Transforming Data to Satisfy Privacy Constraints

Vijay S. Iyengar
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598, USA
vsi@us.ibm.com

ABSTRACT

Data on individuals and entities are being collected widely. These data can contain information that explicitly identifies the individual (e.g., social security number). Data can also contain other kinds of personal information (e.g., date of birth, zip code, gender) that are potentially identifying when linked with other available data sets. Data are often shared for business or legal reasons. This paper addresses the important issue of preserving the anonymity of the individuals or entities during the data dissemination process. We explore preserving the anonymity by the use of generalizations and suppressions on the potentially identifying portions of the data. We extend earlier works in this area along various dimensions. First, satisfying privacy constraints is considered in conjunction with the usage for the data being disseminated. This allows us to optimize the process of preserving privacy for the specified usage. In particular, we investigate the privacy transformation in the context of data mining applications like building classification and regression models. Second, our work improves on previous approaches by allowing more flexible generalizations for the data. Lastly, this is combined with a more thorough exploration of the solution space using the genetic algorithm framework. These extensions allow us to transform the data so that they are more useful for their intended purpose while satisfying the privacy constraints.

General Terms

Privacy, data transformation, generalization, suppression, predictive modeling.

1. INTRODUCTION

Unprecedented amounts of data are being collected on individuals and entities. This is being fueled by progress in various technologies like storage, networking and automation in various business processes. Of particular interest are data containing structured information on individuals (referred to as micro-data in [11, 10, 14]). Such data are collected and used by various government agencies (e.g., U.S. Census Bureau and Department of Motor Vehicles) and by many commercial industries (e.g., insurance companies, health organizations, retailers). More and more data are also disseminated and shared within the organization collecting it and with other organizations. The dissemination could be to satisfy some legal requirements or as part of some business process. An important issue that has to be addressed is the protection of the privacy of individuals or entities referred to in the released micro-data [19, 20, 6].

An obvious step to protect the privacy of the individuals (or entities) is to replace any explicitly identifying information by some randomized placeholder. For example, a randomized token could replace the uniquely identifying social security number of a person in U.S.A. However, it has been pointed out that this is not sufficient since the released data contains other information which when linked with other data sets can identify or narrow down the individuals or entities [10, 15, 17, 14]. An example in [14] illustrates the identification by linking a medical data set and a voter list using fields like zip code, date of birth and gender.

In addition to the identity disclosure problem discussed above, attribute disclosure occurs when something about an individual is learnt from the released data [12]. Attribute disclosure can happen even without identity disclosure. Also, attribute disclosure in the broad sense can include inferential disclosure in which some characteristic of the individual can be inferred more accurately because of the data release [6]. Attributes whose disclosure needs to be protected in the strictest sense are denoted to be *sensitive* (e.g., physical or mental health of an individual) [19]. One approach to handling sensitive attributes is to exclude them from public use data sets [19]. This paper focuses on identity disclosure but related issues on inferential attribute disclosure will be be discussed where appropriate.

One approach to solving the identity disclosure problem is to perturb the data using techniques like adding noise and swapping values while ensuring that some statistical properties of the entire table are maintained [11]. The re-identification risk in data masked by such perturbation techniques is evaluated using a probabilistic formulation. Addition of noise and selective data swapping are used in [11] to generate masked data with small disclosure risk while preserving means and correlations between attributes, even in many sub-domains. The tradeoff between information loss and the re-identification risk using such perturbative methods is being actively researched [4, 21]. Data masked using only additive noise was used to generate classification mod-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGKDD '02 Edmonton, Alberta, Canada
Copyright 2002 ACM 1-58113-567-X/02/0007 ...$5.00.

els which were evaluated using a synthetic benchmark in [1]. For predictive modeling applications, further work is needed to quantify and evaluate the tradeoffs between model accuracy and the probabilistic disclosure risk on real data sets.

An alternative approach to solving this problem is to transform the data by using generalizations and suppressions [15, 17, 14]. An example of a transformation by generalization is to replace the exact date of birth by only the year of birth. The loss of specificity makes the identification process harder. Too much generalization could make the transformed data useless. For example, generalizing the date of birth to century containing it has little value when considering, say, current insurers. Suppression could be viewed as the ultimate generalization since no information is released. As before, the data transformation challenge is to find the right tradeoff between the amount of privacy and loss of information content due to generalizations and suppressions [10, 15, 17, 14].

This paper uses the approach of transforming the data using generalizations and suppression to satisfy the privacy constraints. We extend earlier works in this area along multiple dimensions. First, we consider the tradeoff between privacy and information loss in the context of the usage for which the transformed data is being generated. Examples of usage like building predictive models are used to illustrate how the usage targeting leads to better solutions. Next, we allow more flexible generalizations of the data that expand the space of possible data transformations potentially leading to better solutions. Lastly, we cast this search for the right tradeoff between privacy and information loss as a large-scale optimization problem and aggressively pursue its solution using a genetic algorithm framework.

2. BACKGROUND

In this section we will cover the high level formulation of the data transformation problem. We will use terminology from the earlier work by Samarati [14] where appropriate. Conceptually, the data to be transformed will be viewed as a table. The rows of this table represent individuals (entities) being described and the columns represent attributes of these individuals (entities). There are three types of columns based on their ability to identify individuals (entities). Some columns contain explicitly identifying information (e.g., social security number) which needs to be handled by replacement with an unlinkable token or by suppression. Some columns contain potentially identifying information that could be linked with other tables for the purposes of re-identification of the individuals. The set of these potentially identifying columns (or attributes) has been called a quasi-identifier in earlier works [14]. The remaining columns do not contain any identifying information.

The privacy level to be guaranteed will be specified by the notion of *k-anonymity* [14]. Intuitively, a transformed table satisfies k-anonymity if every combination of values occurring in the potentially identifying columns (quasi-identifier) cannot be matched to fewer than k rows. Specifying higher values for k results in stricter privacy requirements by making re-identification by linking harder.

The k-anonymity requirement is met by generalizing or suppressing values in the potentially identifying columns of the table. The most general form of this would allow individual entries (also called cells) in the table to be generalized or suppressed as needed. However, this approach would complicate the interpretation of the table in applications like data mining, because values in a single column of the transformed table could then have complex relationships. We will use a simpler transformation that generalizes all the entries in a potentially identifying column uniformly. This reduces the solution space to be considered and also fits well with current applications (e.g., predictive modeling tools). This simplification is also used by many earlier works [10, 17, 14]. Similarly, suppressions will be performed by masking out the contents of all the potentially identifying columns in each suppressed row as suggested earlier [17, 14]. A more general notion that allows specific entries (cells) to be suppressed has also been proposed [10].

We extend the earlier works [10, 15, 17, 14] by allowing more flexible generalizations as described next. The form of the allowed generalization depends on the type of data in a potentially identifying column. First, consider a column with categorical information (e.g., zip code, gender, race, marital status). Generalizations (coarsening) for such a column are typically described by a taxonomy tree. Consider an example of a taxonomy tree in Figure 1. The column corresponding to this tree contains information on an individual's type of work. The leaf nodes depict all the possible types of work: self-employed (incorporated), self-employed (not incorporated), federal government, state government, local government, private company, without pay and never worked. These can be grouped as shown in Figure 1 at the next level into self-employed, government, and unemployed. A valid generalization A in our work is represented by a set of nodes S_A in the taxonomy tree that satisfy the property that the path from every leaf node Y to the root encounters exactly one node P in S_A. The value represented by the leaf node Y is generalized in A to the value represented by the node P. This definition of a valid generalization is broader than earlier notions that required that all the generalized values had to be at the same level of the taxonomy tree. For example, {self-employed, federal government, state government, local government, private company, unemployed} represents a valid generalization using the taxonomy tree in Figure 1.

Next, consider a column with numeric information (e.g., age or education in years). Generalization for a numeric column is done by discretization of its values into a set of disjoint intervals. Each interval could be represented by a symbolic value that denotes the interval and is interpreted accordingly. Alternatively a numeric value could be chosen as a representative value for each interval (e.g., median value of entries in the original table that lie within the interval). The choice of representation for the discretized value depends on the application for which the transformed table is being generated. Allowed discretizations can be constrained by specifying the possible end points for the intervals. The choice of possible end points determines the granularity of the intervals explored during discretization. For example, if the number of unique numeric values in a column is reasonably small, an end point can be chosen between each pair of neighboring values. Alternatively, end points can be determined by applying a process like scalar quantization on the value space for this column. For example, the set of intervals {[0,20),[20,40),[40,60),[60,80),[80,∞)} is a valid discretization for a column containing the age (in years) of an individual.

Having defined the privacy goal (in terms of k-anonymity)

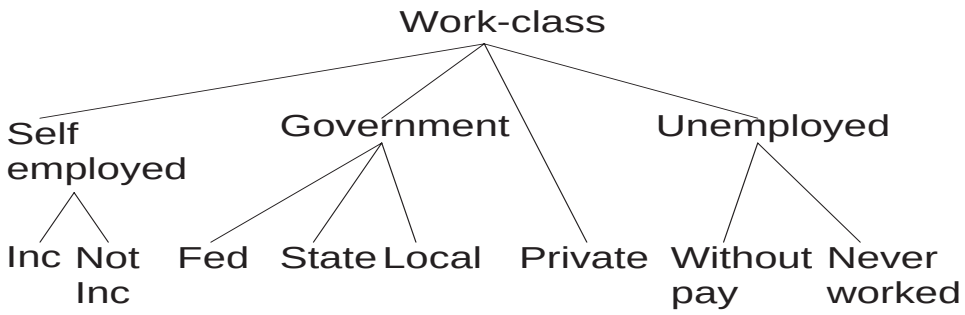

Figure 1: Taxonomy tree for a column containing information on type of work

and the valid generalizations we can now consider the task of transforming the table. Transforming the table using generalizations and suppressions results in loss of information content. We define metrics to quantify the loss of content as it pertains to the anticipated uses of the transformed table. Transformation of the table can then be formulated as optimization of the metric while satisfying the k-anonymity constraint. This is illustrated using various usage examples including the generation of predictive models common in data mining tasks. The following section defines these usage based metrics.

3. USAGE BASED METRICS

In this section, specific uses for the transformed table are considered and the corresponding loss metrics defined. These metrics will be used in the optimization formulation in Section 4. Experimental results in Section 5 illustrate the value of usage based metrics.

3.1 Multiple Uses

The most general case is when the data is being disseminated for multiple uses. We also include in this case the situation when the usage is unknown at the time of dissemination. For this case we define a metric which captures some general notion of information loss as was done in the earlier works [15, 17, 14]. Our metric differs from the earlier ones because it has to handle the more flexible generalizations allowed in our work.

Information in all the potentially identifying columns will be assumed to be equally important in this case. So the total information loss due to generalizations and suppressions (represented by a general loss metric LM) will be computed by summing up a normalized information loss for each of these columns. We will define the loss computation for each type of column next. This information loss for a column will be computed as the average loss for each entry in the column.

Consider a potentially identifying column containing categorical information that is generalized based on a taxonomy tree T (e.g., Figure 1). Consider an entry in this column (e.g., State Government) where the generalized value in the transformed table corresponds to node P (e.g., Government) in the taxonomy tree T. One approach is to quantify the loss when a leaf node value cannot be disambiguated from another value due to the generalization. For example, generalizing the value *State Government* to the value *Government* implies that it cannot be disambiguated from the values *Fed Government* and *Local Government*. The associated loss can be modeled as shown in Figure 2. In our experiments, we simplify the model by assuming the same generalization loss for ambiguity between any two distinct categorical values. Let the total number of leaf nodes in T be denoted by M. Let the number of leaf nodes in the subtree rooted at node P be M_P. Using this simplified model and normalizing using the worst case situation when the generalized node is the root of the taxonomy tree leads to $(M_P - 1)/(M - 1)$ as the loss for this entry. For the earlier example, the normalized loss is $2/7$ when the value *State Government* is generalized to *Government*. The loss for a suppressed entry is the same as the loss when the generalized value corresponds to the root of the tree.

Next, consider a potentially identifying column containing numeric information. As before we can use a notion of ambiguity in defining the information loss for a numeric entry. Consider a entry which is generalized to an interval i defined by the lower and upper end points L_i and U_i, respectively. Let the lower and upper bounds in the table for values in this column be L and U, respectively. The normalized loss for this entry is given by $(U_i - L_i)/(U - L)$. For example, consider the attribute *education* with the mapping to numeric values given in Figure 3. Generalizing the value Doctorate to the interval {[Doctorate, Masters]} has a normalized loss given by $2/15$. A suppressed row can be viewed as being maximally generalized to an interval with the column bounds as its end points. The loss for the column is computed by averaging the loss for each of its entries.

3.2 Predictive Modeling Use

One possible use for the transformed table is to build predictive models for some attribute. For example, a manu-

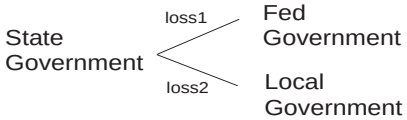

Figure 2: Loss due to generalization of categorical attribute value

1. Doctorate	9. 12th
2. Professional school	10. 11th
3. Masters	11. 10th
4. Bachelors	12. 9th
5. Associate (vocational)	13. 7th-8th
6. Associate (academic)	14. 5th-6th
7. Some college	15. 1st-4th
8. High School grad	16. Preschool

Figure 3: Mapping to numeric values for the attribute *education*

facturer might be interested in modeling the customers who are interested in a specific category of products. The customer profile and transaction data that are collected by a retailer could be used to build such models. A key question is whether we can build these models accurately while satisfying constraints on identity disclosure. The model's accuracy is dependent on the information loss due to generalizations and suppressions. We will define metrics that measure the loss of the purity in the target variables (for predictive modeling) due to aggregation caused by the generalizations. This is a conservative approach since it does not try to tailor the metric to any specific predictive modeling method.

First, we will consider building a classification model where the class label information is in one of the columns of the table. The columns allowed as inputs to the classification model are specified and can include potentially identifying columns.

Generalizing or suppressing the content in potentially identifying columns weakens the discrimination of classes using these columns. The k-anonymity constraint forces multiple rows (at least k) in the table to have the same combination of generalized values for the potentially identifying columns. All rows with a unique combination of generalized values will be said to belong to the same group. For each row r, let $G(r)$ denote the group to which it belongs. Rows in a group G with different class labels cannot be discriminated using the potentially identifying columns. Therefore, for accurate classification, it is preferable if all the rows in G have the same class label. For now we will ignore the fact that information in the non-identifying columns might be able to discriminate the rows in G with different labels. This will be discussed in Section 6. Therefore, the metric for this usage will penalize impure groups that contain rows with different labels.

The classification metric CM is defined in Equation 1 as the sum of the individual penalties for each row in the table normalized by the total number of rows N.

$$CM = \frac{\sum_{all\ rows} penalty(row\ r)}{N} \quad (1)$$

A row r is penalized if it is suppressed or if its class label $class(r)$ is not the majority class label $majority(G)$ of its group G (see Equation 2).

$$penalty(row\ r) = \begin{cases} 1 & \text{if } r \text{ is suppressed} \\ 1 & \text{if } class(r) \neq majority(G(r)) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

This is illustrated using an example that has two potentially identifying attributes (X,Y) with numeric values (in the range 0 to 10) in Figure 4. The two classes in this example are marked by the symbols $*$ and $\square$. The solid lines indicate a solution where {[0,7.5],(7.5,10]} is the generalization for attribute X, and {[0,3.5],(3.5,10]} is the generalization for attribute Y. Consider this solution when the k-anonymity constraint specified has $k = 5$. None of the rows have to be suppressed. The row with $X = 9, Y = 9$ contributes a penalty of 1 because its class label $*$ is not the majority label $\square$ in its group (group defined by $X\epsilon(7.5, 10]$ and $Y\epsilon(3.5, 10]$). The CM metric has value 0.15 for this case, since 3 out of the 20 points do not have the majority class in their groups.

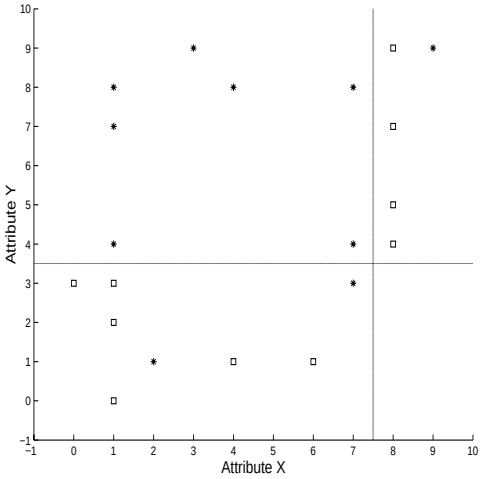

Figure 4: Example with numeric attributes X and Y

The classification metric can be extended to incorporate a cost matrix that indicates the cost of misclassifying a data point with class C1 as having class C2. This is done by modifying the penalty function to reflect the cost of misclassifying a row r from its original class to the majority class of its group. The suppressed rows can also be treated as a separate group in this case.

Next, consider using the transformed table to build a regression model for a dependent variable V that is one of the

columns of the table. The columns that are allowed as inputs to this model are specified and can include potentially identifying columns.

Using the earlier notion of a group based on unique combination of values for potentially identifying columns, the modeling of the dependent variable is impacted by the variability (purity) of its value within the groups. The purity of each group can be quantified by any measure of dispersion. In particular, a classical measure like variance or a robust measure like absolute deviation from the median are good candidates. The regression metric is normalized by the dispersion when all the rows are in a single group.

If there are multiple target variables for predictive modeling, their normalized metrics can be combined using a weighted sum based on a user-defined weighting for each target.

4. SOLVING THE OPTIMIZATION PROBLEM

The metrics defined in the earlier section can be used to measure the loss due to generalizations and suppressions as it pertains to the specified usage of the transformed table. The problem to be solved is to minimize the loss as indicated by the chosen metric while satisfying the specified k-anonymity constraint. The more flexible generalizations allowed in our formulation lead to a larger space of possible solutions that must be considered while solving this optimization problem. This factor, along with the variety in the metrics to be optimized, motivated the use of a general framework for optimization. The *genetic algorithm* framework was chosen because of its flexible formulation and its ability to find good solutions given adequate computational resources [8, 7]. Clearly, this randomized approach does not guarantee finding a globally optimal solution.

Genetic algorithms are iterative optimization procedures that mimic the natural process of evolution. These algorithms work on a set of solutions to a problem (called population). Each solution in the population is represented by a string (called chromosome). In each iteration new solutions are generated from the population in an attempt to obtain better solutions. The notion of a better solution is based on some problem specific metric. Two of the key operations in the iterative process are crossover (combine portions of two solutions to produce two other solutions) and mutation (incrementally modify a solution).

The specific form of the genetic algorithm used in our application is based on the GENITOR work [18]. Extensions to this work that were necessary for our application will be described later. A solution in our application has two parts: the generalizations chosen for the potentially identifying columns and the rows that are suppressed. The chromosome in the genetic algorithm (GA) framework is a bit string that represents the chosen generalizations. The metric computation to determine the goodness of a solution computes the suppressions that are needed if the solution's generalizations are applied to the table.

The chromosome bit string is composed by concatenating the bit strings corresponding to each potentially identifying column. Consider a potentially identifying column with numeric values. The number of bits in the bit string corresponding to a numeric column depends on the granularity at which the generalization intervals are defined. The first step is to define the values that could be used as end points for the generalization intervals. For a column with a small number of possible values a potential end point could be added in between each pair of successive values. For numeric columns with too many values potential end points could be chosen by some discretization procedure (e.g., [5, 9]). The bit string for a numeric column is made up of one bit for each potential end point in value order. A value of 1 for a bit implies that the corresponding value is used as an interval end point in the generalization.

Consider a categorical column with D distinct values which are generalized using the taxonomy tree T. The number of bits needed for this column in the chromosome is $D - 1$ which are assigned as described next. The distinct column values, which are represented by the leaf nodes of the tree T, are arranged in the order resulting from an in-order traversal of T. The leaf nodes from left to right as shown in Figure 1 conform to an in-order traversal of that taxonomy tree. The chromosome bits are assigned to the positions between successive values (leaf nodes) in this order. This is pictorially depicted in Figure 5. There are 7 chromosome bits ($b1$ through $b7$) allocated for this column. For example, bit $b3$ corresponds to the position between the leaf node values, federal government and state government. A value of 1 for this bit position implies that these two values are separated in the corresponding generalization. However, unlike the case of numeric columns, only some combination of values for the bit string correspond to valid generalizations for the categorical column. Recall that the nodes representing a valid generalization must satisfy the property that each leaf node encounters exactly one generalization node on the path to the tree root. Considering our example, if bit $b3$ is 1, then this implies that bits $b2$, $b4$, $b5$, and $b6$ are also 1. The genetic algorithm [18] has to be extended to ensure that the chromosomes in the population represent valid generalizations for the categorical columns. This is done by an additional step that modifies newly generated chromosomes that are invalid into valid ones while retaining as much as possible of the original characteristics.

The genetic algorithm used [18] requires choosing some parameters like the size of the population, the probability of mutating a bit in the chromosome and the number of iterations to be run. These parameters are typically chosen based on some experimentation. Varying the number of iterations is an easy way of controlling the tradeoff between the quality of the solution and computational cost of the algorithm. The population should be large enough to have adequate variety in it for the evolutionary process. Typical choices for these parameters are illustrated using an example in the next section.

5. EXPERIMENTS

We have adapted a publicly available classification benchmark for the experiments. The *adult* benchmark in the UCI repository [2] is based on census data and has been used widely in classification experiments. For our experiments we retain only eight of the original attributes that are all considered to be potentially identifying. These are *age*, *work class*, *education*, *marital status*, *occupation*, *race*, *gender* and *native country*. The binary attribute *salary class* (salary above or below 50,000) is also retained. Records with missing values are discarded because of limitations in our prototype system. This leaves 30162 records in the training set consti-

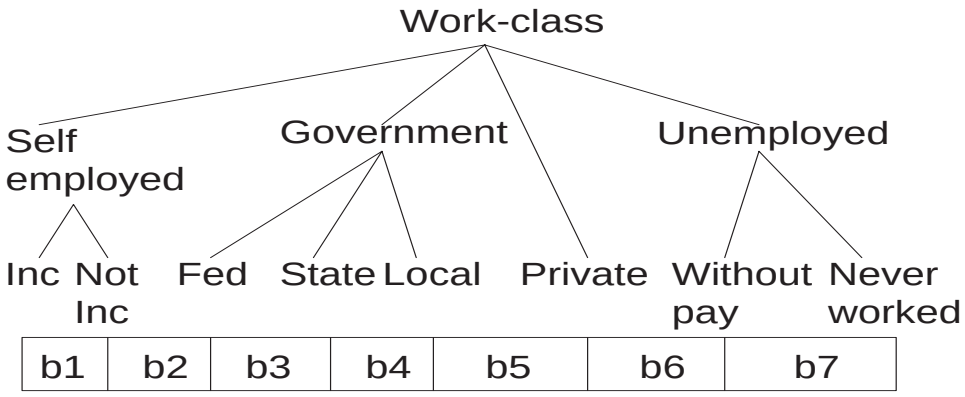

Figure 5: Taxonomy tree from Figure 1 with the chromosome bits

tuting the original table for these experiments.

Two of the attributes, *age* (in whole years) and *education*, are treated as numeric quantities. The mapping to numeric values for the attribute *education* is given in Figure 3. Since the number of distinct values is relatively small for these two numeric attributes, we include a potential interval end point between each pair of successive values. The other six attributes are treated as categorical attributes with a taxonomy tree defining the allowed generalizations for each of them. The taxonomy tree for the *work class* attribute was introduced earlier in Figure 1. Trees for the other categorical attributes are not shown due to space constraints. The allowed generalizations lead to a chromosome of length 157 bits.

Our genetic algorithm was applied to the training set described above. All runs used a chromosome population size of 5000 and ran 0.5 million iterations of the genetic algorithm. The probability of mutating any bit of the chromosome in an iteration was set at 0.002. In experiment 1, the usage for the transformed table was building a classification model for the *salary* class column. The k-anonymity privacy constraint was varied by having k take the values 10, 25, 50, 75, 100, 150, 200, 250 and 500. The optimized classification metric CM achieved for these values of k is plotted as the solid line in Figure 6. It is interesting to note that there is little degradation in the CM metric as the privacy requirement ranges from $k = 10$ to $k = 500$. Also, low values for the CM metric (around 0.18) suggest that the algorithm is able to find good transformations in the context of this usage criterion. For example, the solution for $k = 250$ generalizes away all the information in the attributes *work class*, *occupation*, *race*, *gender* and *native country* (i.e., values in these columns are generalized to the root of the corresponding taxonomy trees). The other attributes are generalized as follows: *age* by {[0,39],(39,∞)}, *education* by {[Doctorate, Bachelors],[Associate (vocational), Some college], [High school grad, Preschool]}, and *marital status* by {Married, Was married, Never married}. Note that the generalization for the numeric attribute *education* is represented using intervals based on the mapping given earlier in Figure 3.

The general loss metric LM was also computed for each solution in this experiment and is plotted as the dashed line in Figure 6. The non-monotonic behavior is easily explained by the fact that the algorithm did not try to optimize the LM metric in this experiment. However, it is interesting to note the very poor values (around 0.8) for the LM metric of these solutions. This illustrates that solutions targeted at one usage need not be suitable for some other purpose.

In experiment 2, the generalized loss metric LM was optimized for various levels of the privacy constraint. The LM metric achieved is shown by the dashed line in Figure 7. In contrast to the poor LM values achieved in experiment 1, the LM values now range from 0.21 to 0.49 as k ranges from 10 to 500. There is a clear tradeoff between the level of privacy (indicated by k) and the loss of information as measured by LM. For example, consider the numeric attribute *education*. At the privacy level of $k = 25$, the chosen generalization for this attribute is {[Doctorate, Professional school],[Masters, High School grad], [12th, 1st-4th]}. Values not included were suppressed (e.g., Preschool). At the more restrictive privacy level of $k = 250$, this attribute is generalized much further using just the interval {[Professional school, 7th-8th]}. Again, values not included in the generalization above were suppressed.

The classification metric CM was also computed for the solutions in experiment 2 and is shown as the solid line in Figure 7. The values for CM display non-monotonic behavior and fall in the range from 0.3 to 0.4. Clearly, these values are poorer than those achieved in experiment 1 when CM itself was optimized.

The transformed data sets produced in these two experiments can also be compared by using them to generate classification models for the binary attribute *salary*. Tree models are generated using the C4.5 [13] classifier using the transformed data sets from experiments 1 and 2. The results

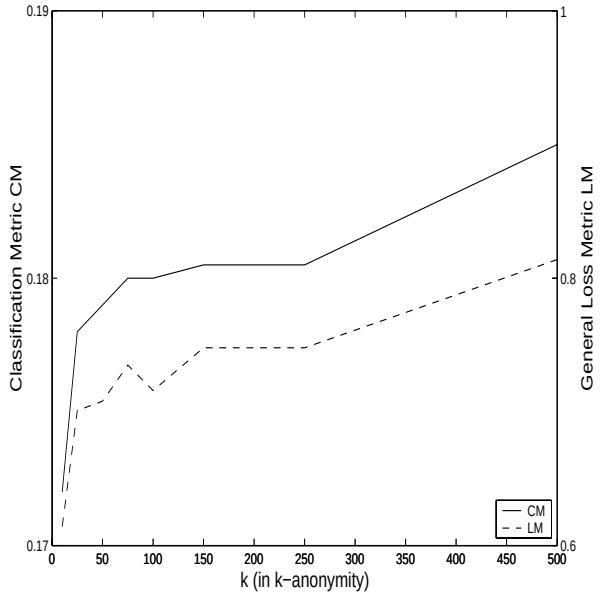

Figure 6: Results achieved optimizing the classification metric CM (Experiment 1)

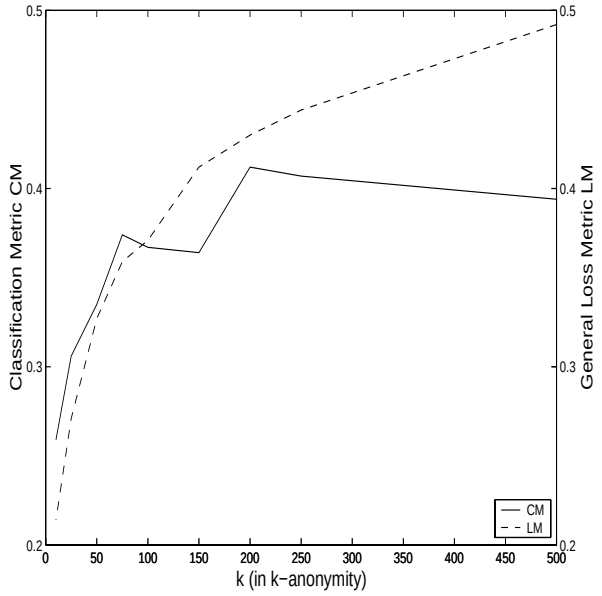

Figure 7: Results achieved optimizing the general loss metric LM (Experiment 2)

are presented in Figure 8. The classifier error reported is the estimate based on 10-fold cross validation. The classification errors for the anonymized data produced by optimizing the CM metric (experiment 1) are in a relatively tight range from 17.3% to 18.5%. This should be compared with the corresponding error of 17.1% for the original data (without any transformations for anonymity). The classification errors are significantly higher for the data sets produced by optimizing the LM metric and reach the maximum possible value at higher values of k. Specifically, for k values of 200 and above the classifiers generated using data from experiment 2 misclassify all the minority class records.

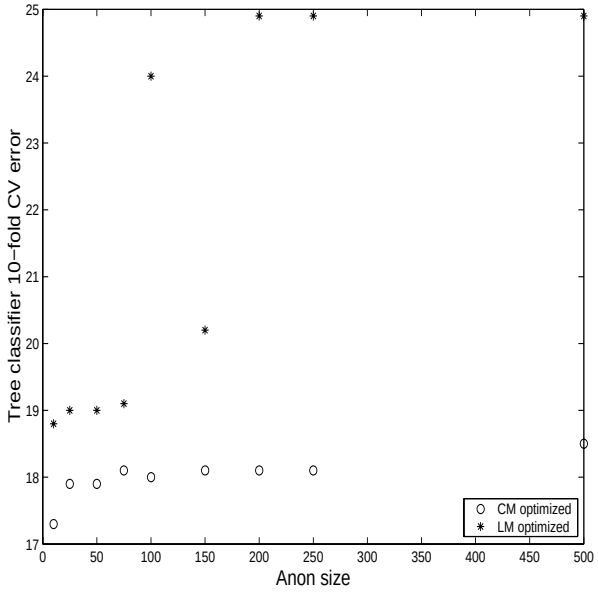

Figure 8: Tree classifier errors using transformed data(using 10-fold cross validation)

Our experiments illustrate the value of using metrics tailored to the purpose for which the data is disseminated. These results can also be viewed as indicating the difficulty in producing truly multi-purpose data sets satisfying k-anonymity requirements. This issue needs to be investigated using the other approach to anonymization, namely, by adding noise and swapping [11]. Such a comparative analysis would be useful to determine the preferred approach for generating multi-purpose anonymized data sets.

The genetic algorithm framework is able to handle the larger solution space that results from the more flexible generalizations considered. Figure 9 shows the performance of the genetic algorithm by plotting the CM metric of the best solution for $k = 100$ (solid line) against the number of iterations run. As expected, the improvement in the best solution is rapid at the beginning and gradual later. In contrast, the number of unique solutions (dashed line) continues to increase after 500K iterations. This implies that the solution space continues to be explored but the locally optimal solution found is not improved. Similarly, Figure 10 shows the performance when the general loss metric LM is optimized for $k = 100$.

All the experiments discussed so far correspond to the situation where some existing data has to be transformed to preserve privacy constraints. The *adult* training set was

used as the representative data in these experiments. It is possible that after the initial data transformation and dissemination, data on new individuals or entities (i.e., new rows in the table) become available for another release. It is interesting to investigate the effectiveness of applying to the new table the data generalizations chosen by the analysis of the original table. This would allow the transformed tables to be concatenated for analysis with a single set of generalizations for the columns. We used the test set of the same *adult* benchmark as the representative second table to be released. The smaller test set had 15060 records (after discarding records with missing values) and the same set of attributes used in all the earlier experiments. The genetic algorithm was applied to the test set to optimize both LM and CM metrics for various levels of the privacy constraint. These optimized metric values are compared with the corresponding values achieved by reusing the data generalizations gotten from the training set. Figures 11 and 12 present the comparisons for the CM and LM metrics, respectively. The solid lines in these figure indicate when the metrics are equal and provide a reference for comparison. As expected, the metric values achieved by using the earlier generalizations are worse than the optimized values. The more pronounced degradation in the CM metric was primarily due to increased suppressions when earlier generalizations were reused. This type of analysis is useful in determining if the degradation in metric is offset by the benefits of having the same generalizations across multiple tables.

For most applications we do not see any real time requirements for the task of transforming the data to satisfy privacy. A computationally intensive approach like the genetic algorithm framework was chosen because the quality of the solution is far more important than the time taken to generate it. Our prototype system took 18 hours to transform the training set with 30K records using 500K iterations on an IBM Model 6868 Intellistation (1 GHz Pentium III processor, 1GB memory). Very large data sets can be handled by running the algorithm on a random sample of the records and then applying the generated generalizations to the whole data set. However, as illustrated in Figures 11 and 12, this will cause some degradation in the quality of the solution generated.

6. DISCUSSION

The earlier section demonstrated the viability of our approach using an example with eight potentially identifying attributes. In general, the size of the solution space depends on the number of such attributes and the granularity at which they need to be considered. Determining which attributes should be considered as potentially identifying is based on an assessment of possible links to other available data. This needs to be done with typical databases in each domain (e.g., retail). Clearly, as the number of potentially identifying attributes grows, identity disclosure risk increases. The corresponding increase in the number of unique combinations of potentially identifying values will have an impact on the k-anonymity approach. Also, the complexity of the optimization problem increases due to the the larger solution space to be searched. Further experiments are needed to investigate the applicability of this approach to wider data sets.

In each domain, in addition to the identifying attributes one needs to determine the sensitive attributes. It has been suggested that sensitive attributes be removed completely from data sets being publicly released [19]. Further work is needed to determine adequate ways of handling these attributes if they are included. Clearly, they cannot be targets of predictive modeling using our methods since that will result in their inferential disclosure. This is because the optimization we perform for predictive modeling would group together rows with similar values for the target attribute. This optimization improves the model accuracy while satisfying the identity disclosure constraint, but it also increases the inferential attribute disclosure for the sensitive attribute being targeted. While this is an explicit issue with the k-anonymity approach to anonymization, further investigation is needed on issues related to the inferential disclosure of sensitive attributes even for other approaches (e.g., additive noise and swapping).

In many cases only a sample of the data is released. The privacy protection due to sampling has been considered in various works (e.g., [6, 16, 3]). Applying the k-anonymity approach to the release of a sample opens up some new issues. One approach could be to require that the released sample satisfy the k-anonymity requirement. The choice of k would have to be made taking into account the sampling effect. Alternatively, the k-anonymity requirement could be first applied to the entire population before a sample of the transformed table is released. The sizes of the groups in the released sample will depend on the form of sampling used (e.g., random, stratified). Further work is needed to explore the k-anonymity approach in the context of sampling.

For predictive modeling usage the metrics defined in Section 3 consider predictability using only the potentially identifying attributes. This was done independent of the predictive capabilities of the other non-identifying attributes. Considering both identifying and non-identifying attributes during the data transformation process could lead to better solutions. Finding an effective way of doing this with potentially large numbers of non-identifying attributes needs further exploration.

7. CONCLUSION

We have addressed the important problem of transforming data so that the dual goals of usefulness and privacy can be satisfied. The data transformation was done by generalizing or suppressing potentially identifying content in the data. Usage based metrics were defined to quantify the loss of information due to the transformations tailored to the data usage. Usages considered included classification and regression modeling, frequently employed in data mining applications. The data transformation problem was then solved by using the genetic algorithm framework to optimize the appropriate metric. Experimental results were presented using a benchmark based on census data. These results demonstrated the viability of our approach and also the benefits of the usage based metrics. We also discuss the limitations of our approach and several open problems in this area.

8. ACKNOWLEDGMENTS

We would like to thank Murray Campbell, Vittorio Castelli and Anshul Gupta for helpful discussions and an anonymous reviewer for detailed and insightful comments. We also gratefully acknowledge the support provided by DARPA (Project F30602-01-C-0184).

9. REFERENCES

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of ACM SIGMOD Conference on Management of Data*, 2000.

[2] C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Science, URL=http://www.ics.uci.edu/∼mlearn/-MLRespository.html,1998.

[3] G. Chen and S. Keller-McNulty. Estimation of identification risk in microdata. *Journal of Official Statistics*, 14(1):79–95, 1998.

[4] J. Domingo-Ferrer, J. Mateo-Sanz, and V. Torra. Comparing SDC methods for microdata on the basis of information loss and disclosure risk. In *Proceedings of NTTS and ETK*, 2001.

[5] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of Twelfth International Conference on Machine Learning*, 1995.

[6] G. Duncan and D. Lambert. Disclosure-limited data dissemination. *Journal of the American Statistical Association*, 81(393):10–28, 1986.

[7] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[8] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[9] S. Hong. Use of contextual information for feature ranking and discretization. *IEEE Transactions on Knowledge and Data Engineering*, 9(5):718–730, 1997.

[10] A. Hundepool and L. Willenborg. μ- and τ- argus: Software for statistical disclosure control. In *Proceedings of Third Internation Seminar on Statistical Confidentiality*, 1996.

[11] J. Kim and W. Winkler. Masking microdata files. In *ASA Proceedings of the Section on Survey Research Methods*, pages 114–119, 1995.

[12] D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9(2):313–331, 1993.

[13] J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[14] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge Engineering*, 13(6):1010–1027, 2001.

[15] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report Technical Report, SRI International, March 1998.

[16] C. Skinner. On identification disclosure and prediction disclosure for microdata. *Statistica Neerlandica*, 46(1):21–32, 1992.

[17] L. Sweeney. Datafly: A system for providing anonymity in medical data. In *Proceedings of Eleventh International Conference on Database Security*, pages 356–381. Database Security XI: Status and Prospects, 1998.

[18] D. Whitley. The genitor algorithm and selective pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of Third International Conference on Genetic Algorithms*, pages 116–121. Morgan Kaufmann, 1989.

[19] L. Willenborg and T. D. Waal. *Statistical Disclosure Control in Practice*. Springer-Verlag, 1996.

[20] L. Willenborg and T. D. Waal. *Elements of Statistical Disclosure Control*. Springer-Verlag, 2000.

[21] W. Yancey, W. Winkler, and R. Creecy. Disclosure risk assessment in perturbative microdata protection. Technical Report Research Report Statistics 2002-01, Statistical Research Division, U.S. Bureau of the Census, 2002.

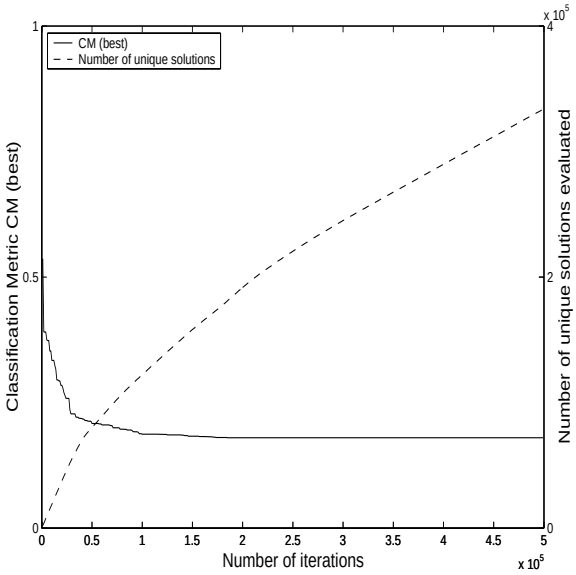

Figure 9: Performance of algorithm optimizing CM (k = 100)

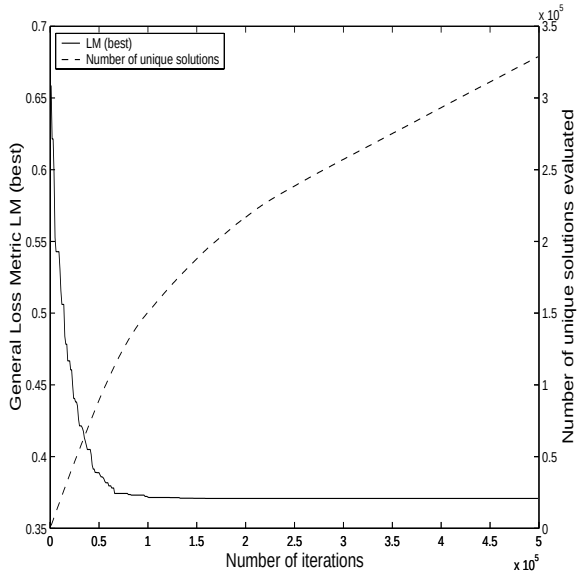

Figure 10: Performance of algorithm optimizing LM (k = 100)

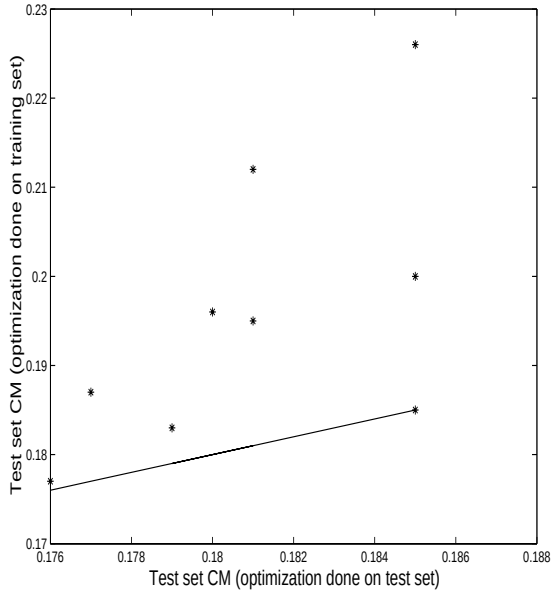

Figure 11: Effectiveness (CM) of reusing earlier generalizations on new data

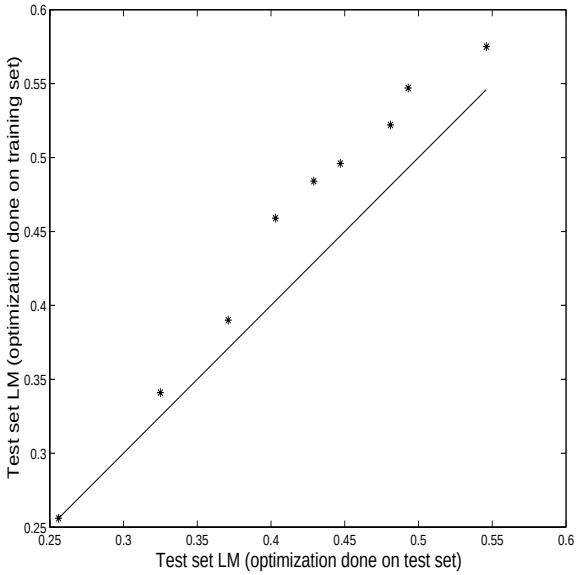

Figure 12: Effectiveness (LM) of reusing earlier generalizations on new data