TRANSACTIONS ON DATA PRIVACY 5 (2012) 469 - 503

An Enhanced Utility-Driven Data Anonymization Method

Stuart Morton*, Malika Mahoui*, P. Joseph Gibson and Saidaiah Yechuri***

* School of Informatics IUPUI, 719 Indiana Ave, WK 307, Indianapolis, IN 46202
** Marion County Public Health Department, 3838 North Rural Street, Room 721, Indianapolis, IN 46205
E-mail smmorton@iupui.edu, mmahoui@iupui.edu, jgibson@hhcorp.org, yechuris@iupui.edu

Abstract. As medical data continues to transition to electronic formats, opportunities arise for researchers to use this microdata to discover patterns and increase knowledge that can improve patient care. We propose a data utility measurement, called the research value (RV), which reflects the importance of an database attribute with respect to the other database attributes in a dataset as well as reflect the significance of the content of the data from a researcher's point of view. Our algorithms use these research values to assess an attribute's data utility as it is generalizing the data to ensure k-anonymity. The proposed algorithms scale efficiently even when using datasets with large numbers of attributes.

1 Introduction

With the advances made in technology during the last few decades, health organizations have amassed large amounts of electronic, health related data. This data constitutes a valuable resource for researchers, analysts and decision makers. For example, epidemiologists may use emergency visits to detect potential arising outbreaks that need to be further investigated and appropriate actions can be taken in a timely manner. Health related information is also made available to the general public as a contribution to public health awareness and education. For example, electronic birth and death certificates may: 1) provide a rich source for researchers investigating risk factors for infant deaths or other poor birth outcomes, 2) provide advocates, health care providers, and government or nonprofit agencies with specific local information about maternal and child health issues, and 3) help guide policy development. Like other health departments across the country, the Marion County Public Health Department (MCPHD) of Indiana provides the public with access to Datamart, an Internet application that presents aggregate birth and death certificate data [6]. Users

may obtain summary information on features such as birth risk factors aggregated by year (since 1997), by census tract, by race, etc.

In order to preserve the anonymity of statistical data, two main approaches have been adopted: restricting the query capabilities, (also known as query restriction), and adding noise to the data (also called data perturbation) [24, 26]. Under query restriction three techniques have been utilized, data partitioning, cell suppression, and query size control (also called blocking) [23]. This last technique ensures that the value of a cell returned as a result of a query is generally above a threshold value. This approach is used in Datamart, where aggregate values less than five are replaced by the character "#". The advantage of this approach is that it is simple to implement and ensures privacy preserving as long as the threshold value is appropriate. Its drawback is that it penalizes the utility of the data especially for cases where actual values (i.e. instead of the "#" character) are necessary in order to make use of the query results.

Under the data perturbation approach, noise is added either to the data or to the results of the queries. Recently k-anonymity was proposed to assess the disclosure risk of confidential information. K-anonymity ensures that the identity of an individual cannot be reversely identified within a set of k individuals. Algorithms have been proposed to achieve k-anonymity mostly using suppression and generalization [21]. Loss of information is a trade-off of this approach as attributes are either abstracted to higher concepts (e.g. age value is generalized to range values) or suppressed. A great effort in the proposed algorithms was towards improving the efficiency of the k-anonymization process as it is known that achieving an optimal k-anonymity solution is NP-hard [3, 17].

Few contributions have focused on the utility of the information when it is transformed to satisfy k-anonymity. Work such as in [15, 21, 28] characterizes information loss in terms of the number of entities (individuals that falls within each group that satisfies k-anonymity (minimum is k) or in terms of the size of the generalized domain of the attributes. Xu et al. [28] takes into account the importance of the attributes in their specification of the information loss, providing the ability to give a weight for each attribute that needs to be generalized. Samarati et al. [19], use generalization heights to represent the information loss of a generalized dataset; but this approach does not take into account that a generalization height in one attribute may be not as costly as in another attribute. Another utility metric, discernability [4], assigns a cost to each tuple based upon how many other tuples are identical to that tuple. Although this is an interesting approach, however it does not take into consideration data distribution. As stated in [13], an anonymized dataset where the original distribution attributes are uniformly distributed represent less information loss than an anonymized group where the original attributes were skewed.

While the existing approaches allow for automatic characterization of information loss, they do not account for the non-linearity of the change in the value of the data as it becomes more generalized. The value of data to a researcher is often not proportional to the number of specific values or combinations of values in a dataset. For the researcher, it is much more important to provide an anonymized dataset that provides de-identified content while still maintaining the content or meaning of the original data. For instance, in health care research, age generalizations that preserve general inflection points in health care status, such as the late teens, 65 years old, and 80 years old, may be more valuable than generalizations that obscure those boundaries but include more age groupings. Losing an age group boundary at 80 years old may only decrease the data's utility slightly, while losing the 65 year old boundary may produce a significant change in the data utility. One approach to assess the utility of the data after the anonymization process is to determine the amount of informative patterns that can be discovered using data mining techniques in comparison to the patterns that are discoverable in the raw dataset. When anonymizing a dataset, the input of the data content expert can provide insight into the needs of the end user (such as maintaining important age boundaries), so that information may be maintained as much as possible in the anonymized dataset.

In this paper we propose a fully user-driven utility metric to guide the process of k-anonymization; and we describe two utility-based privacy preserving approaches that implement the new data utility metric while still ensuring k-anonymity. As described in [11], a utility-based privacy preserving algorithm has two goals: 1) protecting private information and 2) reducing information loss due to generalization. Our new utility metric considers information loss from the perspective of the end user, who often desires to assess patterns that may not be preserved in a sanitized dataset that conforms to a distribution-based utility metric. The experiments we have conducted using real data show that our approach scales well to datasets that contain large numbers of attributes and multiple generalization levels within those attributes, while incorporating the view of the data from an end user perspective as the attributes undergo generalization. More specifically, the contributions of this paper are the user-driven utility metric and the two proposed algorithms which are designed to approach the aspect of utility-based anonymization from a holistic view (global optimization) and an intra-attribute view (local optimization).

The paper is organized as follows. Section 2 provides some definitions to support the new approach for characterizing the utility metric for k-anonymization. Related work is presented in Section 3, and the new utility metric is described in Section 4. Section 5 describes the methodology, which includes the dataset and the anonymization algorithms that implement the new data utility metric. The

results of the experiments are presented in Section 6. Section 7provides a discussion of the results, and a summary of the paper and future research directions follow in Section 8.

2 Definitions

The basic definitions provided here are also presented in [14, 15] as we find that their description of attributes generalization is very concise and applies to our work.

2.1 Attribute Identifiers

Let T= (t_1,t_2,…t_m) be a table storing information about individuals, described with a set of attributes A = (A_1, A_2, . . . ,A_n). We distinguish three types of attributes in A, labelled as explicit identifiers, quasi-identifiers and sensitive identifiers as defined in [16].

An attribute A_i is labelled as explicit identifier if it can be used to uniquely identify an individual. Examples include social security number and name. To preserve the privacy of the published data we assume that the explicit identifier attributes undertake a transformation process such as randomization [8]. Quasi-identifiers are defined in the next section, and sensitive identifiers are attributes that contain data that are considered to be extremely personal, such as disease state or a salary.

2.2 Quasi-Identifier Attribute

A set of attributes (A_1, A_2, . . . ,A_n) of a table T is called a quasi-identifier set if these attributes can be linked with external data to uniquely identify at least one individual in the general population Ω [16]. It is assumed that the quasi-identifier attributes are known based upon the specific knowledge of the domain experts.

In the work described in [16], a sub-class of quasi-identifier attributes are defined and labeled as sensitive attributes. An example of a sensitive attribute is *cause of death* such as individual X died of cancer. In our work this distinction is not made, which will be addressed in the algorithm discussion.

2.3 Frequency Set

Let Q= (A_1, A_2, …, A_q) be a subset of A. The frequency set of T with respect to Q is a mapping from each unique combination of values (v_0, … v_q) of Q in T (the value groups) to the total number of tuples in T with these values of Q (the

counts) [13]. In other words, the frequency set of T with respect to Q stores the set of counts of each unique combination of values of Q in T.

2.4 K-Anonymity Property

Relation T is said to satisfy the k-anonymity property (or to be k-anonymous) with respect to attribute set A if every count in the frequency set of T with respect to A is greater than or equal to k [22]. Similar to [13], in order to determine the frequency set from table T with respect to a set of attributes A, we are utilizing the COUNT(*) functionality of SQL with A as the attribute list in the GROUP BY clause of the query. In addition to the value returned by COUNT(*), we are using the MIN(list) function to allow of all the calculations for the frequency to be performed at the SQL database level. For example, a sample query of the patient database may look like this expression:

select min(myCount) as count from (select count(*) as myCount from DB1 group by q_1, q_2)

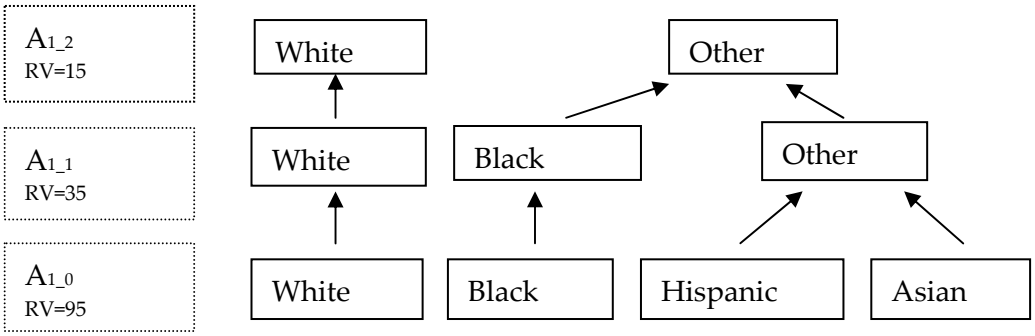
The result from this query is compared against the k-anonymity threshold value "k" for the combinations of attributes q_1 and q_2.

2.5 Attribute Generalization and Suppression

The basic idea of generalization is to abstract the domain of attributes to make it more difficult to distinguish individual values and therefore increasing the chances of achieving k-anonymity. Examples of generalization include generalizing zip code values by replacing the last digit with wild card (i.e. *) or generalizing individual age values into a range of values. Suppression of attributes is simply regarded as the case where the attribute is generalized to the highest or most general level (e.g. zip code attribute is generalized to *****). Please note that we will refer to the highest/most general level of an attribute as the root level in the attribute generalization tree later in the paper. As an attribute approaches the root level in the generalization tree, the information loss for that particular attribute increases. Minimizing the level of an attribute's generalization during the anonymization process will minimize the amount of information loss. Therefore there is a need for the existence of different levels of attribute domain generalization to be available for the transformation process so that the trade-off between information loss and anonymization can be requested. Let D represents the set of attributes domains including both categorical and numerical domains; and let $<_{DG}$ denotes the domain generalization relationship between domains; where the notation "$D_{l\_i} \leq D_{l\_j}$" between two domains D_{l_i} and D_{l_j} defined on attribute A_l, means that either D_{l_i} is identical to D_{l_j}, or D_{l_j} is a generalization of D_{l_i}. The mapping between values from D_{l_i} and D_{l_j} can be rep-

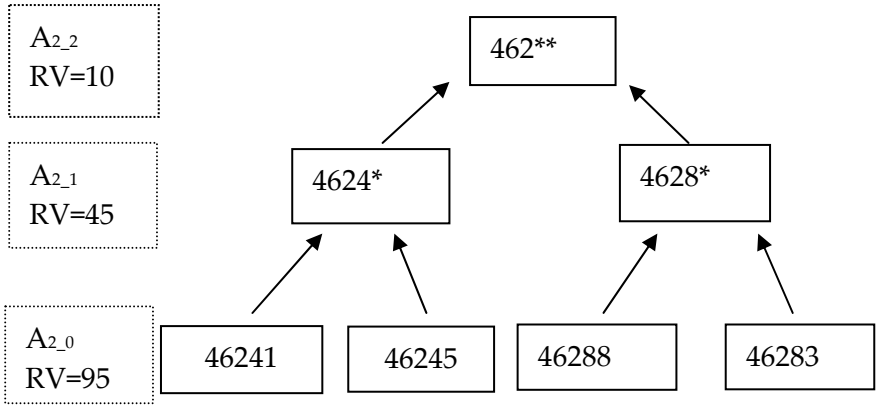
resented by a many-to-one generalization function denoted by γ. By Convention i<j; and D_{I0} represents the most specific domain (also noted D_I) for attribute A_I.

Figure 1. Generalization of race attribute



For each attribute we can define a hierarchy of domain nodes totally ordered using $<_{DG}$; where the root of the hierarchy represents the most generalized domain, and the leaf nodes represent the most specific domain (i.e. original domain of the attribute). Figure 1 and Figure 2 provide examples of the domain generalization hierarchies for the race and zip code attributes.

Figure 2. Generalization of the zip code attribute



Direct edges between two nodes are the results of direct generalization produced by applying the generalization function γ; and paths between nodes are implied generalization between domains produced by a series of composition of the generalization function, denoted by $\gamma+$. Each generalization level for an attribute is labeled with the attribute number x and the generalization level y (A_{x_y}). The most specific data for an attribute is labeled with a zero, A_{x_0}, and as the attribute becomes more generalized the value increases by one. In Figure 1, the most generalized level is labeled as A_{1_2}.

2.6 Full Domain Generalization

As described in [5, 24], several models exist to transform table T to the k-anonymized view V, including the global recoding. In global recoding the initial values of each quasi-identifier attribute are mapped to new values to satisfy k-anonymization. Several approaches exist for global recoding; see [4, 5] for more information. Using full-domain generalization approach, initial values of each quasi-identifier attribute are mapped to values in the same domain in the attribute domain hierarchy. More formally, let T be a relation with quasi-identifier attributes $A_1,\ldots A_n$. A full-domain generalization is denoted by a set of functions, $\Phi_1,\ldots\Phi n$, each of the form $\Phi i : D_{Ai} \rightarrow D_{Qi}$, where $D_{Ai} <_{DG} D_{Qi}$. Φi maps each value "q" from D_{Ai} to some value "a" in D_{Qi} such that a = q or a belongs to $\gamma+(q)$. A full-domain generalization V of T is obtained by replacing the value q of attribute Ai in each tuple of T with the value $\Phi i(q)$ [19, 20]. This is in contrast with local recoding [13, 14, 27], where initial values of an attribute Ai can be mapped to values in different domains in the attribute domain hierarchy. For example, the age attribute value of 15 exists in the 15-20 domain as well as in the 14-18 domain. The general idea of local recoding is to minimize the interval size, which may achieve less information loss due to smaller intervals than global recoding. We address the issues of the same values existing in different domains in Section 7.2.

This property allows generalizing attribute domains into higher domains. The hierarchies of an attribute domain generalization can be constructed by progressively mapping the attribute domain values into a higher attribute domain.

The *race* attribute is shown in Figure 1, and the zip code attribute is shown in Figure 2**.** It is initially defined at the most specific level of the attribute to be (White, Black, Hispanic, Asian). It was then generalized to the level (White, Black, Other) and then into an even more generalized level (White, Other). The generalization groupings of the race attribute example demonstrate a concept that is critical to our new data content expert based utility measurement, which is that particular values in an attribute are maintained as much as possible even if they exist in multiple generalization levels. For example, the values of White and Black exist in three generalization levels, and the reasoning for this is that those two values in the race attribute have been designated by the data content expert as critical for research purposes. If a generalization level is required to drop one of these critical values, the data content expert considers the information loss to be significant. With that in mind, the assigned utility metric for a generalization level without the data content experts desired values intact will reflect that loss. More details on the calculation of the utility metric are defined in Section 4.

3 Related Work

The protection of microdata has been an active research issue [25], and many researchers have been utilizing k-anonymity to protect the identities of the individuals in a database. K-anonymity and the deployment of generalization/suppression to satisfy k-anonymity were originally characterized in [20], and a binary search algorithm to find a single full-domain generalization was described in [19]. New optimization methods were developed in [1, 3, 4, 13, 14, 17]. For example in [13], the authors introduce a class of algorithms for a multidimensional data model that produce k-anonymous full-domain generalizations while still maintaining a substantial performance improvement over existing full-domain algorithms.

An area of research within privacy protection has been the analysis of k-anonymity, and whether or not it protects the privacy of data. *L*-diversity, proposed by [16], suggests that k-anonymity is susceptible to homogeneity of the data combined with the knowledge of the attacker. An examples is that if the attacker knows that a dataset includes all persons in a county, and the data shows that all persons in the datasets with syphilis also have AIDS, and the attacker has external knowledge that his acquaintance Jim, a county resident, has syphilis, the data has revealed to the attacker than Jim has AIDS. As initially described, *l*-diversity proposes to protect not only the quasi-identifiers, but provides special attention to a subset of the attributes called *sensitive* attributes (e.g. an attribute storing HIV status for a patient, which the patient would not want disclosed) characterized by having at least l well-represented values exist in a set of records that have the same values for the quasi-identifiers. In contrast to *l*-diversity, [15] proposes a concept called t-closeness. This concept is based on the premise that for any equivalence class, which is a set of quasi-identifiers, the distance between the distribution of a sensitive attribute in the equivalence class and the distribution of the attribute in the whole table is no more than a threshold of t.

We consider all attributes to be sensitive attributes, so during the anonymization process, our algorithms ensure that we have k records to prevent identity disclosure. Even though our anonymized dataset may not satisfy t-closeness, we ensure that we have at least k records for each of the "sensitive" attributes as classified in [16]. An attacker may have external information about any subset of attributes within a dataset, and so any attribute must be treated as a possible contributor to identity disclosure, as illustrated by the two attributes Resident County and syphilis status in the prior example. To achieve k-anonymity, there must be at least k records with any combinations values from all of the attributes in the anonymized dataset, not just values from some subset of attributes that have been categorized as identifiers or quasi-identifiers.

Among studies of the utility of the data after anonymization, the research of Xu et al. [28] is one of the few studies that emphasized the need to build utility aware anonymization in terms of weights among individual attributes. The example they provide highlights the difference in importance that exists between age and zip code attributes when conducting a study for disease analysis. More precisely, age has more importance in this type of study. Therefore, it makes sense to try to minimize the level of generalization of age when compared to zip code during the transformation process. It should be noted that the weights are not intended to create different anonymized datasets for each type of user, but instead provide weight on attributes that are generally more important for the majority of users of the data. For this aim, an attribute weight has been introduced in the utility metric they proposed; although it has been set to one all across the attributes when actually implemented. The utility metric obtained corresponds to the sum of the weighted utility of each attribute. The utility metric, also called *normalized certainty penalty*, is expressed in terms of the loss of information generated by the generalization process. In a case of a numeric quasi-identifier attribute Ai whose initial domain Di is generalized to domains D_{i_0}, D_{i_1}, ... D_{i_m}, the loss of information is expressed in terms of the sum of the ranges of each sub domain D_{i_j} normalized by the range of the initial domain. D_i. Similar reasoning can be extended for categorical attributes.

In other works of k-anonymization such as in [4, 11, 14] the authors have also introduced utility matrices to guide the transformation process; but did not take into account the importance of the attributes. For example in [4], the *discernability* model is introduced to measure the information loss for each attribute Ai, by assigning a penalty to each tuple in the table based on the number of tuples having the same generalized sub-domain D_{i_j}. In the work described in [14], the *normalized average equivalence class size* is introduced. For each quasi-identifier attribute Ai, the information loss is expressed in terms of the number of tuples in the table divided by the number of group-bys for the attribute Ai generated in the next generalization level.

In the work described in [12], the authors propose to release frequency related information about the data called marginals. For example, if there are five people in a zip code that are forty years of age, the authors will release a table with an entry of forty with a count of four. The determination of what marginals are released is dependent on an entropy measure and not based on the needs of the researcher who wants to mine that data. The concept of a normalized certainty penalty (NCP) is introduced in [10] to capture information loss as the data is generalized into intervals, therefore losing accuracy in query answering. For example, a user may want to know how many 18 year old men purchased beer and diapers in 2005, but may only be able to count men ages 16 to 20 years old if

age has been aggregated into five year domains in the dataset. For all numerical attributes, A_i, from table T, NCP is defined as

$$NCP(t) = \sum_{i=1}^{m} w_i * \frac{z_i - y_i}{|A_i|}$$

where $|A_i|$ is defined to be the $max_{t \in T}(t.A_i) - min_{t \in T}(t.A_i)$, i.e. the range of all tuples on attribute A_i. The numerator contains that variables y_i, and z_i, which are the generalized values for x_i. Finally, w_i reflects the weight of the utility for attribute A_i as compared to the other attributes in the dataset.

Categorical attributes follow the following formula for the NCP:

$$NCP(t) = \frac{size(u)}{|A|}$$

where the size(u) is the number of common descendants and |A| is the number of distinct values for attribute A. The NCP is an interesting concept, but its limitation is that it does not take into account the importance of particular values in the dataset that have been identified by the data content expert as critical to the analysis of the researcher. In the next section, we present our utility metric that builds upon aspects of the NCP, but also penalizes a particular generalization level of an attribute if the critical values of that attribute have been generalized. The data content expert, who is very familiar with the needs of the researchers receiving the data, pre-defines these critical data values in rules that may be a value like "white, or black" for a categorical attribute like race, or a range of numbers like 12-18 for age is well accepted as "adolescent." If these values are not present in a particular generalization level, then the information loss increases and the penalty also increases.

4 Utility Metric

We propose that the data curators have more control on defining the utility of the attributes and how they relate to the overall content of the data that is contained in the datasets that they own. The expertise of the data researchers provides an understanding particular to thresholds in the data that need to be maintained through the anonymization process, so that the meaning of the data is maintained. For this purpose, we define the research value (RV) of an attribute to encapsulate the utility of the each attribute with respect to the following conditions:

1) Significance of the attribute relative to the other attributes;
2) The distinct number of elements in a group at a particular generalization level;
3) The number of records that exist for each group in a generalization level;
4) The number of data constraint rules that are maintained in that generalization level.

A data constraint rule defines groupings of data or ranges of data that are critical for an attribute to maintain during the anonymization process in order to maximize the meaning of the data for the end researcher. The data constraint rules can exist in two forms depending on the data type of the attribute: categorical or continuous. Categorical attributes use data constraint rules that preserve distinctions between data values or between data value domains. When all of the data constraint rules are satisfied at a particular generalization level of an attribute, the value of the data constraint rule in the RV is the sum of all possible importance values divided by the sum of all the importance values in the raw dataset, which would be one. Any generalization that violates such a constraint rule would be penalized by multiplying the generalization string's total research value with a value less than one. Using the race attribute as an example, the data expert may assign the following importance values to the elements that exist for race in a database:

Table 1. Possible Data Constraint Rules for Race Attribute

Constraint Rule	Importance value
Do not mix White and Hispanic in same group	5
Do not mix White and Black in the same group	20
Do not mix Hispanic and Black in the same group	10
Do not mix Hispanic and Asian in the same group	5

If a generalization level violated the mixing of Hispanic and Black individuals, then the data constraint portion of the RV would be 30/40 = 0.75.

For a continuous attribute like age as discussed previously, the data constraint rules could define inflection points of ages that would be important for someone who is interested in mining data within the well-defined age groupings. For example, preserving a distinction between age 20 and 21 years may be important to a researcher examining alcohol use, since drinking alcohol usually becomes legal on a person's 21st birthday. Similar to the continuous data constraint

rules, the importance values assigned for each of the ranges of values for an attribute are normalized to one. The research value (RV_k) of a numerical attribute x at generalization level k is defined to be:

$$RV_k = w_x * \frac{\sum_{i=0}^{n} R_i^0 * N_{R_i}^0}{\sum_{i=0}^{k} R_i^k * N_{R_i}^k} * \frac{\sum DR_k}{\sum DR_0}$$

where w_x is defined to be the importance weight of attribute x in reference to the other attributes in the dataset. The numerator $\sum_{i=0}^{n} R_i^0 * N_{R_i}^0$ is the sum of the number of elements within each sub-group i times the range of values for those elements in the i^{th} group at the most specific generalization level of attribute x. $\sum_{i=0}^{k} R_i^k * N_{R_i}^k$ is the sum of the number of elements within each sub-group i times the range of values for those elements in the i^{th} group at the k^{th} generalization level of attribute x. Finally, the data constraint rules portion of the equation is the ratio of the sum of the data constraint rules that exist at the k^{th} generalization level, $\sum DR_k$, divided by the total value of all the data constraint rules at the most specific generalization level, $\sum DR_0$. The weight calculation of each attribute can be determined by establishing a correlation matrix between each of pair of attributes using the original raw dataset. The total sum of all the weights is normalized to be 1. If an attribute does not correlate highly with any other attribute, then that attribute is considered to be an independent attribute and will be assigned a higher weight. On the other hand, if an attribute is highly correlated with the other attributes, then it will be assigned a lower weight. As the number of attributes increases, the complexity of determining the correlations between attributes increases dramatically. For the experiment we describe in this paper, the data expert manually assigned weights for each of the attributes in both the MCPHD and Adult datasets, but the proposed utility metric to calculate the RV values was used at each generalization level of the two datasets.

For categorical attributes in a dataset, the research value (RV_k) of attribute x at generalization level k is defined to be:

$$RV_k = w_x * \frac{|A_k|}{|A_0|} * \frac{\sum DR_k}{\sum DR_0}$$

where all the elements are defined to be the same as those for the numerical attributes, except for the $\frac{|A_k|}{|A_0|}$ ratio which is defined the number of unique elements at generalization level *k* divided by the number of unique elements defined at the most specific generalization level of the attribute.

To demonstrate the calculation of the research value for the kth level of attribute x containing numerical data, the following example is presented. The weight of the attribute x is calculated to be 0.2; there are three groups in this generalization level with 25, 45 and 55 elements in each group spanning 10, 15 and 25 values, respectively. The most specific generalization level of this attribute has 125 elements with a group spanning of 1. The sum of the data constraint rules that exist at the kth level is 50, and the sum of the data constraint rules at the most specific generalization level is 100. Given all of these values, the research value for the kth level of attribute x is determined as:

$$0.2 * \frac{125 * 1}{(25 * 10) + (45 * 15) + (55 * 25)} * \frac{50}{100} = 0.0054$$

At the most specific level, the RV of an attribute is equal to w_x. It becomes apparent that as one moves to more generalized levels within an attribute, the denominator will continue to grow, and thus the RV value will continue to decrease.

Figure 1 demonstrates the research values for the race attribute race across different levels of generalization. As you can see from this example, the starting research values are not the same for all of the attributes, which indicates that the data content expert has designated the race attribute to have a higher significance value than that of the zip code attribute. This distinction in starting research values allows the data content expert the ability to establish not only research values within an attribute but also to reflect an attribute's utility across attributes. As the number of elements increases within a sub-group at a particular generalization level for an attribute, the chances are greater that those set of values will produce a measurable pattern during data analysis. In contrast, if the range of values within a sub-group is very large, then the chances of producing a measurable pattern in the dataset decrease.

It is important to note that given two attributes A_m and A_n such that $D_{m\_0} \leq D_{n\_0}$, then the initial importance status is not necessarily maintained as attributes A_m and A_n are generalized. That is, $D_{m\_i} \leq D_{n\_j}$ where i<j does not always hold in the general case. This is a result of the data constraint rules defined for a particular attribute, and how the generalization levels are defined for those attributes. Informally, the partial order between research values allows for flexibility

in defining domain hierarchies for each attribute and the ability to re-evaluate the utility of the attribute and its importance with respect to the other attributes as the attributes undergoes global recoding. Compared to the information loss defined in [7], the research value metric can be regarded as an opposite metric, wherein the more the attribute undergoes transformation, the less research value it will have.

To optimize the final overall utility of the transformed data using the research value metric, two alternative algorithms are proposed:

- Optimization of the overall research value of the dataset after generalization by maximizing the overall sum of the research values of the transformed attributes. We call this option global research value optimization
- Optimization of individual attribute research value, by maximizing the individual research value of each transformed attribute. We call this alternative local research value optimization

It should be noted that the research values used by these proposed algorithms were manually calculated by the content expert, and that our future work will be to automatically generate the research values of the attributes.

5 Methodology

In this section, we will describe the two algorithms that address the *global research value optimization* and the *local research value optimization* and the datasets that were used for running experiments on the algorithms.

5.1 DataSets

For this project, we utilized two datasets, the public Adult Census data from the UC Irvine machine learning repository [18], and the proprietary death certificate dataset from the Marion County Public Health Department (MCPHD) of Indianapolis, Indiana. We included the Adult Census dataset in order to compare our proposed algorithms against existing methodologies, since the MCPHD is not available to for public download, and the Adult Census dataset is the gold standard for gauging anonymity techniques.

The Adult dataset was configured in a similar manner to [27] using 30,162 tuples and eight attributes (age, work class, education number, marital status, occupation, race, sex and native country). Age and education number were used as numeric values, while the remaining attributes were used as categorical attributes. Work class and marital status used a three level hierarchy structure. For the other categorical attributes, a two level hierarchy was used with the

most specific level having all values, and the second level was set to "ALL," (i.e. complete suppression).

For the Marion County Health Department's death certificate dataset (a total of 216,000 records) comprised of 76 attributes was paired down to 36 attributes based upon their utility for data mining. These attributes include: race, sex, college education, cause of death, etc are listed in Table 2. For each attribute, we created n levels of generalization, where n ranged from one to six. The original version of the MCPHD dataset contained a wide range of values in each attribute. This variety produced many outlier values that needed to be reclassified for categorical attributes, or removed in the case of numerical data after performing a distribution analysis of each attribute to identify outliers.

Table 2. Marion County Public Health DB Attributes

Marion County Public Health Department Database Attributes
Race, Sex, Age in Days, College, Industry, Autopsy, Census Tract, Cause of Death Certifier Type, Citizenship, City of Birth, Date of Birth, Date of Death, Disposition Method, Education, Farm, Informant Relationship, Injury AM/PM, Injury Census, Injury County Code, Injury Date, Manner of Death, Marital Status, Military Motor Vehicle Accident, Occupation Category, Occupation Code, Place of Death City, Place of Death Code, Place of Death State, Place of Death Zip, Pregnant, State of Birth, US Vet, Zipcode, Injury Time, Time of Death

These identified outliers were then recoded to a general category within the attribute. As described above, every level of generalization for each attribute was assigned a research value (RV), which ranged from zero to one hundred (i.e. normalized values). Table 3 demonstrates another generalization example of the two attributes (Race and Gender), and the corresponding research values for each level of generalization within the attribute.

Table 3: Research Value Examples

Attribute	RV	Generalization Levels
Race	0.95	W, B, L, A, O*
Race	0.72	W, B, L, O
Race	0.58	W, B, O
Race	0.10	W, O
Gender	0.85	M, F, O$
Gender	0.65	M, U or F, U

(*W= White, B=Black, L=Latino, A=Asian, O=Other, M=Male, $F = Female, U= Unknown)

5.2 Data Preparation

Using the MCPHD and the Adult census raw datasets, a perturbed dataset was created by running a SAS® script that formatted the data into generalization columns. For each attribute, x columns were created based upon the number of levels of generalization each attribute contains, as discussed in the previous section. In Figure 1, we can see how the race attribute, which has three generalization columns, will be created (A_{1_0}, A_{1_1}, and A_{1_2}). In the A_{1_0} column, all of the records will contain either "White," "Black", "Hispanic", "Asian." In the A_{1_1} column all records containing either "Asian" is abstracted into "Other"; and in A_{1_2} column all records containing either "Black", "Hispanic" or "Other (from the previous level) are abstracted into "Other." The last level (not shown in Figure 1 and 2) is the most general level with a zero research value, where all records contain the same value, "any", for the generalized attribute. Throughout the hierarchy creation process, no attribute values were allowed to be in two groups at once. For example, a generalization of the age attribute could not have overlapping groupings, like age 13-17 and age 15-22. When values are allowed to cross groupings, it makes it very difficult to discriminate what grouping of a particular value (i.e. 15 in this example) is responsible for a pattern in the dataset. In effect, the groupings for age would range from 13-22, because you could not assert if 15 was in the 13-17 grouping, or the 15-22 grouping; thus the utility of the dataset has been decreased. Publications using local recoding where values are allowed to cross groupings during the anonymization process include [2, 3, 9, 28]. This process was repeated for all thirty attributes in the MCPHD dataset and then for the Adult census dataset.

After all thirty attributes of the MCPHD had been generalized; multiple combinations of those thirty attributes were created to test the effectiveness of the

two proposed algorithms. A combination contained as little as three attributes and up to the maximum of thirty attributes. The criteria for the selection of the attributes that were selected for each combination fell into two categories: 1) Random or 2) Maximum number of generalization levels. The maximum number of generalization levels approach would examine two attributes A_1 and A_2 and select A_1 if it had more generalization levels than A_2, or vice versa. In the case where A_1 and A_2 had the same number of generalization levels, then one would be selected randomly. The random combinations were labeled as ri (e.g. r03 and r08), which indicates a random selection of i attributes from the original pool of thirty attributes. The maximum combinations were labeled in a similar manner mi (e.g. m03 and m08). For the Adult dataset, all of the attributes were used in during the testing phase of the algorithms.

5.3 Algorithms

To ensure that a dataset is k-anonymous, it is critical to test the worst case scenario for the data, which in this case is a combination of all possible attributes being searched in a single query. This is due to the fact that as the number of attributes that are combined in a query increases, the chances of k-anonymity being violated also increases. Herein after we represent this combination of attributes as a string called generalization string $A_{1\_1}A_{2\_2}\ldots A_{m\_n}$ composed of the combination of the individual attribute generalization level strings A_{i_j}. For example, the race and zip code attributes would create generalization strings like $A_{1\_0}A_{2\_0}$, $A_{1\_0}A_{2\_1}$, $A_{1\_1}A_{2\_0}$, etc.

The aim is to efficiently compute a dataset generalization string that optimizes (globally or locally) the overall research value of the transformed view V. Let us call this string *globally (resp. locally) optimized generalized string*.

One problem with this strategy is that a dataset with large numbers of attributes will create millions of possible combinations of generalization strings. From efficiency perspective the bottleneck point for either one of the alternatives is the computation of the frequency set for any dataset generalization string, as it involves a database call to compute a select-group-by SQL statement.

Given a set of attributes A = $\{A_1, A_2, \ldots, A_n\}$ and a set D of attribute domain generalization hierarchies, the initial number of dataset string generalizations is function of the number "n" and the number of levels of each attribute domain hierarchy in D. Therefore to reduce the number of initial dataset generalization strings, we need to reduce either the number of initial attributes and/or the hierarchy depth of the attributes. The pre-pruning phase addresses both options.

5.3.1 Pre-Pruning

The strategy employed in the pre-pruning phase is supported by the following properties also used in [14]. The first property called the generalization property states that if two sets of attributes P and Q have their domains satisfying $D_P \leq D_Q$, and if T is k-anonymous with respect to P; then T is k-anonymous with respect to Q.

The second property called *subset* property states that if T is k-anonymous with respect to a set of attribute P, then T is also anonymous with respect to any subset of P.

Using the negation of the subset property we can infer that if T is not k-anonymous with respect to an attribute in A, then it is not k-anonymous with respect to any superset obtained by combining the attribute with the other attributes of A. Using the negation of the generalization property we can infer that if T is not k-anonymous with respect to A_{i_j} then it is not k-anonymous with respect to A_{i_l} such that l<j. The outline of the pruning strategy is depicted in Figure 3.

Figure 3. Pre-Pruning Algorithm

Input: Table *T* containing attribute names, research values (RV) of attributes, generalization levels of each attributes.
Output: List *L* containing generalization levels passing *k*-anonymity
Method:

```
1.  While an attribute and generalization level (gl_i) >= k exists {
2.      For each generalization level gl_i of attribute a_m test k-anonymity {
3.          If gl_i passes k-anonymity {
4.              Add gl_i and RV_i to List L
5.          }
6.          Else {
7.              Select attribute a_{m+1}
8.          }
9.      }
10. }
```

The pre-pruning strategy uses the negation of both properties, by checking for each attribute whether or not it satisfies k-anonymity (Line 3). If it does not, then it can be pruned from the composition of the initial generalization string set. To account for the existence of different domains for each attribute, we refine the pruning process to prune for each attribute any generalization domain that does not meet k-anonymization. For example, in Figure 1, if attribute A_{10} does not meet the k-anonymization threshold but A_{11} and A_{12} do, then the domain hierarchy of attribute A1 will be trimmed to include only the top two levels; and therefore only A_{11} and A_{12} will be used in generating the set of initial dataset generalization strings. If an attribute A1 fails k-anonymity at the most

generalized level, then that attribute is removed from the dataset, because it would cause other attributes that were combined with A1 to also fail k-anonymity. The benefit of this pre-pruning process is a more efficient k-anonymity algorithm by minimizing the number of calls to the database to test the generalization strings for k-anonymity.

5.3.2 Global Optimization of the Utility Metric

The aim is to compute the dataset generalization string that meets the k-anonymity threshold and have the best global research value. The global research value is computed by summing all of the research values from each respective attribute in the successful generalization string. This method requires that the research values of all combinations of the attributes' generalizations be calculated. This may produce a very large number of generalization strings, as is the case for the MCPHD dataset, but the pre-pruning eliminates a large portion of the strings that do not satisfy k-anonymity. To minimize the number of database calls, we deploy a binary search over the list of all dataset generalization strings sorted in ascending order on their global research value. At each step of the binary search, we apply several strategies to minimize the number of generalization strings that need the computation of the frequency set. The pruning steps depend on whether the selected generalization string fails the k-anonymity test. The details of the global optimization algorithm are shown in Figure 4.

For the case of a success, the generalization string is added to the list of successful strings SL and the pruning strategies are applied. The first pruning strategy eliminates all generalization strings with a global research value less than the successful string. The second pruning strategy eliminates all generalization strings that are more general that the successful strings. A generalization string is considered to be more general if all of the component attributes of that string have a generalization level $gl_k > gl_i$ where *i* is the generalization level of the current string, and k is the generalization level of the string that could be removed from the list.

For the case when the current generalization string fails, then the only pruning strategy that applies is to remove all generalizations strings that are more specific than the current string.

Figure 4 Global Optimization Algorithm

Input: List *L*
Output: A *k*-anonymous *T'*
A list *S* of successful generalization strings and their corresponding research values
Number of root nodes within any successful generalization string
Method:

```
1.  Init: Create all possible strings (GSi) from L, sort by total RV. Store in List A
2.  While there exists a generalization string GSi in List A {
3.      Select GSi as midpoint(List A)
4.      If min(count(GSi)) >= k {
5.          Add GSi to success list SL,
6.          Remove all GSk from List A with RVk < RVi
7.          Remove all GSk from List A where glk > gli
8.      }
9.      Else {
10.         Remove all GSk from List A where glk < gli
11.     }
12.     Remove GSi from List A
13. } // End while
14. For each GSi in SL {
15.     Determine root # of attributes where glx = glMAX
16. }
17. Apply GSi from SL with (>> RVk && min(root)) on table T
```

The binary search process is repeated for the remaining list of non- pruned generalization strings until no more strings are left to be analyzed. If multiple successful generalization strings were found after running the algorithm, all having similar research values, then it would be at the discretion of data content expert to select a generalization string that would be most beneficial from the end-user perspective. The number of root nodes (most specific levels of an attribute) is determined to provide the data content expert the ability to choose from multiple success strings after the anonymization process is complete.

5.3.3 Local Optimization of the Utility Metric

The objective of the local optimization approach is to achieve K-anonymity with optimum RV values for each attribute (i.e. local). As opposed to the global optimization approach where the focus is to find the best RV combined over all attributes, the aim of this approach is to balance the global RV value between the attributes. In other words, finding generalization strings that minimize the cases where generalization strings include very specific attributes at the expense of most general attributes. For example, using attributes race and zip code in Figures 1, the best generalization string using the global strategy would generate the generalization string $A_{1\_2}A_{2\_0}$ (combined RV=0.90) while the local strategy may generate the generalization string $A_{1\_1}A_{2\_1}$ (combined RV=0.50).

Unlike the global approach, the local approach does not use a combined list of all possible strings to select a generalization string for k-anonymity testing. Instead, each attribute is regarded separately (As shown in Figure 5); and at each step, a generalization level within each attribute is selected and combined with the other selected generalization levels of the other attributes in order to create a combined generalization string to be tested for K-anonymity. If that particular generalization string succeeds, then the next selected generalization level in each attribute moves half way up the height of the attribute towards the more specific data of an attribute (i.e. the data is not grouped or suppressed). On the other hand, if the generalization string fails, then the next selected generalization level selected in each attribute moves half way up the height of that attribute towards the more general data. This continues until it is not possible to move in all of the attributes that compose the generalization string. If the current generalization string passes k-anonymity, then the current string is added to the success list SL along with its total research value. No pruning occurs in the local optimization algorithm, but a hybrid version of the local optimization as described in Section 5.3.4 does use pruning.

To facilitate a binary search in each of the attributes of the generalization string, we utilize pointers to maintain the current selection level of the attribute, and also the highest and lowest points still available for selection.

Figure 5 Local Optimization Algorithm

Input: List of n attributes
Output: A k-anonymous T''
A list S of successful generalization strings with research values
Number of root nodes within any successful generalization string
Method:
1. // Initialize the following index pointers: Hi, Lo & Mid
2. While (Total Stops <> # of attributes) do{
3. Select a generalization string GS_i using Mid index j of all attributes A_1 to A_n
4. If min(count(GS_i)) >= k {
5. Add GS_i to success list SL
6. Set Lo = Mid
7. } Else {
8. Set Hi = Mid
9. }
10. Mid = (Hi + Lo)/2
11. } // End while
12. For each GS_i in SL {
13. Determine root # of attributes where $gl_x = gl_{MAX}$
14. }
15. Apply GS_i from SL with (>> RV_k && min(root)) on table T

This procedure continues until the current selection level does not change during an iteration, which is classified as a stopping condition for that attribute. When all of the attributes have met their "stopping condition," the algorithm terminates. At this point, similar to the global approach, all of the success strings are examined for any roots. The aim is to eliminate success generalization strings with attributes at the most general level. The generalization string with the greatest research value and fewest number of root attributes would then be applied against the raw database to ensure anonymity while still maintaining some of the utility of the data.

5.3.4 Hybrid Utility Algorithm

The hybrid approach is a combination of the local approach and the global approach that takes advantage of the quick examination of strings via the local algorithm and then uses the wider scope of the global algorithm to identify any remaining success strings. Unlike the local optimization approach, the hybrid optimization makes use of the list of all possible generalization strings for a dataset, and pruning of those strings as the algorithm executes. It starts of using the local algorithm until all of the high and low pointers for all of the attributes are equal. Once this point is reached, if there are any entries left in the remaining list of generalization strings, the global algorithm is then called until no entries exist in that list.

5.3.5 Distributed Version of the Global Optimization

As described in Section 5.3.2, the global approach assumes that all possible generalization strings are generated a priori and provided as an input (residing in main memory) for the algorithm. This assumption generates implementation issues as soon as we have a number of attribute combinations greater than twelve. To address this issue we propose a distributed version of the algorithm that leverages the subset property described in Section 5.3.1. The main idea of the distributed version of the algorithm is to decompose the generalized string into subsets of generalization strings that can fit in memory; and then run the generalization algorithm described in Section 5.3.2 on each of the subsets looking for successful strings within those subsets. Once all of the strings have been analyzed for a particular subset, the algorithm then starts on the next subset. After all of the subsets have been analyzed, the successful strings from all of those subsets are combined and then tested for k-anonymity. Any successful strings from these combined strings are then tested for any attributes that are at the root level and the string with the highest RV is applied to the raw database. Since the datasets we used only produced successful generalization strings using three and six attributes, the distributed approach was not needed, but as we

increase the number of attributes beyond twelve, the distributed approach will be needed to ensure the scalability of the algorithm.

6 Experiments

6.1 Algorithm Performance

The performance of the local optimization algorithm is $\log_2$(max height of n attributes in generalization string) is based on the fact that the local algorithm uses a binary search technique, and it repeats until no more moves are allowed in any of the attributes. For the global optimization algorithm, the performance of the algorithm is $\log_2$ (generalization of all strings).

6.2 Utility Measurement

6.2.1 MCPHD Dataset

The global optimization utility metric algorithm was tested using multiple k values on the Marion County death certificate database to test how the algorithms would perform. For this dataset, the research values were established by the data content expert and not the utility function that was introduced in Section 4. Currently, we are testing our algorithms with the research values generated using our utility function to compare the outcomes from the values generated by the data content experts and our new utility function. We plan on submitting this as a future publication.

Results from the MCPHD dataset using k values of three and five with multiple combinations of attribute are shown in Table 4. As the k value increases, the amount of successful records drops off dramatically (in most cases, there were zero successful strings found) for datasets that contained more than twenty-four attributes. So the data is not shown for those cases. We discuss the possible reasons for no successful generalization strings using the MCPHD dataset in Section 7. In Table 4 and Table 5, any empty entries found in the tables indicate that no successful generalization string was found for that run. Datasets mYY contain the attributes that have the most generalization levels within the attribute, while the rXX datasets have randomly selected attributes. m12 had fewer total strings due to the fact that the pre-pruning phase eliminated a considerable amount of generalization levels in the attributes for that dataset, and thus the total number of combinations of generalization strings was less than the r12 for example.

Table 4: Global Optimization Utility Algorithm

K Value	Run Time (Secs)	Data-set Name	Total Strings Gener-ated	Highest RV of Success-ful String	Mini-mum RV	Maxi-mum RV
3	8	m03	1	0.15	0.15	0.15
5	9	m03	1	0.15	0.15	0.15
3	41	m08	28	1.10	0.15	2.00
5	41	m08	28	1.10	0.15	2.00
3	49	m12	28	1.00	0.25	1.85
5	49	m12	28	1.00	0.25	1.85
3	25764	m24	44800		0	6.03
3	17	r03	20	0.95	0	1.75
5	13	r03	5	0.48	0	0.995
3	270	r08	700	1.00	0.10	3.67
5	282	r08	700	1.00	0.10	3.67
3	2459	r12	8400	1.00	0.10	4.87
5	2498	r12	8400	1.00	0.10	4.87
5	14961	r24	24		0	6.07

Table 4 shows the different runs that use K values of three or five. Within each run, the highest total research value for the dataset is listed along with the maximum and minimum research values for the run. The maximum research value corresponds to an anonymized dataset that contains attributes that all contain their most specific generalization levels, while the minimum research value corresponds to an anonymized dataset that contains attributes that all contain their most generic generalization levels. The empty entries in the highest RV of a successful string column indicate that no selected generalization strings passed the k-Anonymity test.

Table 5 shows the results of running the local optimization utility algorithm under the same k value conditions as the global algorithm. This table contains the same fields as that of the global optimization utility algorithm to allow for comparisons of the two algorithms on the same datasets.

Table 5: Local Optimization Utility Algorithm

K Value	Run Time (Secs)	Dataset Name	Total Strings Generated	Highest RV of Successful String	Minimum RV	Maximum RV
3	17	m03	1	0.15	0.15	0.15
5	9	m03	1	0.15	0.15	0.15
3	32	m08	28	1.10	0.15	2.00
5	42	m08	28		0.15	2.00
3	42	m12	28	1.15	0.25	1.85
5	52	m12	28		0.25	1.85
3	119	m24	44800	1.20	0.25	6.03
3	12	r03	20	0.95	0	1.75
5	11	r03	5	0.95	0	0.95
3	40	r08	700	1.05	0.10	3.67
5	59	r08	700		0.10	3.67
3	54	r12	8400	1.05	0.10	4.87
5	73	r12	8400		0.10	4.87
5	115	r24	56000		0.25	5.97

6.2.2 Adult Census Dataset

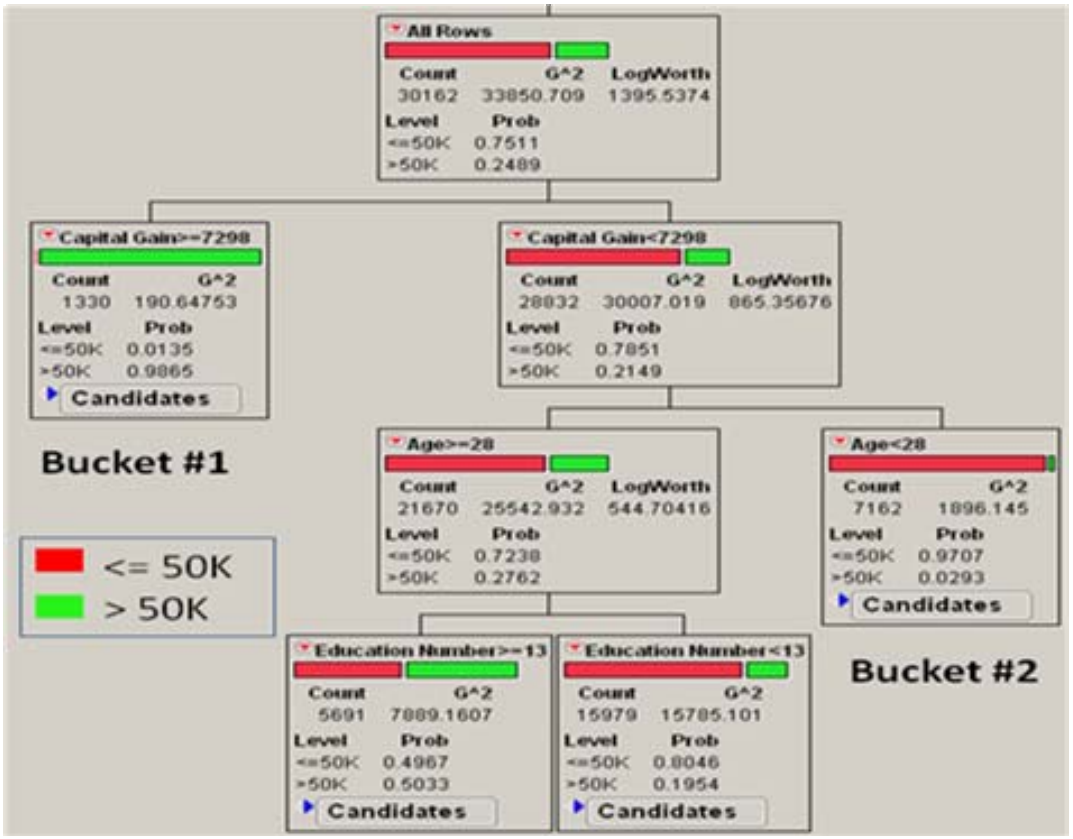
As a means to show how our algorithm performs against existing utility methodologies, our two proposed algorithms were run using the public Adult census dataset using a range of *k* values (k=3, k=5 and k=10).

Unfortunately, the local optimization algorithm was not able to find any solutions using the Adult DB, and this will be addressed in the next section. We then ran the Bottom Up algorithm as defined in[28] using the same three values of k to compare with our methodology and utility metric. The authors of [29] presented a Top Down and Bottom Up algorithm, but both showed very similar results, with the only difference being the execution time, which was not a concern for us in this exercise. For this dataset, we did use our new utility function to establish the research values for each of the attributes and the generalization levels of those attributes.

In order to examine the effects of the anonymization process, we used recursive partitioning (RP), which is a multivariable technique that is used to find patterns in large datasets, on the raw Adult dataset to see which of the attributes

were most responsible for differentiating individuals who make <=50K or >50K in yearly salary. Salary was chosen, because it is the attribute of interest in the Adult dataset for analysis. As seen in Figure 6, out of the original 30162 records, 75% of the individuals had a yearly salary of <=50K and 25% had a salary >50K.

Figure 6 -- Raw Adult Dataset Recursive Partitioning



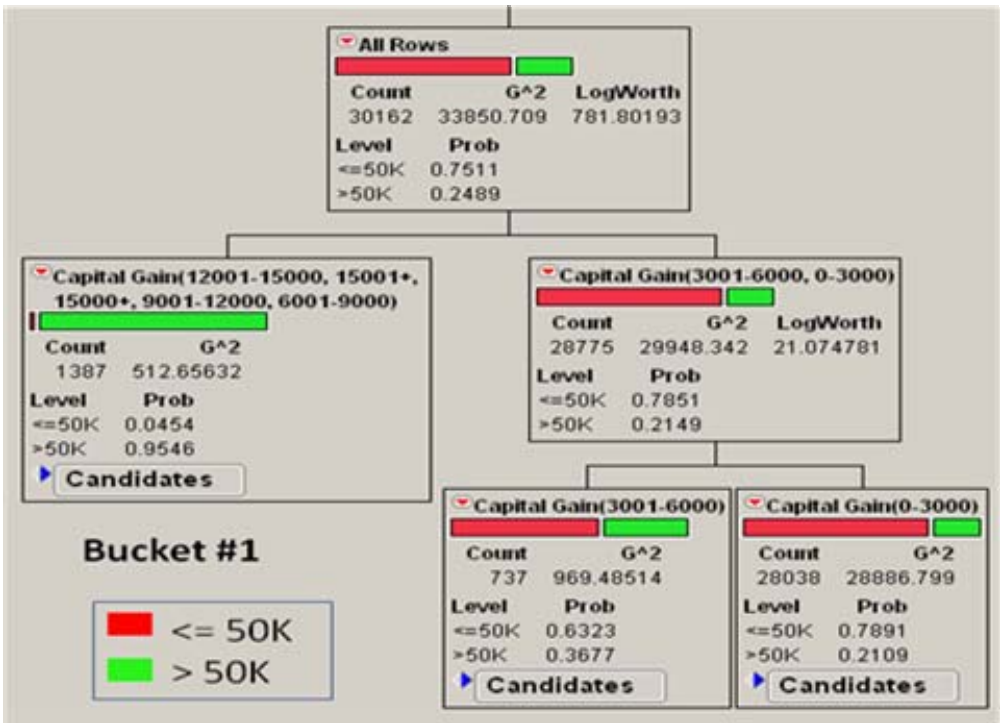
The three attributes that significantly differentiated the two salary groups are Capital Gain, Age and Education Number. Bucket #1 indicates that when an individual has Capital Gains >=$7298, 98% of the 1330 individuals have a yearly salary of >50K. On the other hand, bucket #2 shows that when Capital Gains < $7298 and Age is < 28 years old, 97% of 7162 individuals have salaries <=50K.

After running the global optimization algorithm using a k value of 5, the anonymized dataset was analyzed using recursive partitioning and it produced the breakdown as shown in Figure 7. Bucket #1 shows that when the Capital Gain is >=$6001, 95% of the 1387 individuals have a yearly salary of >50K.

The Bottom-Up algorithm with a k value of 5 was also run against the Adult dataset and the recursive partitioning results are shown in

Figure 8. Bucket #1 has a mixture of Capital Gains that range from zero to $15,000+, so no conclusions can be drawn from this bucket. When the Education Number is Pre-college for all values of Capital Gains, 86% of the 18686 individuals from the 516 clusters in bucket #2 have Salaries <=50K.

Figure 7 -- Global Optimization RP using k=5



Both algorithms were also run on the Adult dataset using a k value of 10. The Global Optimization algorithm did not produce a valid solution where the Salary attribute is not generalized to a value of both <=50K and >50K. On the other hand, the Bottom-Up algorithm produced a result that is found in Figure 9.
As in the previous runoff of the Bottom-Up algorithm, Bucket #1 had Capital Gain represents a full range of values. Therefore no conclusion can be determined. Bucket #2 has a full range of Capital Gain values and Education Numbers of Pre-College have 86% of the 18686 individuals have a Salary <=50K. When the value of k was raised to be 10, neither the local nor the global optimization algorithms could produce a solution where the salary attribute did not contain the most generalized values for the salary (i.e. salary="both"). In contrast, Figure 9 shows that the Bottom Up algorithm was able to find a solution. As present in the k=5 solution, Bucket #1 had the full range of Capital Gain val-

ues. Bucket # 2 represents Capital Gain values <$7000 that have 86% of 20,000+ individuals having a Salary <=50K.

Figure 8 -- Bottom-Up Recursive Partition using k=5

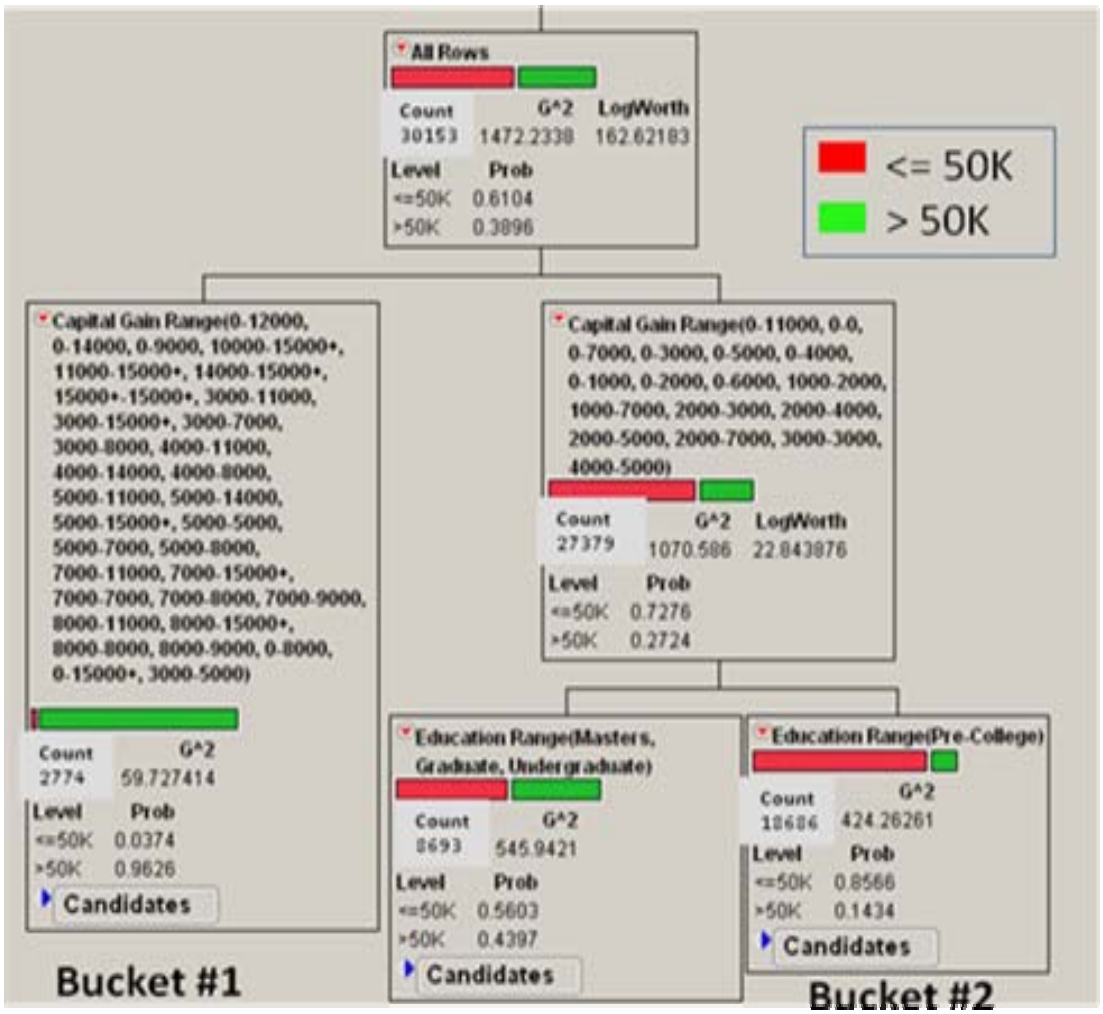
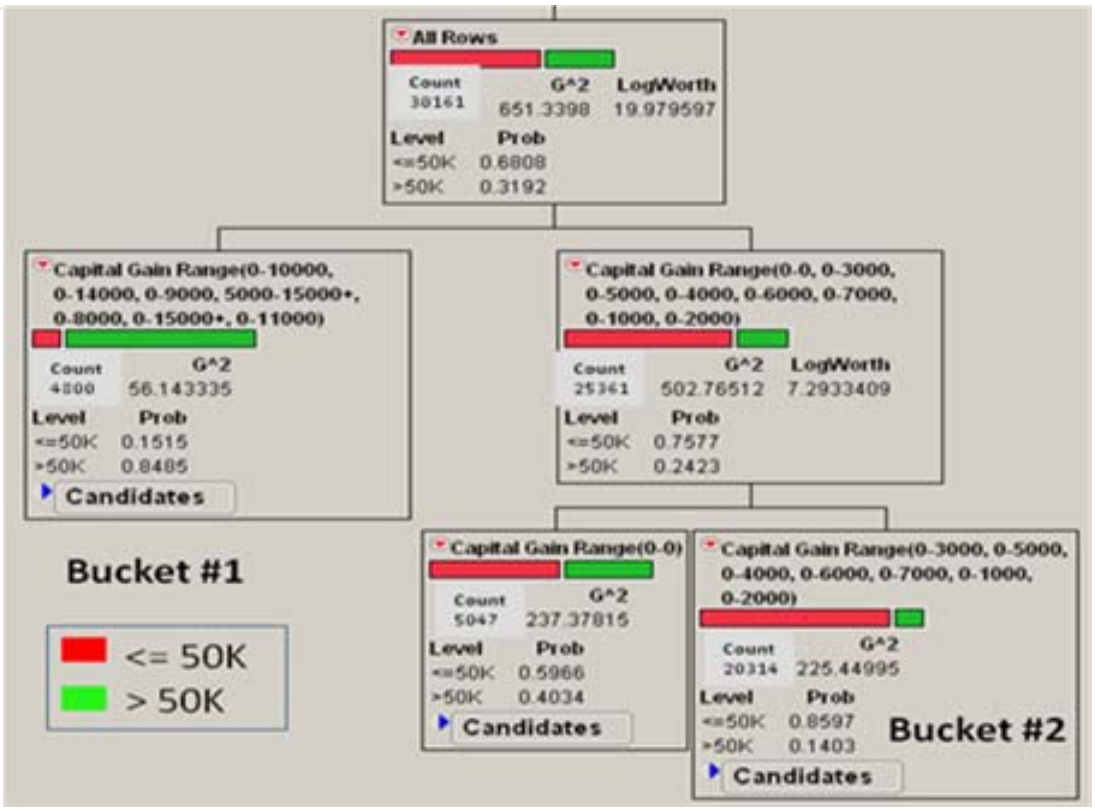


Figure 9 -- Bottom-Up Recursive Partition k=10



7 Discussion

The goal of the algorithms described in Section 5.3 is the optimization of data utility in a dataset while still protecting the anonymity of the individuals stored in that dataset. As shown in Table 4 and Table 5, the global optimization utility metric algorithm and the local optimization algorithm are extremely close in terms of the highest research value of the successful strings.

7.1 Marion County Public Health Department

For the Marion County Health Department database, the local optimization algorithm (Table 5) performed better than the global optimization algorithm when you compare the highest successful research value discovered relative to the maximum total research value using the mXX datasets, where XX was under 12 attributes. As the number of attributes increased in the mXX series, the global algorithm produced higher success research values due to the increased number of success strings that were found during the execution of the algo-

rithm. The local optimization algorithm failed to find a success string in three of the mXX tests, while the global optimization algorithm only failed to find a successful string when 24 attributes were used. Upon examination of the distribution of the data when all of the attributes were used by both algorithms, it became apparent that there were many outlier instances where the combination of all the attributes failed to produce a record count greater than one, This leads us to believe that certain large datasets will require some sort of suppression of records after an overall distribution analysis has been completed. These records would be selected for suppression with the goal of minimizing the impact on data utility, but this exercise is out of the scope of this paper.

The ability of the global optimization algorithm on the rXX dataset was far superior to that of the local optimization algorithm in regards to both the highest research value discovered and the ability at least one successful generalization string. As expected, the time performance to run the two algorithms favoured the local algorithm due to the fact that the local algorithm only examines a small subset of all possible generalization strings when compared to those examined by the global algorithm. This explains why the local algorithm did not find success strings for all of the experiments. For this study, we examined all possible combinations off-line of each dataset without concern for performance and without pruning. This was done to determine all possible successful generalization strings from that dataset to ensure that the global optimization algorithm was in fact correctly executing and finding the maximum utility in the research value while still upholding the given k-anonymity criteria.

The results from the hybrid approach, which were not included in Table 4 or Table 5, mirrored the global algorithm in terms of execution time and the highest research values found for a successful string, which is not surprising due to the fact that the local algorithm only examined a small subset of all possible strings and then removed them from the total pool of possible strings. After the local algorithm completed its execution, the global algorithm then examined the remaining possible strings, which in the case of an attribute pool of 24 attributes, was a large number of strings; thus the time and successes swayed toward the run results where only the global algorithm was executed.

As for the distributed approach, it was used when the number of attributes in a testing datasets surpassed 18 attributes due to memory limitations of Java. This allowed the global optimization algorithm to effectively examine the set of 24 attributes in a timelier manner, even though no successful strings were discovered where k was satisfied due to the presence of outliers in the dataset, which prevented a tuple (record) count to surpass the k criteria as described in the previous paragraph.

7.2 Adult Census Dataset

Both the Local and Global Optimization algorithms were run using the Adult dataset from the UCI website, using multiple values of k (k=3, k=5 and k=10). Additionally, the Bottom-Up algorithm [28] was run on the same Adult dataset. Due to the limited size of the Adult DB, the local algorithm was not able to find any successful strings among the subset of generalization strings that it examined. We are currently re-examining how each attribute is being grouped in the different generalization levels to see if that will aid in the discovery of a successful generalization string while only examining a small subset of all possible strings.

On the other hand, the global optimization algorithm performed quite well. As demonstrated in Figure 7, the anonymized dataset using a k value of 3 or 5 was able to maintain the pattern discovered by the raw dataset where individuals who have Capital Gains >=$6001 had yearly salaries of >50K. In contrast, the Bottom-Up algorithm when run using a k value of 3 or 5, the same pattern was lost, because the Capital Gain in Bucket #1 covered the full range of values. Although the Bottom-Up algorithm produced a result for Bucket #2 to differentiate the individuals who have a yearly salary <=50K, the wide range of Capital Gains and the overlapping of groupings of those Capital Gains diminish the impact of that discovery. Similarly, the Bottom-Up algorithm failed to discover the pattern in Bucket #1 due to the full range of Capital Gain values, and the Bucket #2 had overlapping values and did not find the Education Numbers of Pre-College.

When the k value was raised to 10, our algorithm was not able to find any solutions that did not include the most generalized values for the salary attribute. In contrast, the Bottom Up algorithm had Bucket #1 that had the Capital Gain representing a full range of values therefore no conclusion can be determined. Bucket #2 had Capital Gain values < $7000K where 86% of the 20000+ individuals had a Salary of <=50K.

Similar to the MCPHD, we examined off-line all possible combinations of the generalization strings to ensure that the algorithms were discovering the highest utility strings that passed the given k-anonymity criteria. Although the utility metric defined in [28] can help understand the penalty associated with an anonymized dataset, the NCP does not take into account different sizes of groups within a generalization level, nor does it account for range thresholds defined by the data expert to be critical for data mining exercise, or categorical values that must be maintained within a generalization level that could be used to find patterns or trends in a dataset. Additionally, the Bottom-Up algorithm does not prevent attributes from having overlapping values, and this dramatically dimin-

ishes the utility of the anonymized dataset as demonstrated in the previous section.

8 Summary and Future Work

In summary, we have introduced two approaches for achieving k-anonymity while aiming at maximizing a user driven data utility. Each algorithm has its strengths and weaknesses, but as described before, the ultimate goal is to create a generalized dataset that maximizes the utility of the transformed data. Unfortunately, data utility and anonymity are in an inverse relationship; as we try to improve data utility, we expose the confidentiality of the data, and vice versa. Therefore, we must find an acceptable balance between the two. The Global Optimization algorithm was shown to outperform the Bottom-Up utility algorithm using recursive partitioning.

Based on the experiments we performed, global optimization utility metric algorithm seems to provide this balance although it may require longer run times as the number of attributes in a dataset increase. The pre-pruning portion of the algorithm helped to cut down on the unnecessary database calls with generalization strings that fail k-anonymity. In order to maximize the data utility of a given dataset, it was necessary to rank the importance of the attributes relative to each other, and also within each attribute. Our research value metric is an attempt to encapsulate these constraints. The algorithms we described implementing this concept have the ability to eliminate unwanted generalization strings while still reaching a desirable solution that satisfies both k-anonymity and data utility for a researcher.

We are currently working on an automated generalization approach that clusters tuples together to satisfy anonymity requirements only when it maximizes our proposed RV utility metric. Preliminary results have shown that our automated approach improves upon existing methodologies.

REFERENCES

1. Aggarwal, G. *On k-anonymity and the curse of dimensionality*. in *31st international conference on very large data bases*. 2005. Norway.

2. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigraphy, R., Thomas, D., Zhu, A., *Approximation Algorithms for k-Anonymity.* Journal of Privacy Technology, 2005.

3. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Thomas, D., Zhu, A. *Anonymizing Tables*. in *10th International Conference on Database Theory*. 2005.

4. Bayardo, R., Agrawal, R., *Data Privacy Through Optimal k-Anonymization*, in *Proceedings. 21st International Conference on Data Engineering*2005. p. 217-228.

5. Chaytor, R., *Utility-preserving k-anonymity*, in *Department of Computer Science*2006, Memorial University of Newfoundland. p. 82.

6. DataMart. *DataMart*. Available from: http://health.mil/mhscio/programs_products/jmis/dhss/products/cdm.aspx.

7. Duncan, G., Fienberg, S., Krishnan, R., Padman, R., Roehrig, S., *Disclosure Limitation Methods and Information Loss for Tabular Data*, in *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*2001. p. 135-166.

8. Evfimievski, A., *Randomization in privacy preserving data mining.* ACM SIGKDD Explorations, 2002. **4**(2): p. 43-48.

9. Gionis, A., Mazza, A., Tassa, T. *k-Anonymization Revisted*. in *24th International Conference on Data Engineering* 2008. Cancun, Mexico: IEEE.

10. Hua, M., Pei, J., *A Survey of Utility-based Privacy-Preserving Data Transformation Methods*, in *Privacy-Preserving Data Mining*2008, SpringerLink. p. 207-237.

11. Iyengar, V., *Transforming Data to Satisfy Privacy Constraints*, in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining*2002.

12. Kifer, D., Gehrke, J. *Injecting Utility into Anonymized Datasets*. in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*. 2006. ACM Press.

13. LeFevre, K., DeWitt, D., Ramakrishnan, R. *Incognito: Efficient Full Domain K-Anonymity*. in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* 2005.

14. LeFevre, K., DeWitt, D., Ramakrishnan, R. *Mondrian Multidimensional K-Anonymity*. in *In IEEE International Conference on Data Engineering (ICDE)*. 2006.

15. Li, N., Li, T., Venkatasubramanian,S., *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*, in *2007 IEEE 23rd International Conference on Data Engineering* 2007.

16. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam., M., *L-Diversity: Privacy Beyond k-Anonymity.* ACM Transactions on Knowledge Discovery from Data, 2007. **1**(1): p. 12.

17. Meyerson, A., Williams, R., *On the complexity of optimal k-anonymity*, in *23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*2004.

18. Repository, U.C.I.M.L.; Available from: http://www.ics.uci.edu/mLearn/mlrepository.html.

19. Samarati, P., *Protecting Respondents' Identities in Microdata Release.* J IEEE Transactions on Knowledge and Data Engineering, 2001. **13**(6): p. 1010-1027.

20. Samarati, P., Sweeney, L., *Protecting privacy when disclosing information:k-anonymity and its enforcement through generalization and suppression*, 1998, SRI Computer Science Laboratory.

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

22. Sweeney, L., *K-Anonymity: A model for protecting privacy.* International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, 2002. **10**(5): p. 557-570.

23. Verykios, V., Bertino, E, Fovino, I., Provenza, L., Saygin, Y., Theodoridis, Y., *State-of-the-art in Privacy Preserving Data Mining.* SIGMOD Record, 2004. **33**(1): p. 50-57.

24. Willenborg, L., deWaal, T., *Elements of Statistical Disclosure Control*. Vol. 155. 2001: Springer. 261.

25. Willenborg, L., deWaal, T., *Elements of Statistical Disclosure Control*. Springer Verlag Lecture Notes in Statistics. Vol. 155. 2000: Springer.

26. Wortman, J., Adam, N., *Security-Control Methods for Statistical Databases: A Comparative Study.* ACM Computing Surveys, 1989. **21**(4): p. 515-556.

27. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A. *Utility-based Anonymization for the Privacy Preservation with Less Information Loss*. in *ACM SIGKDD Explorations*. 2006.

28. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A. *Utility-based anonymization using local recoding*. in *Twelfth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. Philadelphia, PA: ACM Press.