AN IMPROVED UTILITY DRIVEN APPROACH TOWARDS K-ANONYMITY USING DATA CONSTRAINT RULES

Stuart Michael Morton

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the School of Informatics,
Indiana University

October 2012

Accepted by the Faculty of Indiana University, in partial
fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Mathew Palakal Ph.D., Chair

Malika Mahoui Ph.D.

P. Joseph Gibson Ph.D.

Hadi Kharrazi Ph.D.

July 26, 2012

ACKNOWLEDGMENTS

I would like to thank all of my coauthors for the work in the various chapters, especially Dr. Malika Mahoui, Dr. P Joseph Gibson and Saidaiah Yechuri for their significant contributions to this dissertation. I would also like to thank Dr. Hadi Kharrazi and Dr. Mathew Palakal for their advice and comments. Finally, I would like to thank my wife and my parents for all of their support over the last five years.

ABSTRACT

Stuart Michael Morton

AN IMPROVED UTILITY DRIVEN APPROACH TOWARDS K-ANONYMITY USING DATA CONSTRAINT RULES

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. Now more than ever, it is critical to protect the identities of the patients contained in these databases. Even after removing obvious “identifier” attributes, such as social security numbers or first and last names, that clearly identify a specific person, it is possible to join “quasi-identifier” attributes from two or more publicly available databases to identify individuals.

K-anonymity is an approach that has been used to ensure that no one individual can be distinguished within a group of at least k individuals. However, the majority of the proposed approaches implementing k-anonymity have focused on improving the efficiency of algorithms implementing k-anonymity; less emphasis has been put towards ensuring the “utility” of anonymized data from a researchers’ perspective. We propose a new data utility measurement, called the research value (RV), which extends existing utility measurements by employing data constraints rules that are designed to improve the effectiveness of queries against the anonymized data.

To anonymize a given raw dataset, two algorithms are proposed that use pre-defined generalizations provided by the data content expert and their corresponding research values to assess an attribute’s data utility as it is generalizing the data to ensure k-anonymity. In addition, an automated algorithm is presented that uses clustering and the RV to anonymize the dataset. All of the proposed algorithms scale efficiently when the number of attributes in a dataset is large.

Mathew Palakal Ph.D., Chair

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

AN ENHANCED UTILITY-DRIVEN DATA ANONYMIZATION METHOD

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 a 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.

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 [9]. 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) [38, 40]. Under query restriction three techniques have been utilized, data partitioning, cell suppression, and query size control (also called blocking) [37]. 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 [33]. 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 [4, 26].

Few contributions have focused on the utility of the information when it is transformed to satisfy k-anonymity. Work such as in [22, 33, 42] 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. [42] 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. [30], 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 [7], 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).

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.

Attribute Identifiers

Let $T= \{t_1,t_2,\ldots t_m\}$ be a table storing information about individuals, described with a set of attributes $A = \{A_1, A_2, \ldots ,A_n\}$. 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 labeled 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 [11]. 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.

Quasi-Identifier Attribute

A set of attributes $\{A_1, A_2, \ldots, A_n\}$ 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 [25]. 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.

Frequency Set

Let $Q= \{A_1, A_2, \ldots, A_q\}$ be a subset of A. The frequency set of T with respect to Q is a mapping from each unique combination of values $\{v_0, \ldots v_q\}$ 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.

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 [34]. Similar to [20], 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:

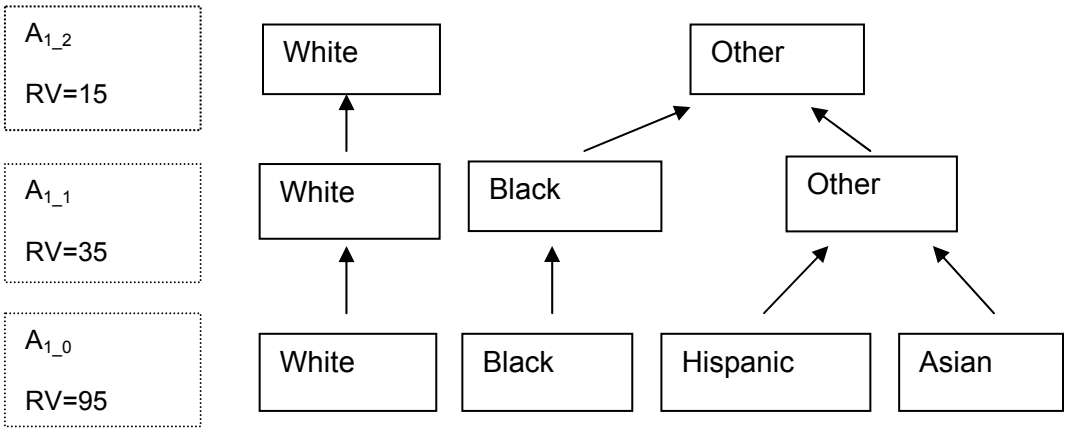
select min(myCount) as count from (select count(*) as myCount from DB1 group by q_1, q_2)

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

Attribute Generalization and Suppression

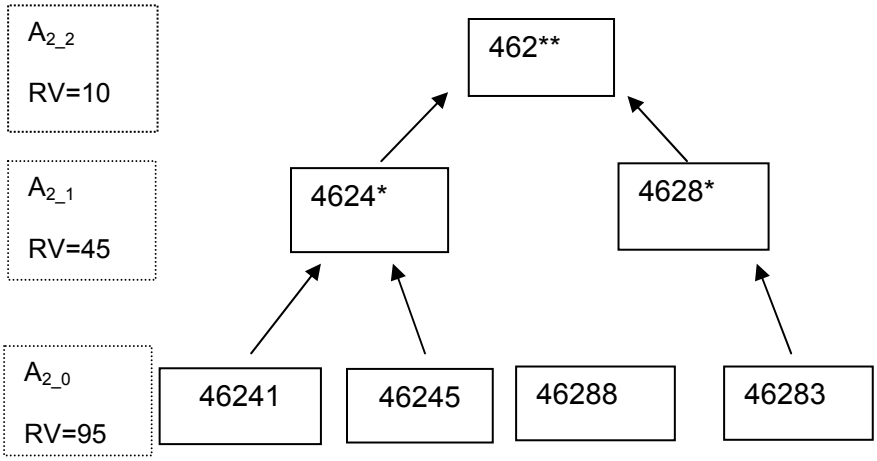
The basic idea of generalization is to abstract the domain of attributes to make it more difficult to distinguish individual values and therefore increasing the chances of achieving k-anonymity. Examples of generalization include generalizing zip code values by replacing the last digit with wild card (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 represented by a many-to-one generalization function denoted by γ. By Convention i<j; and D_{l0} represents the most specific domain (also noted D_l) for attribute A_l.

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 γ+. 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}.

Full Domain Generalization

As described in [8, 38], 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 [7, 8] 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 unique value $\Phi i(q)$ [30, 32]. 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 the Utility Metric section.

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 the Utility Metric section.

Related Work

The protection of microdata has been an active research issue [39], 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 [32], and a binary search algorithm to find a single full-domain generalization was described in [30]. New optimization methods were developed in [2, 4, 7, 20, 21, 26]. For example in [20], the authors introduce a class of algorithms for a multi-dimensional 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 [25], 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, [22] 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 in a dataset 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 [25]. 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 [7, 16, 21] 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 [7], 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, 28] 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

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

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.

Utility Metric

We propose that the data curators should have more control 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 critical thresholds in the data that should be maintained through the anonymization process, so that the meaning of the data is well maintained. In [27, 28], a new utility measurement called the research value (RV) was introduced to encapsulate the utility of the each attribute with respect to the following conditions:

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

A data constraint rule (DCR) defines groupings of data or ranges of data that, if maintained, help 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 the 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, the data constraint rules could define inflection points of ages that would be important for someone who is interested in mining the data. 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.

Table 2. Possible Data Constraint Rules for Numerical Attribute

Attribute	Constraint Rule	Importance value
Age	64-65	25
Education Number	12-13	20
Capital Gain	10000-10001	45

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, $\sum DR_0$. Each dataset has an inherit value even if all of the data constraint rules are broken during the merging of two clusters, so to resolve this, the data owner has the ability to add a base value to the data constraint ratio, so that it does not zero out a RV if all of the rules are broken. The weight

calculation of each attribute can be determined by establishing a correlation matrix among 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:

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

where all the elements are defined to be the same as those for the numerical attributes, except for the $\frac{|A_k|}{|A_o|}$ 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 k^{th} 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, which include the base value as determined by the data owner, that exist at the k^{th} level is 50, while 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 k^{th} 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. 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 [10], 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.

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.

DataSets

For this project, we utilized two datasets, the public Adult Census data from the UC Irvine machine learning repository [29], 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 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 [41] 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 3. 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 3. 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 4 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 4. 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)

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. This is a weakness of the local recoding methodology. Publications using local recoding where values are allowed to cross groupings during the anonymization process include [3, 4, 13, 42]. 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.

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.

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.

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 (GS_i) from L, sort by total RV. Store in List A
2. While there exists a generalization string GS_i in List A {
3. Select GS_i as midpoint(List A)
4. If min(count(GS_i)) >= *k* {
5. Add GS_i to success list *SL,*
6. Remove all GS_k from List *A* with $RV_k < RV_i$
7. Remove all GS_k from List *A* where $gl_k > gl_i$
8. }
9. Else {
10. Remove all GS_k from List *A* where $gl_k < gl_i$
11. }
12. Remove GS_i from List *A*
13. } // End while
14.
15. For each GS_i in *SL* {
16. Determine root # of attributes where $gl_x = gl_{MAX}$
17. }
18.
19. Apply GS_i from *SL* with (>> RV_k && 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.

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 the next section 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.
13. For each GS_i in *SL* {
14. Determine root # of attributes where $gl_x = gl_{MAX}$
15. }
16.
17. 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.

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.

Distributed Version of the Global Optimization

As described in Global Optimization Algorithm section, 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 Global Optimization Algorithm section. 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 the Global Optimization Algorithm section 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.

Experiments

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).

Utility Measurement

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. 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 5. 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 will discuss the possible reasons for no successful generalization strings using the MCPHD dataset. In Table 5 and Table 6, 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 5. Global Optimization Utility Algorithm

K Value	Run Time (Secs)	Dataset Name	Total Strings Generated	Highest RV of Successful String	Minimum RV	Maximum 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	1.2	0.25	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 5 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 6 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 6. 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

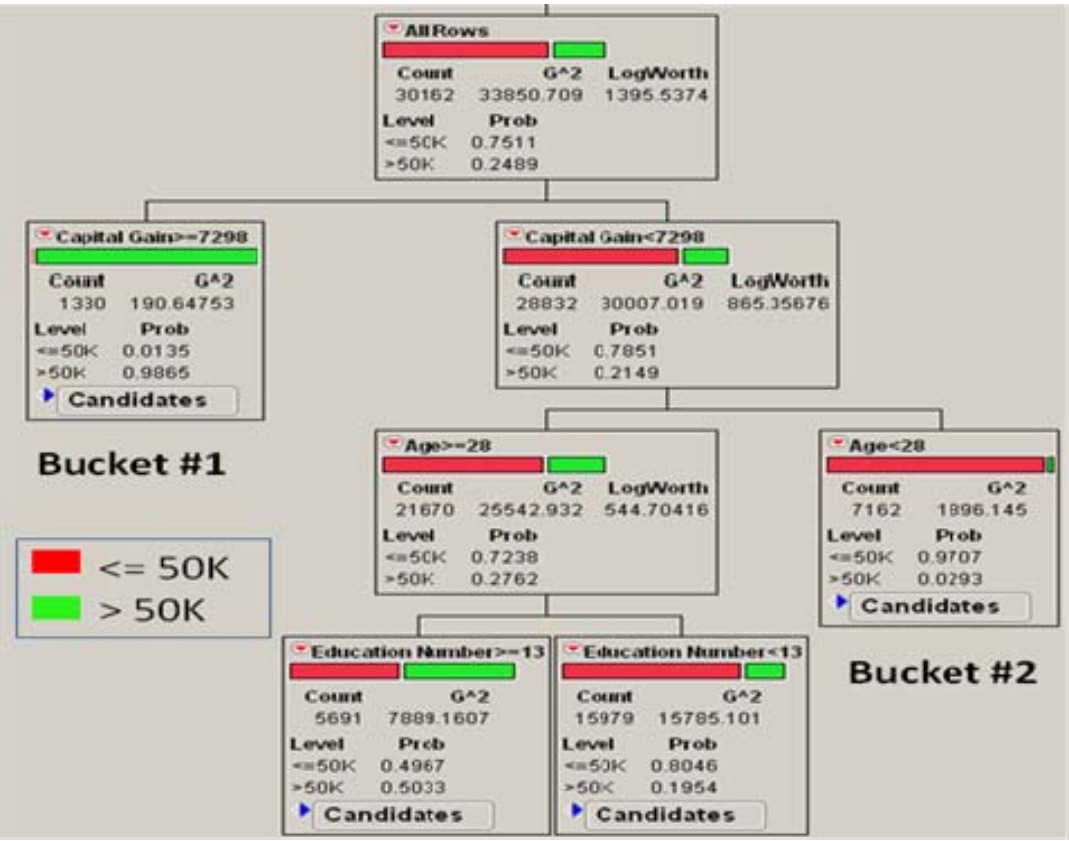
Adult Census Dataset

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 [42] 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

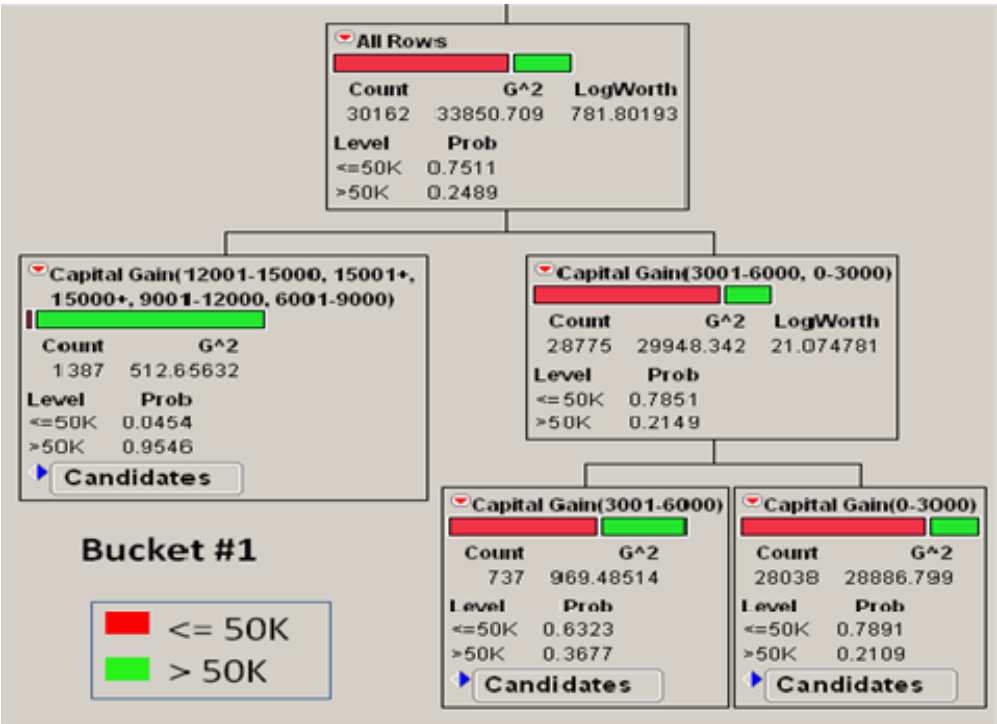


The three attributes that significantly differentiated the two salary groups are Capital Gain, Age and Education Number. Bucket #1 indicates that when an individual has Capital Gains >=$7298, 98% probability of the 1330 individuals having a yearly salary of >50K. On the other hand, bucket #2 shows that when Capital Gains <$7298 and Age is <28 years old, 97% probability of 7162 individuals having 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% probability of the 1387 individuals having 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 values. 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

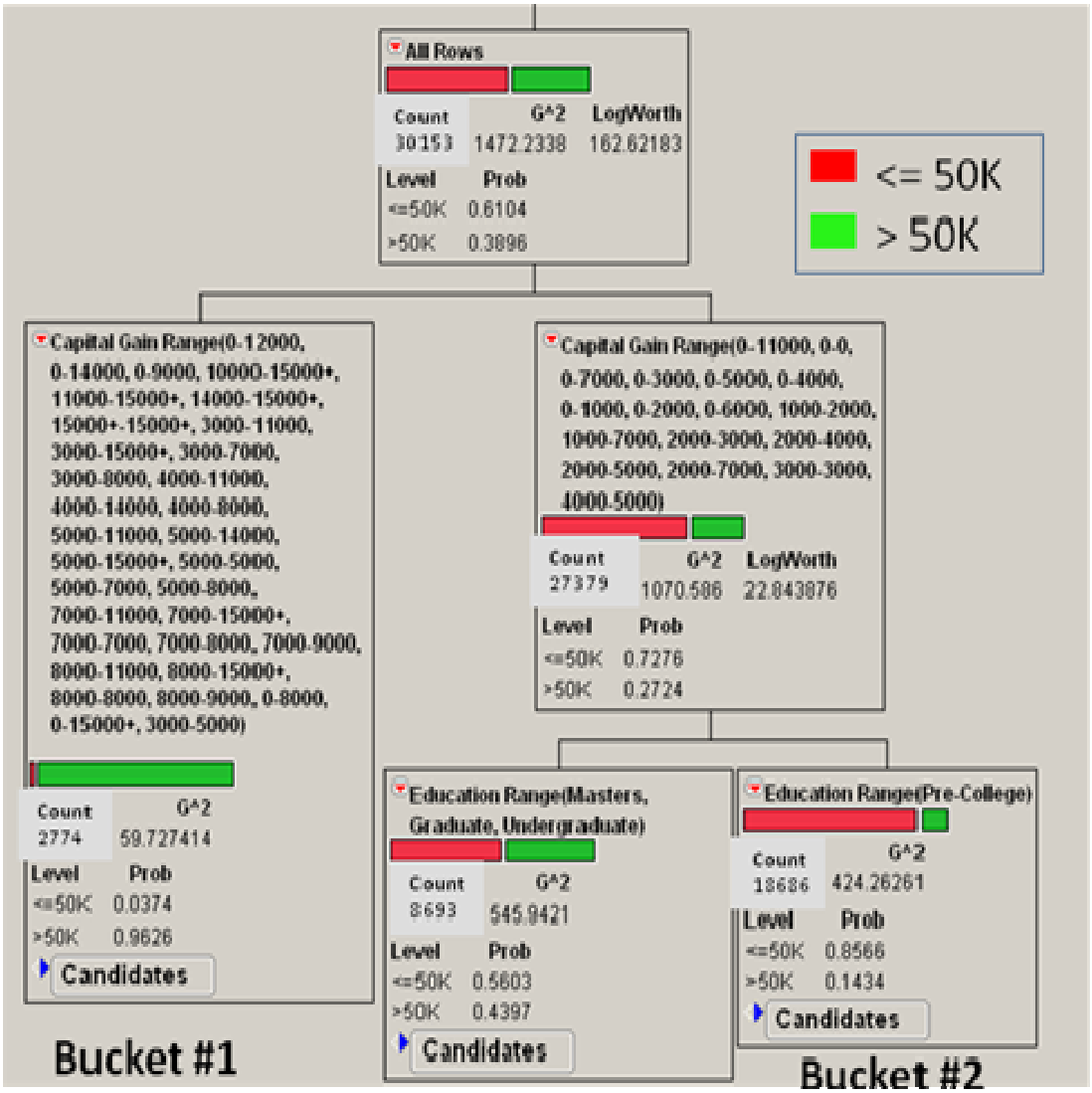
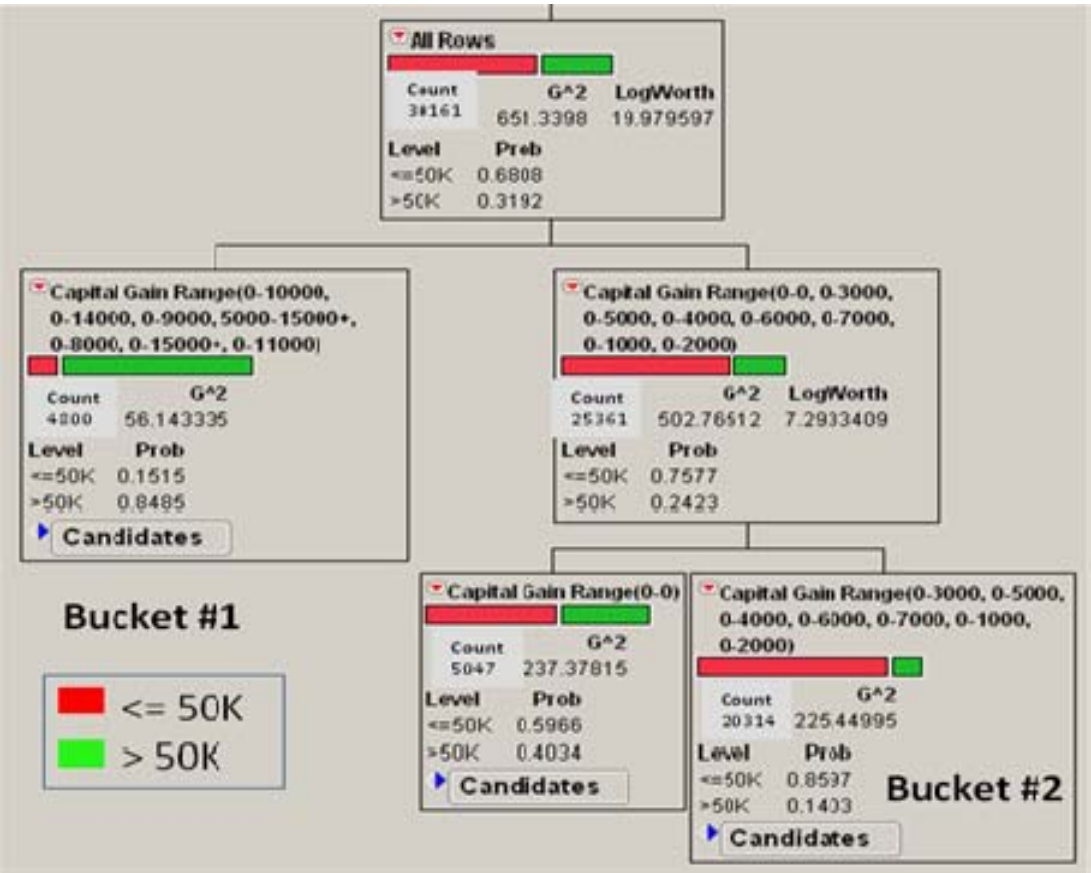


Figure 9. Bottom-Up Recursive Partition k=10



Discussion

The goal of the algorithms described in the previous section provide the optimization of data utility in a dataset while still protecting the anonymity of the individuals stored in that dataset. As shown in Table 5 and Table 6, 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.

Marion County Public Health Department

For the Marion County Health Department database, the local optimization algorithm (Table 6) 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 algorithm. The local optimization algorithm failed to find a success string in three of the mXX tests, while the global optimization algorithm was able to find a successful string for each of the mXX datasets. 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.

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 [42] 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 <$7000 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 [42] 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 diminishes the utility of the anonymized dataset as demonstrated in the previous section.

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.

AN IMPROVED DATA UTILITY CLUSTERING METHODOLOGY USING DATA CONSTRAINT RULES

Abstract

Many data privacy models have been built in the last few years using the *k*-anonymization methodology including *l*-diversity, p-sensitive k-anonymity, and *t*-closeness. While these methods differ in their approaches and quality of the results, they all focus on ensuring the anonymization of the data while at the same time attempt to protect the quality of the data by minimizing the loss of the information contained in the original data set. In this paper, we propose an automated k-anonymity approach that uses clustering to maximize the utility of the data while ensuring that the data privacy is maintained. Our method employs data constraint rules, which are defined by the data research expert to represent especially informative distributions in categorical attributes or inflections points in a continuous attribute. The values of the data constraints are an integral component of our utility function, which is used to maximize the utility of the anonymized dataset. Finally, we present our experimental results that show that our approach meets or exceeds existing methods that do not incorporate data constraint rules.

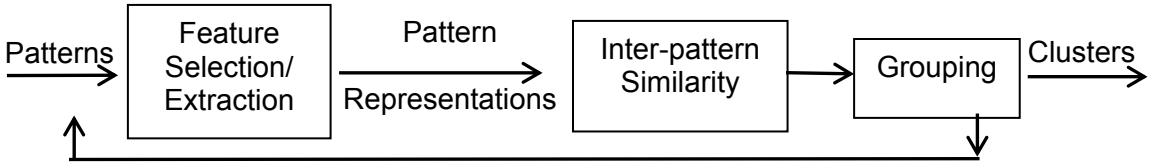
Introduction

The explosion of personal data that is now electronically available from multiple application areas, such as finance, healthcare, and social networks, has increased concern about protecting the privacy of the individuals that are represented in those datasets before the data is released for secondary uses. Existing government regulations stipulate that the privacy of the individuals described in electronic datasets must be protected during their collection, storage, distribution and use [14]. It is usually necessary to modify the original raw dataset before it is released for other use, such as by healthcare researchers identify risk factors in a community.

A method used that is widely used to protect the privacy of individuals is modification or masking, where the original values in a dataset are changed so that an adversary cannot link records from multiple data sources to expose the private information of an individual [5]. Existing anonymity models for masking are k-anonymity [33, 34] or one of its follow-up augmentations: l-diversity [25], p-sensitive k-anonymity [36], and t-closeness [22]. At its roots, k-anonymity was designed to ensure that the identity of an individual cannot be reversely identified within a set of k individuals. This is accomplished by satisfying the k-anonymity property that states that every tuple in a dataset is indistinguishable from at least (k-1) other tuples with respect to attributes defined to be quasi-attributes. These quasi-identifiers are defined in detail in the following section. Existing augmentations of k-anonymity ensure that either the count or the distribution of sensitive attributes, also defined in the next section, is guaranteed for every k group of tuples.

Recently, work in the area of k-anonymity has focused on clustering methods to provide data privacy and utility. Clustering is an unsupervised classification technique to put patterns (observations, data items, or feature vectors) into groups or clusters, and the main activities involved in clustering are shown in Figure 10 [1].

Figure 10. Stages in Clustering



The crucial component of any successful clustering algorithm is a similarity measurement that defined the distances between the feature spaces of two clusters. Early clustering methods used Euclidean distance, which is the straight line distance between two points. Traditional clustering methods require that a particular number of

clusters exist in the solution, but for a dataset to k-anonymous, each cluster in the solution must contain at least *k* records.

To translate conventional clustering methodologies to address the data privacy problem, [17] introduced the concept of the k-member clustering problem to anonymize a dataset while minimizing intra-cluster distances using a normalized distances between numerical values and hierarchical distances from a taxonomy tree for categorical data. In contrast, [12] cluster records into an arbitrary space according to a maximum cluster radius that contains at least r members. When the data is published, three types of features about the cluster are included: 1) quasi-identifier value for the cluster center, 2) number of elements in each cluster, and 3) a set of values for the sensitive attributes. Recently, a new anonymity clustering approach was presented in [6] where the clustering minimizes information loss during the anonymization process by creating a set of specified privacy constraints that can span clusters. The intent of the privacy constraints in the anonymization process is to limit generalization that obscures especially important distinctions within attributes.

In this work, we propose an automated clustering-based approach that is utility driven and minimizes the information loss. Unlike the existing approaches, our utility function uses the concept of data constraint rules that are based upon well-defined inflection points for numerical attributes and data expert groupings of categorical attributes. The contributions of this work are the following:

- We propose a novel clustering-based methodology inspired by the framework introduced in [41], but it is driven by a new utility function that extends the normalized certainty penalty to include the data constraint rules
- We implement two efficient anonymization algorithms based upon our clustering methodology. Both of the algorithms investigate all possible combinations in order to minimize the information loss while ensuring that the

dataset protects the privacy of the individuals represented in the dataset and the data constraint rules are followed as much as possible. The first algorithm clusters the data according to our new utility function, and the second algorithm extends the utility function to include the record suppression.

- Finally, we perform extensive experiments using a public benchmark dataset and a proprietary patient dataset. The results of our experiments confirm that our new algorithms meet or exceed existing algorithms.

Background and Related Work

In this section, we summarize the k-anonymity approach and principles as well as provide the basic concepts of clustering and the types of problems solved by clustering.

K-Anonymity

The basic definitions provided here are also presented in [21, 22] as we find that their description of attributes generalization is very concise and applies to our work.

Attribute Identifiers

Let $T= \{t_1,t_2,\ldots t_m\}$ be a table storing information about individuals, described with a set of attributes $A = \{A_1, A_2, \ldots ,A_n\}$. We distinguish three types of attributes in A, labelled as explicit identifiers, quasi-identifiers and sensitive identifiers as defined in [25]. 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 special attributes that contain data that are considered to be extremely personal, such as disease state or a salary.

Quasi-Identifier Attribute

A set of attributes $\{A_1, A_2, \ldots, A_n\}$ 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 [19]. It is assumed that the quasi-identifier attributes are known based upon the specific knowledge of the domain experts. In the work described in [25], 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.

Frequency Set

Let $Q= \{A_1, A_2, \ldots, A_q\}$ be a subset of A. The frequency set of T with respect to Q is a mapping from each unique combination of values $\{v_0, \ldots v_q\}$ of Q in T (the value groups) to the total number of tuples in T with these values of Q (the counts) [16]. 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.

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 [34]. Similar to [20], 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 then compared against the k-anonymity threshold value “k” for the combinations of attributes q_1 and q_2.

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 *****).

Figure 11. Generalization of the Race attribute

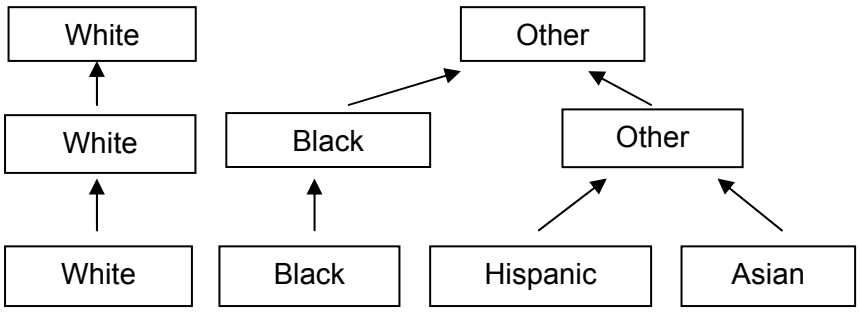


Figure 11 provides an example of the domain generalization hierarchy for the Race attribute. As an attribute approaches the most generalized level of a hierarchy tree, the information loss an attribute increases dramatically. Minimizing the height of an attribute in its hierarchy tree during the anonymization process will minimize the amount of information loss and increase the accuracy of queries executed against the generalized dataset.

Recoding Methods

Two specific types of generalization are local and global recoding [15, 21, 35, 42]. In global recoding, the data space is partitioned into a set of non-overlapping regions, such that all tuples from a single attribute domain region are mapped to the

same generalized or changed tuple [42]. Local recoding on the other hand can map an individual tuple to different generalized tuples. In other words, “allows the same detailed value to be mapped to different generalized values in each anonymized group [35].”

Table 7. Global and Local Recoding Example

a. The original table

tId	*Age*	*Education*	*Zip Code*	*Annual Income*	*Target Customer*
t1	24	Bachelor	53711	40*k*	Y
t2	25	Bachelor	53712	50*k*	Y
t3	30	Master	53713	50*k*	N
t4	30	Master	53714	80*k*	N
t5	32	Master	53715	50*k*	N
t6	32	Doctorate	53716	100*k*	N

3-anonymous table by global recoding

Age	*Education*	*Zip Code*	*Annual Income*	*Target Customer*
[24-32]	ANY	[53711-53713]	40*k*	Y
[24-32]	ANY	[53711-53713]	50*k*	Y
[24-32]	ANY	[53711-53713]	50*k*	N
[24-32]	ANY	[53714-53716]	80*k*	N
[24-32]	ANY	[53714-53716]	50*k*	N
[24-32]	ANY	[53714-53716]	100*k*	N

3-anonymous table by local recoding

Age	*Education*	*Zip Code*	*Annual Income*	*Target Customer*
[24-30]	ANY	[53711-53713]	40*k*	Y
[24-30]	ANY	[53711-53713]	50*k*	Y
[24-30]	ANY	[53711-53713]	50*k*	N
[30-32]	GradSchool	[53714-53716]	80*k*	N
[30-32]	GradSchool	[53714-53716]	50*k*	N
[30-32]	GradSchool	[53714-53716]	100*k*	N

Using the example from [15], you can see in Table 7 that the Age value 30 is mapped to different types of Education in the 3-anonymous table using local recoding, while the Age of 30 is mapped to only “Any” in the 3-anonymous table using global recoding. In general, local recoding may provide less information loss than global recoding, but it introduces the issue of mapping a single value to many values which is

detrimental when performing data analysis queries on the resulting anonymized datasets.

Related Work

K-anonymity and the deployment of generalization/suppression to satisfy k-anonymity were originally characterized in [31], and a binary search algorithm to find a single full-domain generalization was described in [30]. New optimization methods were developed in [2, 4, 7, 20, 21, 26]. For example in [15], the authors introduce a class of algorithms for a multi-dimensional 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 [25], 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, [17] 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 [25]. 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. [33] 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 A_i 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 [7, 16, 21] 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 [7], 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}. Described in [14], 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 [11] 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

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

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 [27, 28], we introduced our new 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.

Clustering

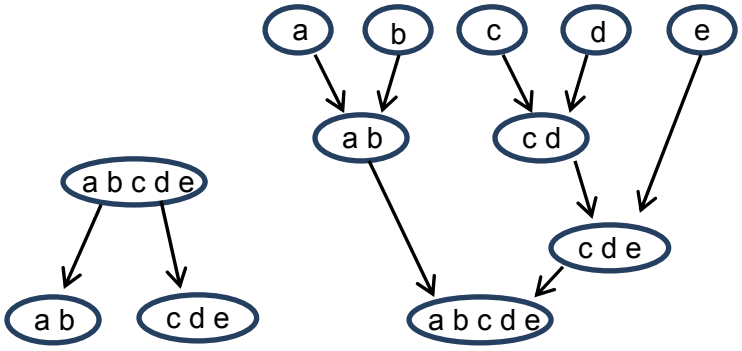
Clustering is a methodology that aggregates similar elements into groups, such that the elements in one cluster are more similar than elements in another cluster [1]. In the research community, clustering is used for data-mining, pattern classification and document retrieval. As shown in Figure 10, the typical activities involved in clustering are creating a pattern representation, defining a measurement that defines the distance between patterns, clustering/grouping of the patterns, potential abstracting the data and assessing the output of the clustering. The pattern representation defines the number of classes or features that are available to the clustering algorithm, while the distance measurement used to determine clusters during the execution of the algorithm can be as simple as a Euclidean distance, or a more sophisticated measurement as used in [12].

Grouping of elements can be “hard” or “fuzzy”, where hard clustering allows a pattern to only be in one cluster and fuzzy clustering allows membership to multiple clusters. Abstracting the data offers the user or the algorithm the ability to represent the data in a simplistic format to improve the clustering process, and assessing the output of the clustering is a means to evaluate how successful the algorithm was creating groups of data from the initial input set of data.

Clustering Approaches

In [1], they discuss multiple techniques that have been used to cluster data. Agglomerative and divisive techniques equate to bottom-up versus top-down strategies, where the agglomerative starts with each pattern in a distinct cluster and then aggregates the clusters until some stopping criteria has been met. On the other hand, the divisive approach starts with all the patterns in one cluster and then divides that cluster into multiple clusters until a stopping criterion has been met. Figure 12 illustrates the differences between agglomerative and divisive clustering.

Figure 12. Agglomerative vs. Divisive Clustering



Finally, as described in the previous section, allocation of patterns of the data in clusters can be either “hard” or “fuzzy”, where hard clustering allows a pattern to only be in one cluster and fuzzy clustering allows membership to multiple clusters.

Related Work

Clustering has been used for many years in the arena of data mining to aggregate similar collections of data, but it has recently entered into the realm of data privacy protection through *k*-anonymity. In [17], they proposed a greedy *k*-member clustering algorithm that attempts to minimize information loss while maintaining data quality. The algorithm works by randomly selecting a record *r* that is used as the seed for the first cluster and then additional records are added to the cluster so that the information loss is minimized within the cluster until at least *k* records belong to the cluster. Once the cluster reaches *k* records, a new record *r* that is furthest from the previous cluster is selected to start a new cluster in order to repeat the same aggregation process. Any records that are left over after this process is completed will be assigned to the closest clusters. This algorithm is slow and is susceptible to outliers that can cause increased information loss. A similar approach was implemented by [24] to anonymize a dataset. Unlike the previous algorithm, they chose the seed record randomly and continued to add records to the cluster until a user defined information loss threshold was exceeded. Any clusters that contained less than *k* records were deleted. The deletion of records could lead to significant information loss, and determining the information loss threshold is difficult to accomplish a priori.

In order to improve utility and performance times from the previous described algorithms, [23] offered an efficient k-mean clustering approach that first creates all clusters at a time and sorting them based upon their quasi-identifiers. The number of clusters to create is determined by dividing the total number of records by the desired *k* value. After the total cluster number is established, the algorithm selects *p* records to use as seeds for the *p* clusters. For each of the records in the dataset, the record is assigned to the closest cluster and the center point of that cluster is then updated. If any cluster has more than *k* records, the excessive records that are most dissimilar to the

rest of the records in the cluster are reassigned to more appropriate clusters. This algorithm is again susceptible to outliers and does not address the *l*-diversity issue. In [18], they describe an algorithm that attempts to provide at least *k* records and $l \geq 2$ distinct sensitive attribute values in each cluster that have minimal intra-cluster distances via a two-step method. The first step establishes a set of clusters from the input dataset such that each cluster satisfies the *k*-anonymity requirement, and the second step is responsible for ensuring that each cluster contains at least $l \geq 2$ distinct sensitive attribute values. If the clusters from the first step provide the *l*-diversity criteria, the second step is not required. Unfortunately, no experimental results are presented to indicate if the algorithm provides effective protection for the sensitive attributes.

Unlike the preceding approaches, [6] introduce an algorithm that attempts to preserve data utility based upon utility requirements after a dataset has been anonymized. The utility requirements are implemented using utility constraints that specify the mapping of each item to a generalized item that can occur during the anonymization process. A data owner is responsible for creating the utility constraints based upon particular application requirements. The results from this work appear promising for constraining (bounding) the data during the automated generalization process, but it does not address inflection points within numerical attributes nor groupings of categorical attributes that are addressed by our proposed data constraint rules.

Methodology

In this section, we will describe the two algorithms, the utility metric, and the two datasets that were used during the experimental phase of our work.

DataSets

For this project, we utilized two datasets, the public Adult census data from the UC Irvine machine learning repository [29], 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 data is not available to for public download, and the Adult Census dataset is the gold standard for gauging anonymity techniques. The attributes used from the Adult dataset are Age, Education Number, Capital Gain and Salary, and the attributes used from the MCPHD dataset are Age, Sex, Education Number, Race and Cancer Status.

Utility Metric

We propose that the data curators should have more control 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 critical thresholds in the data that should be maintained through the anonymization process, so that the meaning of the data is well maintained. In [27, 28], a new utility measurement called the research value (RV) was introduced to encapsulate the utility of the each attribute with respect to the following conditions:

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

A data constraint rule (DCR) defines groupings of data or ranges of data that, if maintained, help 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 8. Possible Data Constraint Rules for the 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, the data constraint rules could define inflection points of ages that would be important for someone who is interested in mining the data. 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.

Table 9. Possible Data Constraint Rules for Numerical Attribute

Attribute	Constraint Rule	Importance value
Age	64-65	25
Education Number	12-13	20
Capital Gain	10000-10001	45

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, $\sum DR_0$. Each dataset has an inherit value even if all of the data constraint rules are broken during the merging of two clusters, so to resolve this, the data owner has the ability to add a base value to the data constraint ratio, so that it does not zero out a RV if all of the rules are broken. The weight calculation of each attribute can be determined by establishing a correlation matrix among 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:

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

where all the elements are defined to be the same as those for the numerical attributes, except for the $\frac{|A_k|}{|A_o|}$ 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 k^{th} 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, which include the base value as determined by the data owner, that exist at the k^{th} level is 50, while 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 k^{th} 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. 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 [10], 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 algorithms are proposed:

- Utility-driven clustering algorithm that maximizes the research value
- Suppression version of the utility-driven clustering algorithm that selects clusters based upon potentially suppressing values from the attributes contained in the cluster.

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. To ensure that the anonymized dataset accounts for this issue, all of the initial clusters contain tuples that are a result of executing a SQL "select" statement on the raw dataset using a "group by" on all of the attributes. For this paper, we will only present numerical attributes due to the fact that the categorical attributes in the MCPHD data have a specifically complicated hierarchical structure and the solution for resolving these MCPHD categorical attributes is not relevant to this paper.

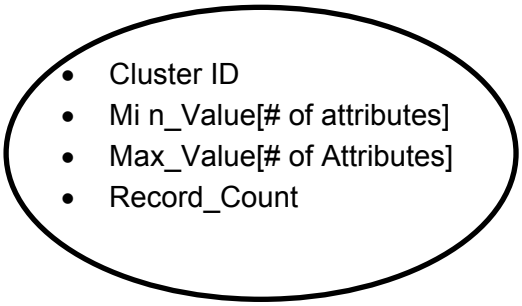
Utility-Driven Clustering (No Suppression)

The intent of this algorithm is to create an anonymized dataset that satisfies contains at least *k* records for all the attributes (both quasi-identifiers and sensitive identifiers) while maximizing the utility of the data at the same time. Unlike [27, 28], where all of the possible generalization groupings for each attribute along with the corresponding research values were pre-determined before running the global or local optimization algorithms, this automated utility-driven clustering algorithm starts with the raw data and then creates clusters until all clusters have at least *k* members. The first step in the algorithm is to execute the following SQL select statement to create thee initial group of clusters:

select $attr_1$, $attr_2$, $attr_3$, count(*) as Count from DB group by $attr_1$, $attr_2$, $attr_3$

where $attr_x$ is an attribute from the raw dataset, and the count is the number of tuples where the values for each of the attributes are equal. The results of the query are then stored into a cluster object that establishes a cluster ID, min and max values for each attribute and a count of the tuples that are represented in the cluster as shown in Figure 13. For the initial clusters, then min and max values are set to the same value for numerical attributes.

Figure 13. Cluster Object



Once the initial clusters have been established, the algorithm then identifies the clusters that contain less than *k* records (tuples) and stores these clusters into the under-K (UK) list. An entry from the UK list is then compared one by one to each of the clusters from the master list of clusters. A temporary RV is calculated during each comparison between the cluster from the UK list and a cluster from the master list. This temporary RV reflects the modified min and max values for all of the attributes if the two clusters were to be combined, the combined tuple count if the two clusters were to be merged, as well as the total count of the data constraint rules of the potential merger. After all the clusters from the master list are examined against the entry from the UK list, the merged cluster that resulted in the largest RV will be merged with the corresponding cluster from the master list by modifying the appropriate min and max values for all the attributes as well as update the tuple count for the merged cluster. The cluster from the UK list will then be deleted. The algorithm would then create a new UK list, which could contain the newly created node if its record count was below *k*, and repeat the

comparison process until all clusters contain at least *k* records. Figure 14 provides an overview of the utility driven clustering algorithm that does not use record suppression.

Figure 14. Utility-Driven Clustering (No Suppression)

Input: List *L*

Output: A *k*-anonymous Table *T*

Method:

```
1.  Init: Create the initial master cluster list (ML) and the under-k list (UK)
2.  While (true){
3.        Create the UK list
4.       If (Size(UK) > 0){
5.          Select cluster I from UK
6.          RV = 0
7.          Accepted_Cluster_ID = 0
8.          For each cluster c in ML select cluser_i{
9.              Calculate the New_RV_i of merged cluster c and I
10.             If New_RV_i greater than RV {
11.                 Set RV = New_RV_i
12.                 Set Accepted_Clister_ID = cluster_i.Cluster_ID
13.             }
14.             }
15.         Update cluster c_ID that equals Set Accepted_Clister_ID
16.         Remove cluster c from UK list
17.     }
18.     Else
19.     Break
20.  }
```

Utility-Driven Clustering (Suppression)

The utility-driven clustering with suppression is similar to the clustering method that does not use suppression with the exception of the following points:

- During the merger process, the algorithm calculates three separate RVs.
    - One RV that does not use suppression
    - One RV that suppresses one or more attribute values from the under K (UK) list
    - One RV that suppresses one or more attribute values from the master list
- The greatest RV of the three is then compared against the running RV from the other possible merged clusters
    - If it is greater than the running RV, then this potential cluster mergers' RV becomes the running RV, and this merger is marked as the best merger.
    - If it is not greater, then a new a new entry from the UK list is examined.

After all the clusters from the master list have been examined against the entry form the UK list, the one with the greatest RV becomes a new cluster in the master list, and the UK cluster is then recalculated to see if any clusters still have a record count under *k*.

Results and Discussion

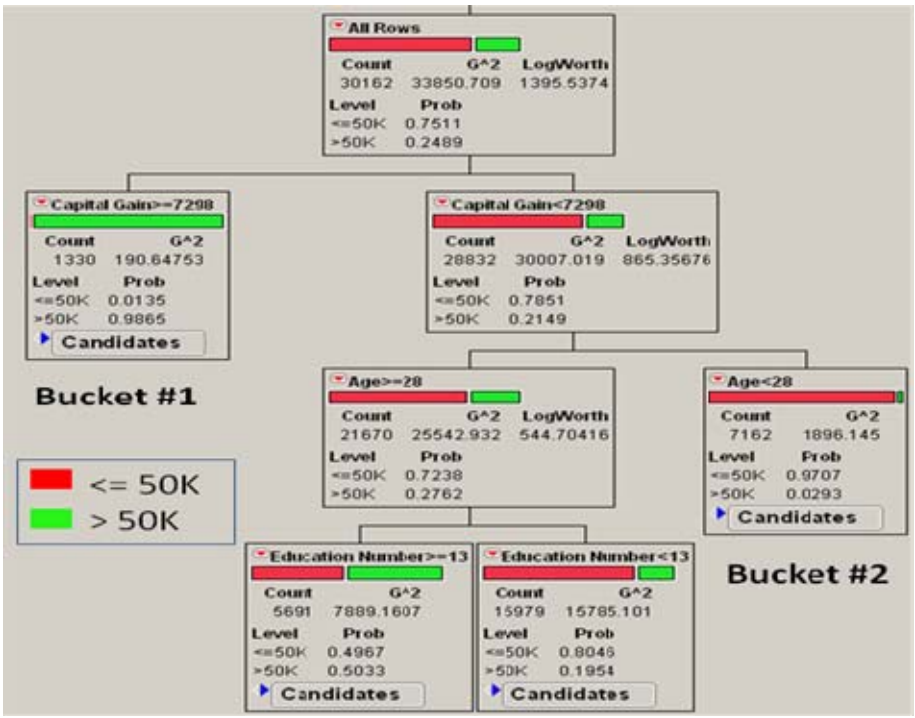
As a means to show how our algorithm performs against existing utility methodologies, our two proposed algorithms were run using the public Adult Census dataset and the proprietary death certificate dataset from the Marion County Public Health Department (MCPHD) of Indianapolis, Indiana using a range of *k* values (k=3, k=5 and k=10). The two proposed utility driven clustering algorithms described in the previous section along with the Bottom-Up algorithm [42] and a simple clustering algorithm using Euclidean distance were evaluated using the two datasets. To evaluate the effects of the anonymization process of each of the algorithms relative to the raw values of the two datasets, we used recursive partitioning (RP), which is a multivariate technique that finds the attributes that are able to differentiate a control parameter. The

results and discussion of the experiments of each of the algorithms are described in this section.

Adult Consensus Dataset

The control parameter in the Adult Consensus dataset is salary, which has two values: <=50K or >50K in yearly salary. Figure 15 shows the raw Adult Consensus dataset that has 30,162 records where 75% of the individuals had a yearly salary of <=50k and 25% of the individuals had a yearly salary of >50K.

Figure 15. Raw Adult Consensus Dataset Recursive Partitioning

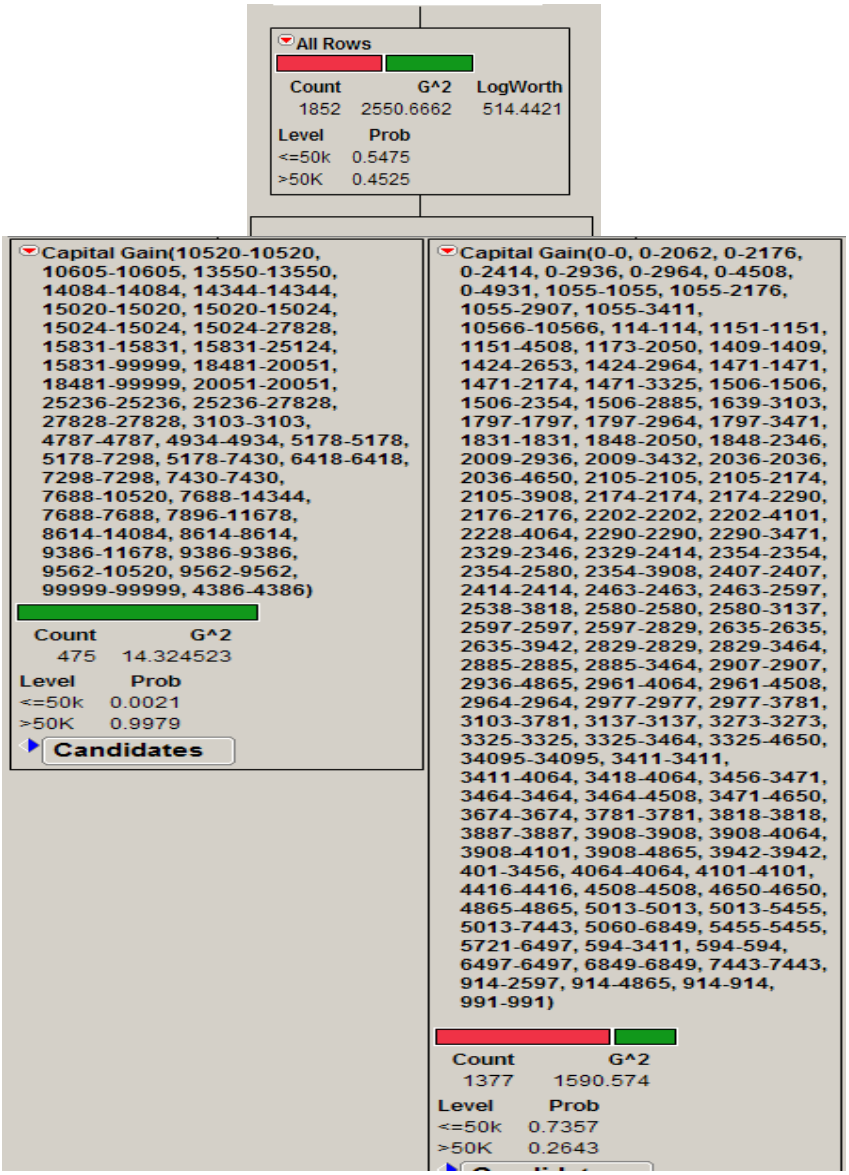


Out of the attributes contained in the Adult Consensus dataset, the three attributes that significantly differentiated the two salary groups in order of importance were Capital Gain, Age and Education. The first split point involved a Capital Gain value of $7298, where Capital Gain values >=$7298 produced a statistically significant leaf node that had 98% of 1330 individuals who had yearly salaries >50K. This leaf node is labeled as Bucket #1 for comparison purposes against the algorithms run during the experimental phase of the project. On the other side of the first partition, individuals who

had a yearly salary <=50K did not produce a node that was statistically significant from the parent node, so it was necessary to split this node to further refine the differentiation criteria for a salary. Splitting the right Capital Gain node again with the Age attribute value of 28 produced Bucket #2, which had 97% of the 7162 individuals that had a yearly salary of <=50K. The left node form the Age split did not produce a significant change, so a further split was required that utilized the education attribute to produce Bucket #3. Bucket #3, which is a coin toss in terms of classification of the salary attribute, can also be considered a 100% improvement form the original node that had only 25% of the individuals with a yearly salary of >50K, but the end users of the anonymized dataset would not find this 50/50 coin toss useful.

Each of the algorithms was then tested against the Adult Consensus dataset using *k* values of 3, 5 and 10. After the proposed automated utility driven clustering algorithm without suppression was executed using a *k* value of 3, the anonymized dataset was analyzed using recursive partitioning and it produced the dissection of clusters as shown in Figure 16. The 1852 count in the parent node represents the 30162 records in the Adult Census dataset. The left node shows a 99.8% probability of individuals in this node that have a salary >50K (Bucket #1), but there is some overlap with the right node in terms of Capital Gains. After filtering the overlap in ranges of Capital Gains between the right and left nodes, 144 individuals of the 1528 individuals who had Capital Gains exceeding $35K had a salary >50K. Partitioning beyond the Capital Gain node produced a cut point using Education, but there was complete overlap between the two nodes.

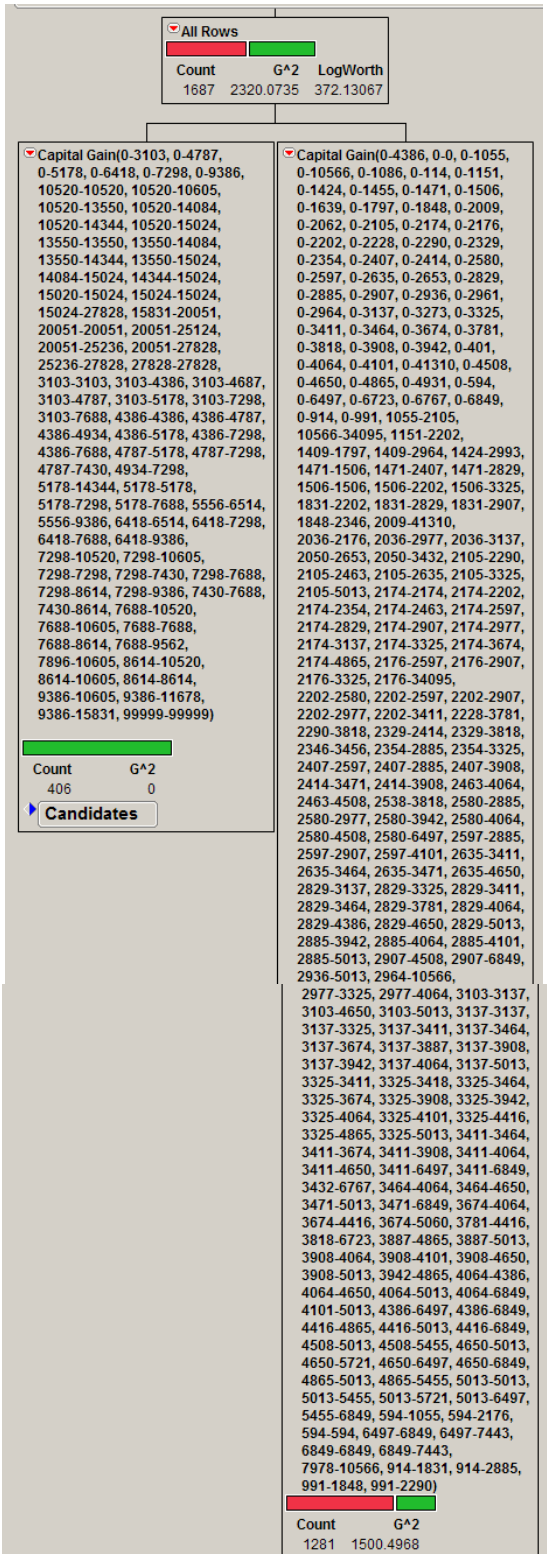
Figure 16. Utility Driven Clustering without Suppression, k =3



The Bottom-Up algorithm produced similar results to the proposed algorithm where 100% of the individuals in the node who had Capital Gains in the left node had a salary >50K as shown in Figure 17.

.

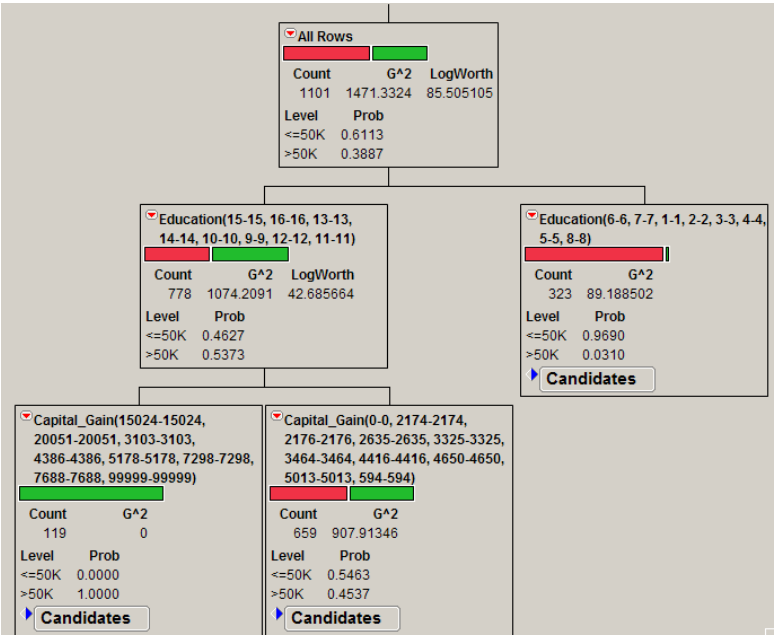
Figure 17. Bottom-Up Algorithm, k=3



It was again necessary to filter the overlap between the left and right nodes, resulting in 148 of the 1621 individuals in the left node with Capital Gains >$34K had a Salary >50K. Similar to the utility driven algorithm, the Bottom-up algorithm did not produce any further splits towards Bucket #2 that were plausible due to complete overlap in the Education attribute.

Interestingly, the results from the utility-driven with suppression for a *k* value of 3 produced a partition that first split using the Education attribute as shown in Figure 18. The bottom left node has 100% of the 1381 of the 1397 individuals who have an Education >9 and Capital Gains >$5178 have salaries >50k, and in the top right node, 3706 individuals with and Education less than 7 have salaries <=50K.

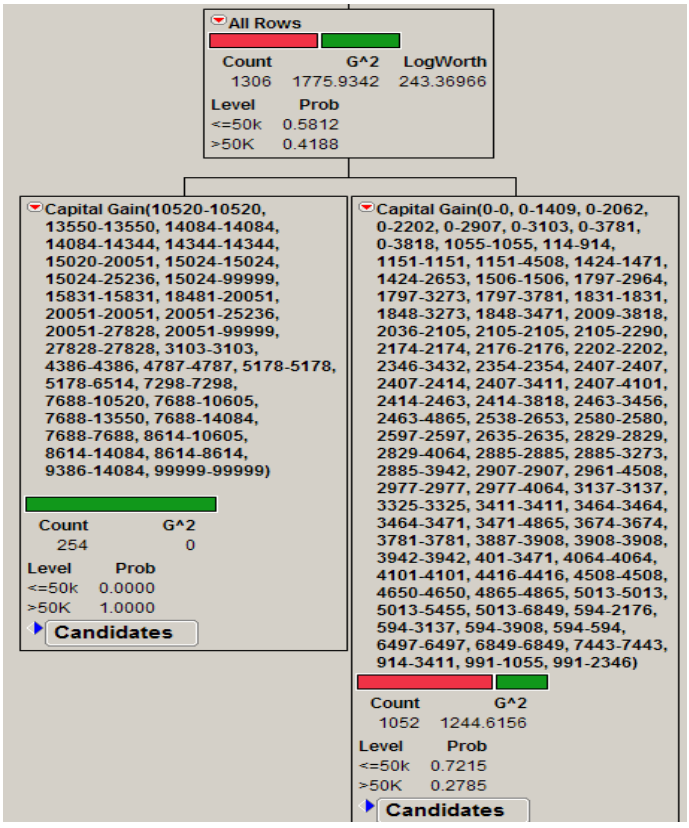
Figure 18. Utility-Driven Clustering with Suppression, k=3



The Euclidean distance clustering did not produce any partitions that did not involve complete overlap between the left and right split nodes for a *k* value of 3, 5 or 10. When using *k* values of 5 or 10, the Bottom-Up algorithm did not generate any partitions of significance, but the utility driven clustering algorithm did work for the remaining values of *k*.

When the utility driven clustering algorithm used a *k* value of 5, it created a first partition using Capital Gain for 100% of the individuals in the left node that had a salary >50k. As before using a *k* value of 3, it was necessary in this case to again filter the overlap between the right and left nodes, but it not to the degree as was seen when *k* was set to 3. Eliminating the overlap, 1019 of the 1440 individuals in the left node in Figure 19 who had Capital Gains >$7400 had a Salary >50K, which is significantly closer to the cutoff of Bucket #1 seen in the raw dataset in Figure 15. It should be noted that the utility driven clustering using a *k* value of 5 had far fewer clusters than the Bottom-Up and Euclidean, and even the utility driven clustering using a *k* value of 3.

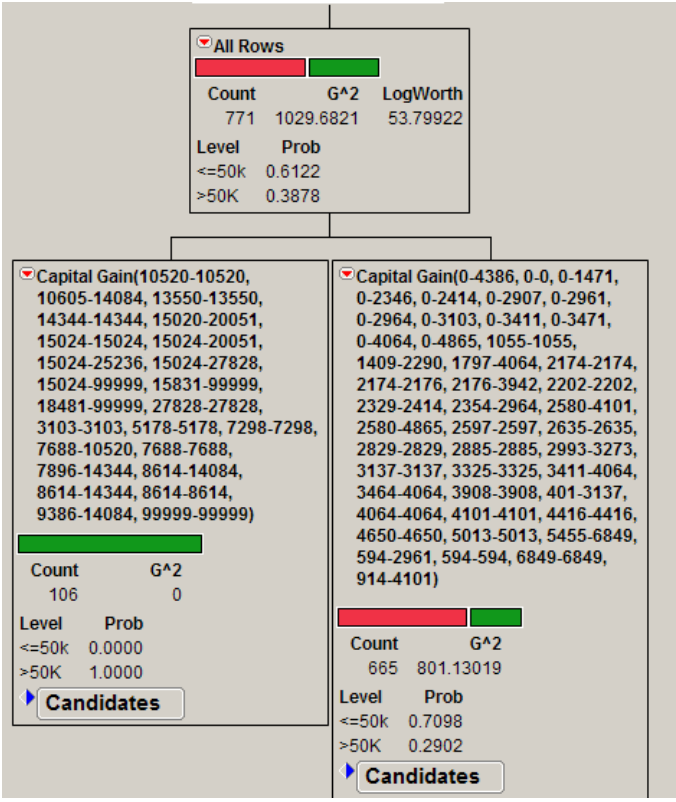
Figure 19. Utility Driven Clustering, k =5



Only the utility driven clustering algorithm produced any results for a *k* value of 10 with 100% of the individuals in the left node had a salary of >50K, but it did have some overlap with the right node although not as significant as the two previous *k* values

using the utility driven clustering. After the removal of the overlap, 1174 of the 1322 individuals in the left node with Capital Gains >$7444 had a salary >50k. This iteration still did not produce any results for the Bucket #2 as shown in Figure 20.

Figure 20. Utility Driven Clustering, k=10

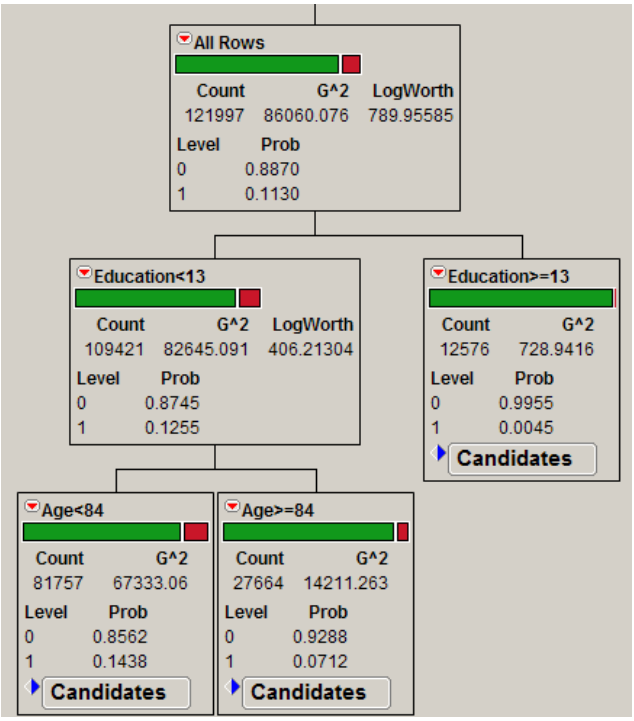


Overall, the proposed utility driven clustering algorithm that did not use record suppression outperformed both the Euclidean Distance clustering and the Bottom-Up algorithm. Only the new utility driven clustering algorithm without suppression was able to produce a result for Bucket #1 for *k* values of 5 and 10, and none of the algorithms were able to generate a solution for Bucket #2 or Bucket #3. The number of clusters generated by the utility driven clustering algorithm was much fewer than the Bottom-Up and Euclidean Distance clustering algorithms, which led to the improved partitioning results compared with the other two algorithms.

Marion County Public Health Department (MCPHD)

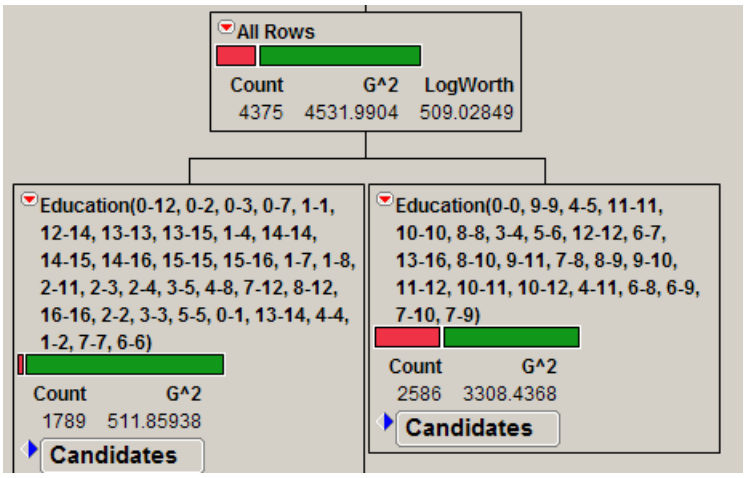
The control parameter for the Marion County Public Health Department is the cancer, which has been binned into two categories: non-cancer and cancer. Figure 21 shows the breakdown for the MCPHD dataset, which is comprised of ~122,000 records where 89% of the records contain non-cancer (0) entries compared to 11% cancer (1) entries in the root node. The attributes from this dataset that were examined were the following: Cancer status, Age, Sex, Education and Race. Unlike the Adult Census dataset, the MCPHD is a much more skewed dataset, and this is very apparent from the recursive partitioning splits as shown in Figure 21 where even after two splits on the left side of the tree, the distribution is not much different than the root node, but the split on Education produced a node where 99% of the individuals in the node had a diagnosis that was not cancer related. The next split was Age, but it did not differentiate significantly from the previous cut, and further attempts to split the data did not generate any noteworthy nodes.

Figure 21. Marion County Public Health Department



Each of the algorithms was run against the MCPHD to determine which algorithm which one would perform the best against the dataset. Unfortunately, none of the four algorithms (Utility driven with and without suppression, Bottom-Up and Euclidean distance) could produce any substantial outcomes anonymizing the MCPHD dataset. Figure 22 shows an example of the partitioning results, which in

Figure 22. Utility Driven Clustering Algorithm



this particular case are from the utility driven clustering algorithm. As you can see from the figure, there is complete overlap between the two Education splits, so conclusions can be drawn to separate the cancer status categories.

Summary and Future Directions

Two new utility driven clustering algorithms were presented to aid in the anonymization process while still maximizing the utility of the data, so that researchers can still find trends and patterns in the data. Central to the new algorithms is a new utility function that builds upon previous utility functions, but extends them by adding the concept of data constraint rules that allow the data owner to define inflection points in the dataset that are based upon well-defined attribute groupings for continuous parameters, and for collections of values for a categorical attribute. The new utility driven clustering algorithm was compared against existing algorithms, and it outperformed those algorithms using the gold standard dataset (Adult Consensus). Unfortunately, neither the new algorithms nor the existing algorithms were able to produce a plausible solution for the Marion County Public Health Department dataset, which can be attributed to the skewed nature of that dataset for the cancer status end point. Future directions for this project would be to add an automated means to provide weighting for each of the attributes in the dataset that can improve the results from the partitioning process. In addition, further improvements to the utility function in terms of an importance factor which can be set between zero and one that the data owner can set that will reflect the granularity portion of the utility function versus the value of the data constraint rules. This allows the data owner to put more emphasis on portions of the utility function based upon the needs of the users of the data.

DISCUSSION

Introduction

In the previous two chapters, four novel algorithms that employ a new user-driven utility function have been presented. The results of the experiments that were performed using each of the four novel algorithms to evaluate the effectiveness of the algorithms as compared to existing algorithms were also presented in the preceding chapters. The purpose of this chapter is to summarize those overall results, as well as identify and discuss the limitations of each of the four algorithms. Finally, future work will be discussed.

Results

In this section, the results of the experimental runs of the proposed algorithms will be presented and discussed in terms of their performance relative to the existing methodologies using the standard reference anonymity database, Adult Census [29].

Utility Functions

Listed in Table 10 are the utility functions that were used during the experimental phase of this project, where the intent of the experiment was to maximize the utility of the Adult Census database while still providing privacy protection for the individuals in that database. The Euclidean Distance is simply the sum of the ranges (max minus min) of each attribute when two clusters are being merged. During the anonymization process, the two clusters that produce the smallest Euclidean Distance would be merged.

Table 10. Utility Functions

Name	Euclidean Distance	Normalized Certainty Penalty (NCP)	Utility-Driven RV (UDRV)
Formula	$Euc(t) = \sum_{i=1}^{n} z_i - y_i$	$NCP(t) = \sum_{i=1}^{n} w_i * \frac{z_i - y_i}{\lvert A_i \rvert}$	$RV(t) = \sum_{i=1}^{n} w_i * \frac{\sum_{i=0}^{n} R_i^0 * N_{R_i}^0}{\sum_{i=0}^{k} R_i^k * N_{R_i}^k} * \frac{\sum DR_k}{\sum DR_0}$

The Normalized Certainty Penalty (NCP) [42] formula was implemented in the Bottom-Up clustering algorithm. Compared to the Euclidean Distance utility function, the NCP adds two new features to the utility function realm: an attribute weight and the overall range of an attribute in the dataset. The NCP is the sum of all attributes where each attribute's contribution is the range of an attribute in a cluster divided by the maximum range for that attribute times the weight factor of the attribute. A weight factor is defined to be the importance of that attribute relative to the other attributes in the dataset. The weight factor can be determined using correlation analysis and/or feature selection, as well as input from the data expert.

Finally, the Utility-Driven Research Value (UDRV), which was used by the optimization algorithms and the proposed automated clustering algorithms, adds the following features to the its utility function: data constraint rules; the sum of the ratio of number of elements in a cluster times the range of values at the most specific generalization level divided by the number of elements in the cluster times the range of values at generalization level *k*. The ratio in the utility-driven research value extends the ratio from the NCP. By adding the number of elements into the ratio portion of the utility function, it provides a means to weight the range of the values at a particular generalization level. As the range of value increases and the number of elements in that range increases, it decreases the utility of that generalized dataset.

Table 11 provides an overview of each algorithm and the associated utility function features, as well as the question of whether an algorithm has the ability to scale to datasets that contain a large number of attributes. Out of all the algorithms, the Euclidean Distance Clustering algorithm had the least number of the listed features, which makes sense since it was designed to be a simple distance solution. As mentioned in the previous paragraph, the NCP utility function added the ability to weigh each attribute to reflect its importance in terms of that particular

Table 11. Utility Function Features

Algorithm Name	Range of Attribute Value	Weight of Attribute	Number of Records in Range	Data Constraint Rules	Scalability to Larger Datasets
Euclidean Distance Clustering	Yes				
Bottom-Up Clustering using NCP	Yes	Yes			
Global Optimization using UDRV	Yes	Yes	Yes	Yes	Yes*
Local Optimization using UDRV	Yes	Yes	Yes	Yes	Yes[&]
Clustering using UDRV (No Suppression)	Yes	Yes	Yes	Yes	
Clustering using UDRV (Suppression)	Yes	Yes	Yes	Yes	

* Using distributed approach, correlation analysis (Feature selection) to reduce # of attributes
[&] Only examines a subset of all possible generalization strings

attribute relative to the other attributes in the dataset. How this weighting is determined can be fashioned by a multitude of available techniques, such as recursive partitioning or feature ranking.

The Global Optimization using UDRV covered the full span of the features including the number of records in a range, data constraint rules and scalability to larger datasets. In order to satisfy the scalability feature, the Global Optimization was modified to allow a distributed approach were the attributes are divided into sub-groupings and then the successful sub-grouping generalization strings are combined and analyzed to ensure k-anonymity. This was necessary, because the Global Optimization algorithm generates all the possible generalization strings for all the attributes before it performs the k-anonymity checks, which for datasets where the number of attributes starts to exceed twenty-five attributes the number of total possible strings will exceed 2^{25} combinations when there are just two possible values for each attribute. For complex datasets that contain a large number of attributes with multiple values per attribute like the Marion County Public Health Department's (MCPHD) Death Certificate dataset (See Table 2 in the first chapter), the number of possible combinations could become overwhelming. To address this escalating number of generalization strings, a correlation analysis of the attributes can be performed in order to eliminate attributes that are highly correlated. Removal of the all of the correlated attributes from a group except for one attribute will not have an adverse effect on the resultant anonymized database.

The Local Optimization using UDRV also contained all of the features listed in Table 11, including the ability to scale to larger datasets, which is accomplished because it does not pre-calculate all the possible generalization strings, but instead only examines a subset of all possible generalization strings. During the testing phase of the optimization algorithms on the MCPHD Death Certificate dataset, the Global Optimization was able to find more successful generalization strings as the number of tested attributes in the dataset increased. It should be noted that the Local Optimization algorithm was able to find the same solution as the Global Optimization algorithm for the twenty-four attribute test set from the MCPHD Death Certificate dataset when using a k

value of 3. Finally, as can be seen in Table 11, the remaining proposed algorithms (Clustering using UDRV with no suppression and Clustering using UDRV with suppression) each contained all of the features except for the ability to scale to larger datasets.

Determining the Proper k-Value

Although it is out of the scope of this work to find an ideal k value for a given dataset, a set of simulations were run to see the effect of increasing the number of attributes in dataset and the chances that the dataset could be anonymized to a particular k value given known number of records in the dataset. This analysis provides some insight into whether it is worth trying to anonymize a particular dataset. In these simulations, the possible values for the attributes have been limited to only two possibilities, so one could imagine the complexity of the problem when a specific attribute or a group of attributes have more than fifty possible values, where the values could occur with different frequencies. As shown in the results from the first chapter for the Marion County Public Health Department's Death Certificate dataset, as the number of attributes increased beyond twenty-four attributes that had many more than two possible values for each of those attributes, neither of the proposed optimization algorithms were able to produce an anonymized dataset that satisfied k-anonymity.

For these simulation tests, the number of attributes tested ranged from three to nine attributes along with corresponding k values of three, five and ten, as well as a simulation using twenty attributes with a k value of 5. Each attribute in the simulation had two possible values: zero or one, which were randomly generated with a 50:50 chance of occurring. All of the attribute values were then combined and a grouping operation was performed to aggregate similar records. These groupings were then tested to see if those aggregations contained at least k entries. Table 12 contains the results of the simulations and the criteria used for each for simulation. As you can see from the table

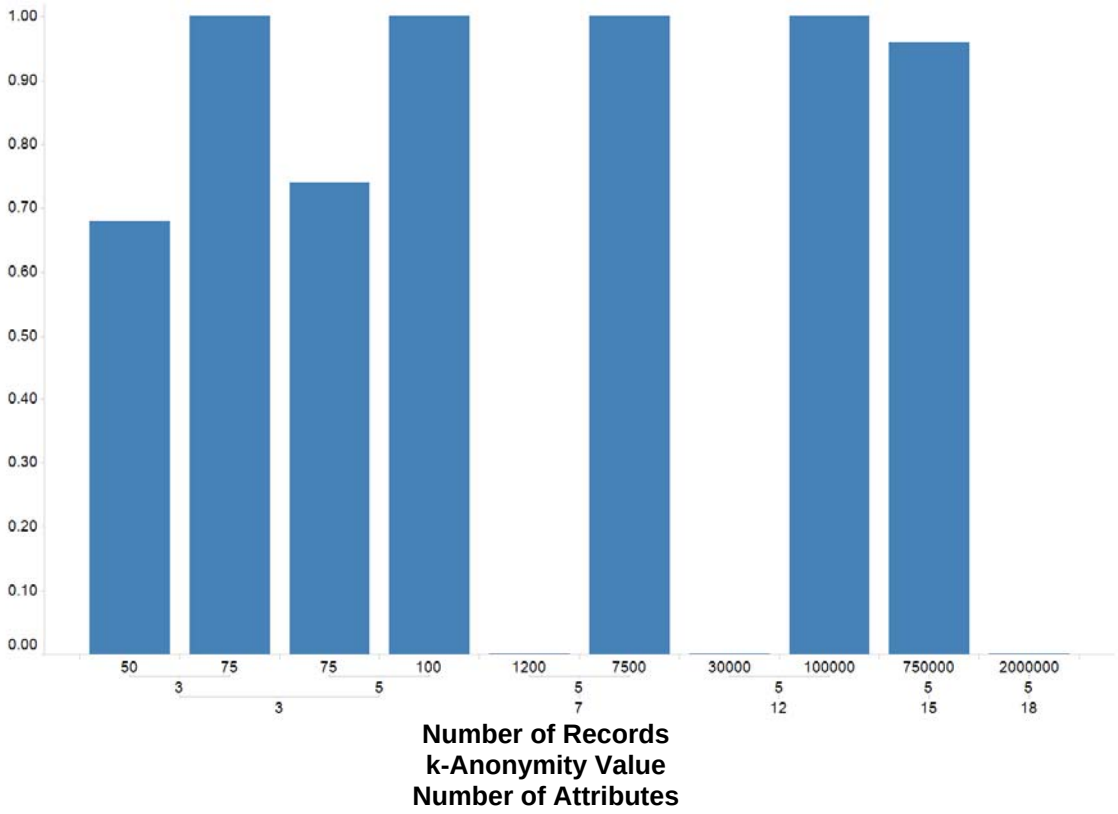
and figure, when there are three attributes in a dataset, the number of required records to satisfy k-anonymity for k values of three or five were at or under 100 records, When the attribute count jumped to seven along with a k value of five, the number of records required to satisfy k-anonymity jumps dramatically to around 7500 records.

Table 12. k Value Simulation

Number of Attributes/ K Value/ Number of Records/ Number of Repetitions	**Success Rate of Achieving k-Anonymity**
3/3/50/50	68%
3/3/75/50	100%
3/5/75/50	74%
3/5/100/50	100%
7/5/1200/500	0%
7/5/7500/500	100%
12/5/30000/500	0%
12/5/100000/500	100%
15/5/750000/200	96%
18/5/2000000/200	0%

For an attribute count of twelve and a k value of five, the record count required to satisfy k-anonymity for each of the five hundred repetitions approaches at least 100,000 records. When the number of attributes was increased to eighteen with a k value of five, even with two million records in the dataset, there were no successes for k-anonymity during any of the two hundred repetitions. Figure 23 graphically shows the increase of records needed to satisfy k-anonymity as the number of attributes increases.

Figure 23. K-Value Simulation Results



This analysis shows that it worthwhile to simulate a possible dataset before performing a k-anonymization process, and it can also aid in understanding why a particular full dataset with greater than thirty-six attributes like the Marion County Public Health Department's Death Certificate cannot be anonymized without significant pre-work. This brings to mind that it will be necessary to work with the data owner to identify attributes in the dataset that will have impact for researchers, as well as aid in the identification of outliers in the numerous attributes that could be removed before the anonymization process begins to increase the chances that an anonymized dataset can be generated.

Experimental Summary

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=5 and k=10). To evaluate the effects of the anonymization process of each of the algorithms relative to the raw values of the dataset, we used recursive partitioning (RP), which is a multivariate technique that finds the attributes that are able to differentiate a control parameter. Although we did not exhaustively test our proposed algorithms performance against the available public datasets, the Adult Census dataset is considered the standard dataset for testing new anonymity algorithms. In Table 13, the results of the proposed algorithms and existing anonymity algorithms are listed. The entries in the table indicate the number of individuals that were present in a partition node, the features or attributes from the dataset that partitioned the data, and the split percentage of the sensitive attribute (Salary: >50K or <=50K) in that node. The results are listed Bucket #1 represents those individuals who have a Salary >50K, while Bucket #2 represents those who have a Salary <=50K.

Table 13. Optimization Algorithms Performance using k=5

	Adult Census Raw Dataset	**Global Optimization using UDRV**	**Local Optimization using UDRV**	**Bottom-Up Clustering using NCP**
Bucket #1: Individuals/ Feature(s)/ Split Percentage	1330/ Capital Gain >=$7300/ 99%	1387/ Capital Gain >=$6000/ 95%	NA	NA
Bucket #2: Individuals/ Feature(s) / Split Percentage	7162/ Capital Gain <= $7300 and Age < 28/ 97%	NA	NA	18686/ Education is Pre-College/ 86%

Examining the information presented in Table 13, only the Global Optimization algorithm was able to produce a generalized solution for Bucket #1 that mirrored the recursive partitioning features that were seen in Bucket #1 from the raw dataset. The Bottom-Up algorithm found a solution for Bucket #2, but the full range of Capital Gain values as well as using the Education attribute did not align with the raw Adult Census dataset for Bucket #2.

Table 14. Optimization Algorithms Performance using k=10

	Adult Census Raw Dataset	**Global Optimization using UDRV**	**Local Optimization using UDRV**	**Bottom-Up Clustering using NCP**
Bucket #1: Individuals/ Feature(s)/ Split Probability	1330/ Capital Gain >=$7300/ 99%	NA	NA	NA
Bucket #2: Individuals/ Feature(s) / Split Percentage	7162/ Capital Gain <= $7300 and Age < 28/ 97%	NA	NA	20000/ Capital Gain <$7000/ 86%

In Table 14, the Bottom-Up algorithm was the only method to find a solution when the k value was increased to 10, but the amount of individuals was much higher than was seen in the raw Adult Census dataset. The remaining tables (Table 15,Table 16 and Table 17) show the comparisons between the Clustering using UDRV with and without suppression against the Bottom-Up clustering and Euclidean distance. Overall, the proposed utility driven clustering algorithm that did not use record suppression outperformed both the Euclidean Distance clustering and the Bottom-Up algorithm. Only the new utility driven clustering algorithm without suppression was able to produce a result for Bucket #1 for *k* values of 5 and 10, and none of the algorithms were able to generate a solution for Bucket #2 or Bucket #3. The number of clusters generated by the utility driven clustering algorithm was much fewer than the Bottom-Up and Euclidean

Distance clustering algorithms, which led to the improved partitioning results compared with the other two algorithms.

Table 15. Proposed Clustering Algorithm Performance, using k=3

	Adult Census Raw Dataset	**Clustering using UDRV (No Suppression)**	**Clustering using UDRV (Suppression)**	**Bottom-Up Clustering using NCP**	**Euclidean Distance**
Bucket #1: Individuals/ Feature(s)/ Split Percentage	1330/ Capital Gain >=$7300/ 99%	144/ Capital Gain >$35K/ 100%	1381/ Capital Gains > $5178 and Education >9/ 100%	148/ Capital Gain >$35K 100%	NA
Bucket #2: Individuals/ Feature(s) / Split Percentage	7162/ Capital Gain <= $7300 and Age < 28/ 97%	NA	3706/ Education <7 97%	NA	NA

Table 16. Proposed Clustering Algorithm Performance, using k=5

	Adult Census Raw Dataset	**Clustering using UDRV (No Suppression)**	**Clustering using UDRV (Suppression)**	**Bottom-Up Clustering using NCP**	**Euclidean Distance**
Bucket #1: Individuals/ Feature(s)/ Split Percentage	1330/ Capital Gain >=$7300/ 99%	1019/ Capital Gain >$7400/ 100%	NA	NA	NA
Bucket #2: Individuals/ Feature(s) / Split Percentage	7162/ Capital Gain <= $7300 and Age < 28/ 97%	NA	NA	NA	NA

Table 17. Proposed Clustering Algorithm Performance, using k=10

	Adult Census Raw Dataset	**Clustering using UDRV (No Suppression)**	**Clustering using UDRV (Suppression)**	**Bottom-Up Clustering using NCP**	**Euclidean Distance**
Bucket #1: Individuals/ Feature(s)/ Split Percentage	1330/ Capital Gain >=$7300/ 99%	1174/ Capital Gain >$7444/ 100%	NA	NA	NA
Bucket #2: Individuals/ Feature(s) / Split Percentage	7162/ Capital Gain <= $7300 and Age < 28/ 97%	NA	NA	NA	NA

Algorithm Limitations and Future Work

In this section, an analysis of the limitations of the four proposed algorithms is presented. The following list details the limitations of each algorithm, and a short discussion on future work to resolve these limitations will be discussed.

Optimization Algorithms

- Global Optimization using UDRV
  - All generalization levels of an attribute must be pre-defined
  - RVs pre-calculated for each level of generalization of an attribute
  - Computationally intensive as number of attributes increases
  - Does not guarantee *t*-closeness or *l*-diversity
- Local Optimization using UDRV
  - All generalization levels of an attribute must be pre-defined
  - RVs pre-calculated for each level of generalization of an attribute
  - Not guaranteed to find the best generalization string
  - Does not guarantee *t*-closeness or *l*-diversity

Both the Global and Local Optimization algorithms share similar limitations. In particular, each of these algorithms require pre-calculating the utility-driven research value before the algorithms are executed, as well as defining the generalization levels of those attributes. As stated in the previous sections, the Global Optimization algorithm generates all the possible generalization strings, which can become cumbersome as the number of attributes increases. Pre-pruning the generalizations levels within an attribute that do not pass the k-anonymity test aids in reducing the total number of generalization strings that need to be analyzed by the Global Optimization algorithm. Pre-pruning also helps the Local Optimization algorithm reduce unnecessary testing of generalization strings that will not pass the k-anonymity test, but this impact is seen more in the time to complete the anonymization. Although we do not divide attributes into quasi-identifiers and sensitive identifiers, our optimization algorithms do not guarantee that the final anonymized dataset complies with either *t*-closeness or *l*-diversity. This is an item that could be addressed in future work.

Automated Clustering Algorithms

- Clustering using UDRV (No Suppression)
    - Does not guarantee *t*-closeness or *l*-diversity
    - Only examined using continuous data
    - Categorical data will require pre-defined legal groupings
    - Clusters contain overlapping numerical ranges
- Clustering using UDRV (Suppression)
    - Does not guarantee *t*-closeness or *l*-diversity
    - Only examined using continuous data
    - Categorical data will require pre-defined legal groupings
    - Clusters contain overlapping numerical ranges

The clustering algorithms were developed to address the pre-defining generalization issue from the optimization algorithms that requires a substantial amount of manual work before the optimization algorithms could be executed. Again, although we do not discriminate attributes into quasi-identifiers and sensitive identifiers, our optimization and clustering algorithms do not guarantee that the final anonymized dataset complies with either *t*-closeness or *l*-diversity. For this project, only continuous data was examined during the experimentation phase of the project. Categorical data introduces complexity in terms of our data constraint rules, ie which elements from the dataset should be aggregated together and which elements should not be aggregated together to maintain utility in the anonymized dataset.

Using the clinical guidelines from government organizations such as the FDA or the EMA, the established grouping of categorical medical data could be accomplished in a reasonable manner, so that the data expert can define a well-defined set of data constraint rules that comply with agency requirements. Finally, during the anonymization process, overlapping ranges of an attribute were not prevented. From the research perspective, if there is extensive overlapping of ranges between the clusters for a particular attribute, then the utility of the dataset decreases dramatically. For example, for the Age attribute, if one cluster has a range of 1-10 and another cluster has a range of 9-23, then the researcher will not be able to differentiate between children, teens or young adults. The prevention of creating a new cluster where the range of an attribute does not span boundaries of any existing ranges of the attribute in other clusters is very computationally intensive, and should be considered for future work.

One final item that is often discussed in the area of data privacy is the need to add new data to a dataset and ensure that the release of this updated data does not expose previous releases of the data to violations of k-anonymity. Some may try to use a process to control the use of updated versions of the anonymized dataset, but with the

massive release of some datasets on the Internet, that does not seem possible. As an area of future work, releasing update versions of a dataset might only contain data that is more general than the previous release, to prevent cross-examination of the data to find a particular individual.

REFERENCES

1. A.K. Jain, M.N.M.a.P.J.F., *Data Clustering: A Review.* ACM Computing Surveys, 1999. **31**(3): p. 264-323.
2. Aggarwal, G. *On k-anonymity and the curse of dimensionality*. in *31st international conference on very large data bases*. 2005. Norway.
3. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigraphy, R., Thomas, D., Zhu, A., *Approximation Algorithms for k-Anonymity.* Journal of Privacy Technology, 2005.
4. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Thomas, D., Zhu, A. *Anonymizing Tables*. in *10th International Conference on Database Theory*. 2005.
5. Alina Campan, T.M.T., Nicholas Cooper, *P-Sensitive K-Anonymity with Generalization Constraints.* Transactions on Data Privacy, 2010. **3**: p. 65-89.
6. Aris Gkoulalas-Divanis, G.L., *Utility-guided Clustering-based Transaction Data Anonymization.* Transactions on Data Privacy, 2012. **5**: p. 223-251.
7. Bayardo, R., Agrawal, R., *Data Privacy Through Optimal k-Anonymization*, in *Proceedings. 21st International Conference on Data Engineering*2005. p. 217-228.
8. Chaytor, R., *Utility-preserving k-anonymity*, in *Department of Computer Science*2006, Memorial University of Newfoundland. p. 82.
9. DataMart. *DataMart*. Available from: http://health.mil/mhscio/programs_products/jmis/dhss/products/cdm.aspx.
10. 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.
11. Evfimievski, A., *Randomization in privacy preserving data mining.* ACM SIGKDD Explorations, 2002. **4**(2): p. 43-48.
12. Gagan Aggrawal, T.F., Krishnaram Kenthapadi, Samir Khuller, Rina Panigrahy, Dilys Thomas, An Zhu. *Achieving Anonymity via Clustering*. in *PODS '06*. 2006. Chicago, IL.
13. Gionis, A., Mazza, A., Tassa, T. *k-Anonymization Revisted*. in *24th International Conference on Data Engineering* 2008. Cancun, Mexico: IEEE.
14. HIPAA. *Health Insurance Portability and Accountability Act*. 2002; Available from: http://www.hhs.gov/ocr/hipaa.
15. Hua, M., Pei, J., *A Survey of Utility-based Privacy-Preserving Data Transformation Methods*, in *Privacy-Preserving Data Mining*2008, SpringerLink. p. 207-237.
16. 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.
17. Ji-Won Byun, A.K., Elisa Bertino, and Ninghui Li. *Efficient k-anonymization using clustering techniques*. in *12th International Conference on Database Systems for Advanced Applications*. 2007.
18. Kabir, M.E., Wang, H., Bertino, E., and Chi, Y. *Systemic Clustering Method for l-diversity Model*. in *Twenty-First Australasian Database Conference*. 2010.
19. 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.

20. 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.
21. LeFevre, K., DeWitt, D., Ramakrishnan, R. *Mondrian Multidimensional K-Anonymity*. in *In IEEE International Conference on Data Engineering (ICDE)*. 2006.
22. 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.
23. Lin, J.L., and Wei, M.C. *An efficient clustering method for k-anonymization*. in *2008 international workshop on privacy and anonymity in information society*. 2008.
24. Loukides, G.a.S., J. *Capturing data usefulness and privacy protection in k-anonymisation*. in *ACM Symposium on Applied Computing*. 2007.
25. 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.
26. Meyerson, A., Williams, R., *On the complexity of optimal k-anonymity*, in *23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*2004.
27. Morton, S., Mahoui, M., Gibson, P. J. *Data anonymization using an improved utility measurement*. in *IHI*. 2012.
28. Morton, S., Mahoui, M., Gibson, P. J., Yechuri, S., *An Enhanced Utility-Driven Data Anonymization Method.* Transactions on Data Privacy, 2012. **Accepted**.
29. Repository, U.C.I.M.L.; Available from: http://www.ics.uci.edu/mLearn/mlrepository.html.
30. Samarati, P., *Protecting Respondents' Identities in Microdata Release.* J IEEE Transactions on Knowledge and Data Engineering, 2001. **13**(6): p. 1010-1027.
31. Samarati, P., Sweeney, L., *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*, 1998.
32. Samarati, P., Sweeney, L., *Protecting privacy when disclosing information:k-anonymity and its enforcement through generalization and suppression*, 1998, SRI Computer Science Laboratory.
33. 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.
34. Sweeney, L., *K-Anonymity: A model for protecting privacy.* International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, 2002. **10**(5): p. 557-570.
35. Terrovitis, M., N. Mamoulis, and P. Kalnis, *Local and global recoding methods for anonymizing set-valued data.* The VLDB Journal, 2011. **20**(1): p. 83-106.
36. Truta, T.M., Bind V. *Privacy Protection: P-Sensitive K-Anonymity Property*. in *Workshop on Privacy Data Management, with the ICDE*. 2006.
37. 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.
38. Willenborg, L., deWaal, T., *Elements of Statistical Disclosure Control*. Vol. 155. 2001: Springer. 261.
39. Willenborg, L., deWaal, T., *Elements of Statistical Disclosure Control*. Springer Verlag Lecture Notes in Statistics. Vol. 155. 2000: Springer.

40. Wortman, J., Adam, N., *Security-Control Methods for Statistical Databases: A Comparative Study.* ACM Computing Surveys, 1989. **21**(4): p. 515-556.
41. 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.
42. 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.

CURRICULUM VITAE

Stuart Michael Morton

Education:

Indiana University, Indianapolis, IN	PhD	Health Informatics	2012
Indiana University, Indianapolis, IN	MS	Bioinformatics	2006
Purdue University, Indianapolis, IN	MS	Computer Science	1997
DePauw University, Greencastle, IN	BA	Computer Science	1993

Work Experience:

Eli Lilly & Co, Indianapolis, IN 2006 - Current

Assoc Consultant Scientist-ADME

- Mine, assemble and review experimental bioassay data to independently build global in *silico* models, and support project teams to resolve ADME related issues.
- Collaborate with RO/PET Tracer group to build and utilize in *silico* tools to identify potential evaluation molecules for preclinical tracers.
- Work with ADME colleagues to develop new automated tools to increase productivity for their day to day responsibilities.
- Provide cADMET infrastructure support in order to provide a stable working environment for the team.

Covance Inc, Indianapolis, IN. 2005 – 2006

Java Developer

- Developed applications for Covance Central Labs to allow customers to create, maintain and retrieve results from clinical trials.
- Created tools to mine the clinical trial data to provide insight into the types of clinical trials that Covance was providing to customers.

Lucent Technologies, Naperville, IL 1996 - 2005

Java Developer

- Responsible for managing a team of seven developers who designed, implemented and tested Java code that was used to dynamically maintain data on a cutomer's cellular network
- Provided a database backup and restore mechanism that allowed customers to backup and restore their configuration data for disaster recovery.

Publications:

1. Morton, S. Mahoui, M. Gibson, P.J. An Automated Data Utility Clustering Methodology using Data Constraint Rules. To appear in SHB'12: International Workshop on Smart Health and Wellbeing.

2. Morton, S., Mahoui, M. Gibson, P.J. and Yechuri, S. An Enhanced Utility-Driven Data Anonymization Method. Transactions on Data Privacy 5:2 (2012) 469-503.

3. Zamek-Gliszczynski, M. J., Sprague, K. E., Espada, A., Raub, T. J., Morton, S. M., Manro, J. R. and Molina-Martin, M., How well do lipophilicity parameters, MEEKC microemulsion capacity factor, and plasma protein binding predict CNS tissue binding?. J. Pharm. Sci., 2012: 1932–1940.

4. Stuart Morton, Malika Mahoui, P. Joseph Gibson: Data anonymization using an improved utility measurement. IHI 2012: 429-436.

5. Joel P. Bercu, Stuart M. Morton, J. Thom Deahl, Vijay K. Gombar, Courtney M. Callis, Robert B. L. van Lier. In Silico Approaches to Predicting Cancer Potency for Risk Assessment of Genotoxic Impurities in Drug Substances. Regulatory Toxicology and Pharmacology (2010).

6. Morton, S., Bukhres, O., Mossman, M. Mobile Medical Database Approach for Battlefield Environments. Australian Computer Journal 30(2):87-95 (1998).

7. Morton S, Bukhres O, Vanderdijs E, Zhang P, Mossman M, Crawley C, Platt J. 1997. A proposed mobile architecture for a distributed database environment. Proceedings of the 5th Euromicro Workshop on Parallel and Distributed Processing.

8. S. Morton, O. Bukhres. “Mobile Transaction Management in Distributed Medical Databases” Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems (CBMS 1997).

9. Morton S, Bukhres O. Utilizing mobile computing in the wishard memorial hospital ambulatory service. Proceedings of the 12th ACM Symposium on Applied Computing (ACM SAC 1997).

10. Morton S, Bukhres O, Mossman M. 1996. Mobile computing architecture for a battlefield environment. Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications.

11. Morton S, Bukhres O. 1996. Mobile transaction recovery in distributed medical databases. Proceedings of the IASTED Eighth International Conference on Parallel and Distributed Computing and Systems.

Abstract / Presentations:

1. Morton, S., Joshi, E., Benesh, D., Barth, V., Raub, T. Building and Utilizing *in silico* Tools to Identify Potential Molecules for Evaluation as Preclinical Tracers: Facilitating Rapid Receptor Occupancy Assay Development. Neuroreceptor Mapping Meeting. Baltimore, MD. August 2012.

2. Morton S. Data Anonymization using an Improved Utility Measurement. IHI'12, January 28-30, 2012. Miami ,FL.

3. Mattioni BE, Morton SM, Staton BA, Sawada GA,Desai PV, Raub TJ. Application of *In Silico* Models and Physicochemical Property Calculations During Hit Assessment for CNS Targets, PSWC/AAPS Annual Meeting, New Orleans, LA. November 2010.

4. Morton S. 1996. Mobile transaction recovery in distributed medical databases. Paper presented at the IASTED Eighth International Conference on Parallel and Distributed Computing and Systems; November 1996. Chicago, IL