VID)$ and $I(G)$ are available or easily computed from the result of the previous iteration. For every G not in $Cand$, G is non-critical (Theorem 1), so $IP(G) = I(G)$. So, we focus on computing $A_G(VID)$ for $G \in Cand$. We present a method that examines only the vids actually generalized by G, not all vids.

Let A^n_G be the minimum $a(vid)$ for the new vids produced by applying G. Let A^o_G be the minimum $a(vid)$ for all old vids not generalized by G. $A_G(VID) = min\{A^n_G, A^o_G\}$. To compute A^n_G, we apply G to the TEA index as described in Section 5.1, except that the effect is made permanent only if G is actually the best generalization.

To compute A^o_G, we keep track of the number of vids not generalized by G such that $a(vid) = i$, stored in $O[i]$, for $1 \leq i \leq K$. K is typically a few hundreds, so this is a small cost. Before applying G, $O[i]$ is available from the previous iteration. Each time a vid having $a(vid) = i$ is generalized by G, we decrement $O[i]$. At the end of applying G, $O[i]$ stores the correct value. Now, if $O[i] > 0$ for some $1 \leq i \leq K$, let A^o_G be the smallest such i. If $O[i] = 0$ for $1 \leq i \leq K$, we consider two cases: if $A^n_G \leq K$, then $A_G(VID) = A^n_G$; if $A^n_G > K$, then $A_G(VID) > K$, but such $A_G(VID)$ is never used in our metric.

The cost in this step is proportional to the number of vids generalized by G, not all vids.

5.4 Step 3: applying the best generalization

This step applies the best generalization G_{best} found in Step 2. If G_{best} is in $Cand$, we just make the effect of G in Step 2 permanent. If G_{best} is not in $Cand$, we apply G_{best} to the TEA index. In this case, $Cand$ must be empty, otherwise G_{best} must come from $Cand$ following the remark below Definition 4.

5.5 Analysis

The TEA index is typically smaller than the database because a vid may occur in multiple records, but is stored only once in the TEA index. Once the TEA index is created, the bottom-up generalization depends only on the TEA index, not the database. The number of iterations is bounded by the number of possible generalizations, which is equal to the number of non-leaf nodes in all hierarchies. The analysis below focuses on a single iteration.

Step 1 involves walking up the anonymity vids in the TEA index. This cost is bounded by the number of anonymity vids, which is typically small because of the constraint $a(vid) = A(VID)$. Step 2 and 3 together apply at most $|VID|$ generalizations (Theorem 1). The cost of applying a generalization is bounded by the number of vids actually generalized, not the number of all vids.

6 Experimental Validation

Our first objective is to evaluate the quality of generalized data for classification, compared to that of the unmodified data. Our second objective is to evaluate the scalability of the proposed algorithm. All experiments were performed on a 2.4GHz Pentium IV processor with 512MB memory. The implementation language is C++.

6.1 Data quality

We adapted the publicly available "Adult" data [1], used previously in [7]. "Adult" has 6 continuous attributes and 8 categorical attributes. The class label represents two income levels, $\leq$50K or $>$50K. There are 45,222 records without missing values, pre-split into 30,162 and 15,060 records for training and testing. We used the same 7 categorical attributes used in [7], shown in Table 1, and obtained their hierarchies from the authors of [7]. [7] also used the numeric attribute Age. We did not include Age because our current algorithm handles only categorical attributes. In effect, this data is equivalent to their data with Age generalized into ANY in advance. This puts us in a non-favorable position because we do not have other choices of generalizing Age. VID contains all 7 attributes.

We generalized the training set on VID and built a C4.5 decision tree on the generalized data. The found generalization was then applied to the testing set and the error E was collected on the generalized testing set. We compared this error with two errors. B denotes the "baseline error" where the data was not generalized at all, which is 17.4%. W denotes the "worst error" where all the attributes in VID were generalized to ANY, which is 24.6%. $W - B$ measures the

[1] http://www.ics.uci.edu/m̃learn

Hierarchy	# Leaf nodes	# Levels
Occupation	14	3
Education	16	5
Country	41	4
Martial_status	7	3
Sex	2	2
Race	5	2
Work_class	8	3

Table 1. The hierarchies for "Adult" data

contribution of the attributes in VID, and $E - B$ measures the quality lost by our generalization. Figure 3 shows E for various thresholds K. Below are the main findings.

First, $E - B$ is no more than 1.1% for all K in the range from 10 to 500, which is significantly lower than $W - B = 7.2\%$. On one hand, the large $W - B$ implies that the attributes generalized are important. On the other hand, the small $E - B$ implies that the generalizations required for the K-anonymity does not harm the quality much. We observed that most generalizations tended to focus on overfitting values for the K tested, and if a generalized attribute became less discriminating, some previously unused alternatives emerged and were picked by C4.5 classifiers. Our approach takes advantage of such "health generalizations" and "multiple structures" typically present in the data for masking sensitive information while preserving quality.

Second, our results are comparable to the best results in [7] but take much less time. [7] reported the errors from 17.3% to 18.5% for K up to 500, with the baseline error of 17.1%. Our errors ranged from 18.4% to 18.5%, but our data has the baseline error of 17.4%. The error increase relative to the baseline is similar in both cases. On the other hand, our algorithm took no more than 7 seconds to create the index and no more than 22 seconds to generalize the data for all K tested, whereas the genetic algorithm took 18 hours as reported in [7].

6.2 Scalability

This experiment evaluated the scalability of the proposed algorithm by enlarging the "Adult" data. First, we merged the training set and testing set into one set, which gave 45,222 records. For each original record t in the merged set, we added $\sigma - 1$ "variations" of t, where $\sigma > 1$ is the *scale factor*. A variation of t took random values on ρ attributes randomly selected from VID, and agreed with t on the remaining attributes in VID. ρ is called the *novelty factor*. Random values came from the leaves in the corresponding hierarchy. The enlarged data has the 45,222 original records plus all variations, giving a total of $\sigma * 45,222$ records. We used σ and ρ to control the number of distinct vids.

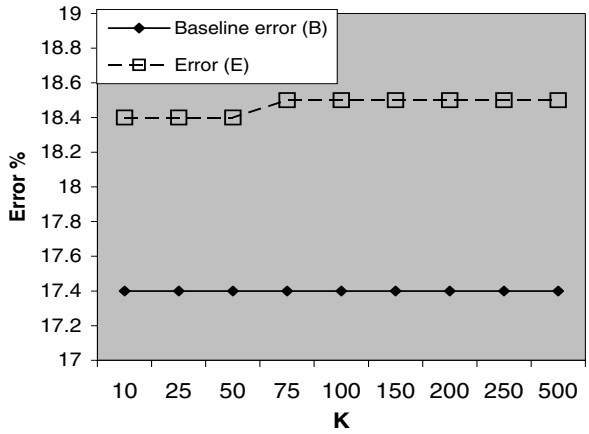

Figure 3. Error versus K

Figure 4 (from top to bottom) plots the time versus the threshold K, scale factor σ, and novelty factor ρ. Another 50 seconds or less were spent on creating the index. As one parameter varied, the other two were fixed. "Pruning-based" refers to the implementation that uses the pruning discussed in Section 5. "Index-based" refers to the implementation that uses the TEA index for performing a generalization, but not pruning.

In all experiments, both methods finished in less than 730 seconds. The longest time was took at $\sigma = 30$, $K = 150$ and $\rho = 3$ (the middle figure) where the data has $45,222 * 30 = 1,356,660$ records and 127,831 distinct vids. These experiments showed a much better scalability than the genetic algorithm in [7].

The first figure shows that K has some but not major effect on the time. The second figure shows that "pruning-based" scales up much better than "index-based" for a large scale factor. In this experiment, we observed that the scale factor has more impact on scalability than the novelty factor in that it increased the number of distinct vids faster. When the number of distinct vids is large, the effectiveness of the pruning in "pruning-based" became more significant.

7 Conclusion

We have investigated data mining as a technique for masking data, called *data mining based privacy protection*. The idea is to explore the data generalization concept from data mining as a way to hide detailed information, rather than discover trends and patterns. Once the data is masked, standard data mining techniques can be applied without modification. Our work demonstrated another positive use of the data mining technology: not only can it discover useful patterns, but also mask private information.

In particular, we presented a bottom-up generalization for transforming specific data to less specific but seman-

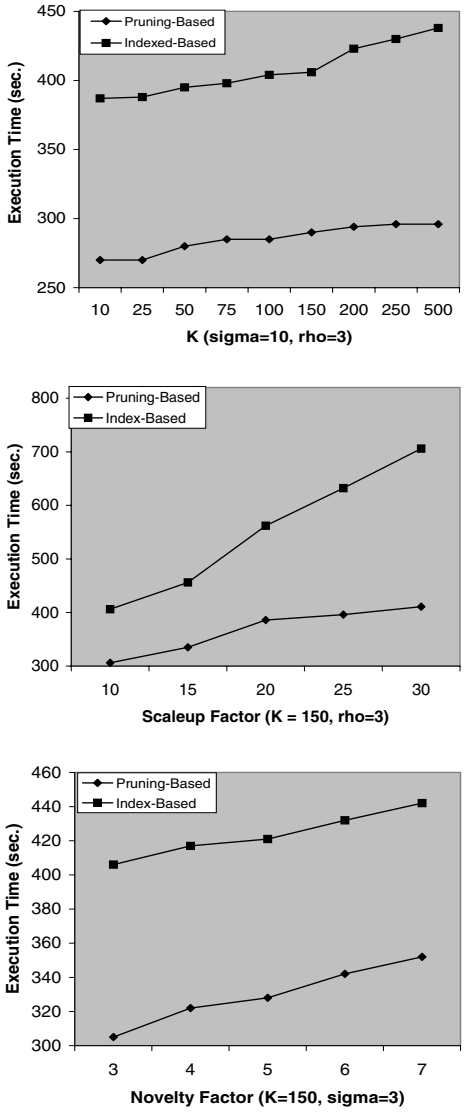

Figure 4. Scalability

tically consistent data for privacy protection. We focused on two key issues, quality and scalability. The quality issue was addressed by a trade-off of information and privacy and an iterative bottom-up generalization process. The scalability issue was addressed by a novel data structure for focusing on good generalizations. The proposed approach achieved a similar quality but much better scalability compared to existing solutions. Our current algorithm greedily hill-climbs a k-anonymity state, therefore, has the possibility of getting stuck at a local optimum. As suggested by one reviewer, local optimum can be escaped by introducing stochastic elements to this greedy heuristic or by using Simulated annealing. We plan to study this possibility in future work.

We believe that the framework of bottom-up generalization is amenable to several extensions that will make it more practical: incorporating different metrics, handling data suppression where a value is taken out entirely, and partial generalization where not necessarily all child values are generalized altogether, and generalizing numeric attributes without a pre-determined hierarchy. We plan to investigate these issues further.

Acknowledgement. Finally, we wish to thank the reviewers for constructive and helpful comments.

References

[1] R. Agrawal and R. Srikant. Privacy preserving data mining. In *SIGMOD*, 2000.

[2] C. F. Clark. The introduction of statistical noise to utility company data on a microdata tape with that data matched to annual housing survey data. In *Draft Project Report, Bureau of The Census, Washington, D.C.*, 1978.

[3] W. A. Fuller. Masking procedures for microdata disclosure limitation. *Journal of Official Statistics*, 9(2):383–406, 1993.

[4] B. Greenberg. Disclosure avoidance research at the census bureau. In *Proceedings of the 1990 Annual Research Conference of the Bureau of the Census, Washington, D.C.*, 1990.

[5] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: an attribute-oriented approach. In *VLDB*, 1992.

[6] A. Hundepool and L. Willenborg. μ- and τ-argus: software for statistical disclosure control. In *Third International Seminar on Statistical Confidentiality, Bled*, 1996.

[7] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *SIGKDD*, 2002.

[8] J. Kim and W. Winkler. Masking microdata files. In *ASA Proceedings of the Section on Survey Research Methods*, 1995.

[9] R. J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[10] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knownledge-based Systems*, 10(5):571–588, 2002.

[11] L. Sweeney. k-anonymity: a model for projecting privacy. *International Journal on Uncertainty, Fuzziness and Knownledge-based Systems*, 10(5):557–570, 2002.