Probabilistic Anonymity

Sachin Lodha
Tata Consultancy Services
sachin.lodha@tcs.com

Dilys Thomas *
Stanford University
dilys@cs.stanford.edu

ABSTRACT

In this age of globalization, organizations need to publish their micro-data owing to legal directives or share it with business associates in order to remain competitive. This puts personal privacy at risk. To surmount this risk, attributes that clearly identify individuals, such as `Name`, `Social Security Number`, `Driving License Number`, are generally removed or replaced by random values. But this may not be enough because such de-identified databases can sometimes be joined with other public databases on attributes such as `Gender`, `Date of Birth`, and `Zipcode` to re-identify individuals who were supposed to remain anonymous. In literature, such an identity-leaking attribute combination is called as a quasi-identifier. It is always critical to be able to recognize quasi-identifiers and to apply to them appropriate protective measures to mitigate the identity disclosure risk posed by join attacks.

In this paper, we start out by providing the first formal characterization and a practical technique to identify quasi-identifiers. We show an interesting connection between whether a set of columns forms a quasi-identifier and the number of distinct values assumed by the combination of the columns. We then use this characterization to come up with a probabilistic notion of anonymity. Again we show an interesting connection between the number of distinct values taken by a combination of columns and the anonymity it can offer. This allows us to find an ideal amount of generalization or suppression to apply to different columns in order to achieve probabilistic anonymity. We work through many examples and show that our analysis can be used to make a published database conform to privacy acts like HIPAA. In order to achieve the probabilistic anonymity, we observe that one needs to solve multiple 1-dimensional *k*-anonymity problems. We propose many efficient and scalable algorithms for achieving 1-dimensional anonymity. Our algorithms are optimal in a sense that they minimally distort data and retain much of its utility.

1. INTRODUCTION

"Over a year and a half, one individual impersonated me to procure over $50,000 in goods and services. Not only did she damage my credit, but she escalated her crimes to a level that I never truly expected: she engaged in drug trafficking. The crime resulted in my erroneous arrest record, a warrant out for my arrest, and eventually, a prison record when she was booked under my name as an inmate in the Chicago Federal Prison." - An excerpt from the verbal testimony of Michelle Brown to a US Senate Committee [9].

Unfortunately, in today's highly networked digital world, incidents like the above with Michelle Brown are commonplace. According to Bureau of Justice Statistics Bulletin [6], 3.6 million households, representing 3% of the households in the United States, discovered that at least one member of the household had been the victim of identity theft during the previous 6 months in 2004. According to the same report, the estimated loss as a result of identity theft was about $ 3.2 billion. Needless to say that preventing identity thefts is one of the top priorities for government, corporations and society alike.

Globalization further complicates this picture. Due to legal directives or business associations, there are multiple scenarios where in organizations need to share or publish their micro-data to remain competitive. This puts personal privacy at further risk. To surmount this risk, attributes that clearly identify individuals, such as `Name`, `Social Security Number`, `Driving License Number`, are generally removed or replaced by random values. But this may not be enough because such de-identified databases can sometimes be joined with other public databases on seemingly innocuous attributes to re-identify individuals who were supposed to remain anonymous. For example, according to one study [33], approximately 87% of the population of the United States can be uniquely identified on the basis of `Gender`, `Date of Birth`, and 5-digit `Zipcode`. The uniqueness of such attribute combinations leads to a class of attacks where data is re-identified by joining multiple and often publicly available data-sets. This type of attack was illustrated by Sweeney in [33] where the author was able to join a public voter registration list and the de-identified patient data of Massachusetts' state employees to determine the medical history of the state's governor.

In literature, such an identity-leaking attribute combination is called as a *quasi-identifier*. It is always critical to be able to recognize quasi-identifiers and to apply to them appropriate protective measures to mitigate the identity disclosure risk posed by join attacks. In fact, Sweeney herself proposed a *k*-anonymity model in [31] for the same. According to her, a database table is said to be *k*-anonymous if for each row in the table there are $k-1$ other rows in the table that are *identical* along the quasi-identifier attributes. Clearly, a join with a *k*-anonymous table would give rise *k* or more matches and create confusion. Thus, an individual is hidden in a

*Supported in part by NSF Grant ITR-0331640

KDD 2007 August 12-15, 2007, San Jose, California

crowd of size k giving her k-anonymity. It also means that the identity disclosure risk is at most $1/k$ for "join" class of attacks.

Although such a simple and clear quantification of privacy risk makes k-anonymity model attractive, its widespread use in practice is severely hampered owing to the following factors:

1. Choice of k is not clear. From pure privacy point of view, larger k would mean more privacy, but it comes at the cost of utility [1]. What is the right choice of k for the given data and the given notion of utility has not been very well understood yet.

2. For k-anonymity model to be effective, it is critical that there is a complete understanding of the quasi-identifiers for the give data-set. But there is no real formalism available for deciding whether an attribute combination could form a quasi-identifier. This is currently done manually, based on folk-lore and human expertise.

3. For a given k, the goal is always to minimally suppress or generalize the data such that the resultant data-set is k-anonymous. However, for some natural notions of measuring this resultant distortion, the minimization problems turn out to be NP-Hard [26, 2, 4].

   On the approximation front, no efficient but good approximation algorithms are currently known. The known algorithms are either $\tilde{O}(k)$ approximations [26, 2] or super-linear [4] - thus making them inefficient or expensive.

1.1 Paper Organization and Contribution

In this paper, we start out by providing the first formal characterization and a practical technique to identify quasi-identifiers. In Section 2, we also show an interesting connection between whether a set of columns forms a quasi-identifier and the number of distinct values assumed by the combination of the columns.

We then use this characterization in Section 3 to come up with a probabilistic notion of anonymity. Again we show an interesting connection between the number of distinct values taken by a combination of columns and the anonymity it can offer. This allows us to find an ideal amount of generalization or suppression to apply to different columns in order to achieve probabilistic anonymity. We work through many examples and show that our analysis can be used to make a published database conform to privacy acts like HIPAA.

In order to achieve the probabilistic anonymity, we observe that one needs to solve multiple 1-dimensional k-anonymity problems. In Section 4, we propose many efficient and scalable algorithms for achieving 1-dimensional anonymity. Our algorithms are optimal in a sense that they minimally distort data and retain much of its utility. The algorithms provided are a stark contrast to previous NP-hard results and comparatively more complicated algorithms for the previous notion of anonymity called k-anonymity [33].

We then experimentally verify our algorithms on real life data sets in Section 5. We sketch the related work in Section 6 and finally conclude in Section 7.

2. AUTOMATIC DETECTION OF QUASI-IDENTIFIERS

DEFINITION 1. *A quasi-identifier set Q is a minimal set of attributes in table T that can be joined with external information to re-identify individual records (with sufficiently high probability).*

Above definition is from [29]. A similar definition can be found in an earlier paper of Dalenius [16]. As the reader can sense, this definition is informal since it does not make "external information" and "sufficiently high probability" explicit. Possibly because of this, we do not know any formal procedure or test for identifying quasi-identifiers. Almost always, researchers and practitioners assume that quasi-identifier attribute sets are known based on specific knowledge domain [23].

We present a more formal definition of quasi-identifier below. In our definition, we do not insist on minimality of attribute set as such although one could easily accommodate it if required. The external information is the *universal table* $\mathcal{U}$ having information about entire (relevant) population. It has n rows. Typically, $\mathcal{U}$ would mean census records that many countries make readily available [10].

DEFINITION 2. **α-quasi-identifier** *An α quasi-identifier is a set of attributes along which an α fraction of rows in the universe can be uniquely identified by values along the combination of these attribute columns.*

EXAMPLE 1. *Empirically it has been observed that* 87*% of the people in the U.S. can be uniquely identified by the combination of* `Gender`, `Date of Birth` *and* `Zipcode`. *Therefore (*`Gender`, `Date of Birth`, `Zipcode`*) forms a* 0.87*-quasi-identifier for the U.S. population. Note that the U.S. census table is our universal table $\mathcal{U}$ here.*

Ideally, given an α and $\mathcal{U}$, it is straight-forward to figure out whether some particular attribute combination forms an α-quasi-identifier in $\mathcal{U}$ by simply measuring the number of singletons in that attribute combination. One may even try an apriori like approach [5] and calculate all α-quasi-identifiers in $\mathcal{U}$. In practice, there are errors in $\mathcal{U}$ that come in during data collection phase itself [12, 11] and the knowledge about $\mathcal{U}$ is never exact. This would lead to erroneous conclusions about a quasi-identifier. Therefore, it does not justify the expensive calculations given above. In fact, one then prefers a quick and inexpensive approach that gives a good *estimate* of the same.

In what follows, we assume that the universal table $\mathcal{U}$ itself is not known. What we know is that it is a *random sample* built *with replacement* from a probability space. Thus our analysis is probabilistic. For the sake of analysis, we require that there is a probability distribution, but in reality, our final results are independent of this probability distribution. Moreover, we work only with the expectations since our goal is to give *good estimates* quickly. Since the sum of random variables is tightly concentrated around the expectation (by bounds like the Chernoff bounds [15]), our analysis and results are quite fair. We do not work out the Chernoff analysis though in order to keep our results and presentation simple.

We build our probability space on the distinct values that an attribute combination can take. Therefore, we need to know the number of distinct values for every attribute combination. Since one can get (or reasonably estimate) the count of distinct values for each attribute in $\mathcal{U}$ [17], we simplify our task with the following assumption.

DEFINITION 3. **Multiple Domain Assumption** *Let d_1, d_2, ..., d_k be the number of distinct values along columns C_1, C_2, ..., C_k respectively. Then, the total number of distinct values taken by the $(C_1, C_2, \ldots, C_k)$ column set is $D = d_1 \times d_2 \times \ldots d_k$.*

EXAMPLE 2. *We study the number of distinct values taken by the set of columns (*`Gender`, `Date of Birth`, `Zipcode`*). The number of distinct values of column* `Gender` *(C_1) is $d_1 = 2$. The*

number of distinct values of column `Date of Birth` *(C_2) can be approximated as* $d_2 = 60 * 365 \approx 2 * 10^4$.[1] *The number of distinct values along column* `Zipcode` *(C_3) is* $d_3 = 10^5$. *The number of distinct values of the column-set (*`Gender`, `Date of Birth`, `Zipcode`*) is* $D = d_1 \times d_2 \times d_3 = 2 * (2 * 10^4) * 10^5 = 4 * 10^9$.

As another example, consider the set of columns (`Nationality`, `Date of Birth`, `Occupation`*). The number of distinct values of column* `Nationality` *(C_1) is* $d_1 = 200$. *Once again, the number of distinct values of column* `Date of Birth` *(C_2) can be approximated as* $d_2 = 60 * 365 \approx 2 * 10^4$. *The number of distinct values of column* `Occupation` *(C_3) is roughly* $d_3 = 100$. *Thus* $D = d_1 \times d_2 \times d_3 = 200 * (2 * 10^4) * 100 = 4 * 10^8$.

Remark: Please note that it may be possible to consider correlations among various attributes and, therefore, arrive at a tighter estimate of D. Such analysis would certainly lead to improved bounds in what follows. Yet we decided not to incorporate correlations - partly because it would have made analysis very tough and main purport of our results could have easily been lost, but largely because we also wanted our results to be viable and useful. Reader will notice that larger estimate for D implies stricter privacy control and more anonymization in what follows. This is acceptable in practice as long as it is easily doable and does not lead to high loss in data utility.

Suppose that a set of columns take D different values with probabilities $p_1, p_2, \ldots, p_D$, where $\sum_{i=1}^{D} p_i = 1$. Let us first calculate the probability that the i^{th} element is a singleton in the universal table $\mathcal{U}$. It means first selecting one of the entries in the table (there are n choices), setting it to be this i^{th} element (which has probability p_i), and setting all other entries in the table to something else (which happens with probability $(1 - p_i)^{n-1}$). Thus, the probability of i^{th} element being a singleton in the universal table $\mathcal{U}$ is $np_i(1 - p_i)^{n-1}$.

Let X_i be the indicator variable representing whether i^{th} element is a singleton. Then, its expectation

$$E[X_i] = P[X_i = 1] = np_i(1 - p_i)^{n-1} \approx np_i e^{-np_i}.$$

Let $X = \sum_{i=1}^{D} X_i$ be the counter for the number of singletons. Now its expectation is given by

$$E[X] = \sum_{i=1}^{D} E[X_i] = \sum_{i=1}^{D} np_i e^{-np_i}.$$

Let us analyze which distribution maximizes this expected number of singletons. We aim to maximize $\sum_{i=1}^{D} x_i e^{-x_i}$, subject to $\sum_{i=1}^{D} x_i = n$ and $0 \le x_i, \forall 1 \le i \le D$.

THEOREM 1. *If $D \le n$, then the expected number of singletons is bounded above by $\frac{D}{e}$.*

PROOF: Please refer to the Appendix A for a detailed proof. □

THEOREM 2. *If $D \ge n$, then the expected number of singletons is bounded above by $ne^{\frac{-n}{D}}$.*

PROOF: Please refer to the Appendix A for a detailed proof. □

Figure 1 shows how the maximum expected fraction of singletons or unique rows in a collection of n rows behaves, as the number of distinct values, D, varies. The graph plots the maximum expected fraction of unique rows as a function of $\frac{D}{n}$. It is the line $\frac{D}{en}$ for $\frac{D}{n} \le 1$ according to Theorem 1. For $\frac{D}{n} \ge 1$, it is the curve $e^{\frac{-n}{D}}$ according to Theorem 2. The curve is both continuous and smooth (differentiable) at $\frac{D}{n} = 1$ with $f(1) = \frac{1}{e}$ and $f'(1) = \frac{1}{e}$.

[1]Throughout this paper we assume that the ages of people belonging to the database comes from an interval of size 60 years.

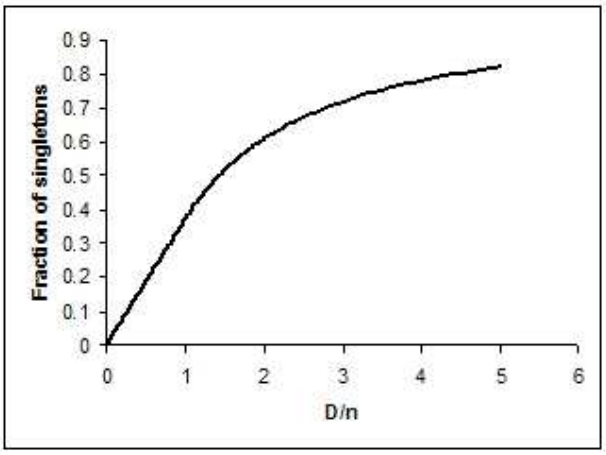

Figure 1: *Quasi-Identifier Test*

Figure 1 forms a ready reference table in order to test whether a set of attributes forms a probable quasi-identifier. For example, if for a set of attributes $D < 3n$, then it is unlikely that the set of attributes will form a 0.75 quasi-identifier. If a set of attributes do not form an α-quasi-identifier according to the the number of distinct values in Figure 1, then they almost certainly do not form an α-quasi-identifier as the plot gives the maximum expected fraction of singletons (as per Theorem 1 and Theorem 2).

EXAMPLE 3. *We now show how (*`Gender`, `Date of Birth`, `Zipcode`*) forms a quasi-identifier when restricted to the U.S. population. The size of the U.S. population can be approximated as* $3 * 10^8$, *that is, the size of the universal table n is* $3 * 10^8$. *The number of distinct values taken by the attribute set (*`Gender`, `Date of Birth`, `Zipcode`*) is* $4 * 10^9$ *from Example 2. Therefore, by Theorem 2, the maximum expected fraction of rows with singleton occurrence is* $e^{-3*10^8/4*10^9} = e^{-0.075} \approx 0.93$. *Thus, (*`Gender`, `Date of Birth`, `Zipcode`*) is a potential* 0.93 *quasi-identifier. Please recall that this combination is already known to be a* 0.87 *quasi-identifier [33].*

EXAMPLE 4. *We now give an example of a set of attributes that does not form a quasi-identifier. Let us consider (*`Nationality`, `Date of Birth`, `Occupation`*). The number of distinct values along these columns is given from Example 2 as* $D = 4 * 10^8$. *Here the size of the universal table is* $n = 6 * 10^9$, *that is, equal to the world population. Since $D < n$, we use Theorem 1 and find that the expected fraction of rows with singleton occurrence is bounded above by* $D/en = 4 * 10^8/2.7 * 6 * 10^9 \approx 0.025$. *Thus these columns almost certainly do not form even a* 0.05 *quasi-identifier as 0.025 is an upper bound on the expected fraction of singletons over all possible probability distributions over quasi-identifier values.*

We now provide a simple test to decide whether a combination of attributes forms a potentially dangerous quasi-identifier, that is, say $\alpha \ge 0.5$.

THEOREM 3. *Given a universe of size n, a set of attributes can form an α-quasi-identifier (where $0.5 \le \alpha < 1$) if the number of distinct values along the columns, $D > \frac{n}{ln(1/\alpha)}$.*

PROOF. Please refer to the Appendix A for a detailed proof. □

2.1 Distinct Values and Quasi-Identifiers

In this section, we have provided an interesting connection between whether a set of columns forms a quasi-identifier and the number of distinct values assumed by the combination of the columns. The main contributions of this association are as follows.

1. We provide a fast and efficient technique to test whether a set of columns forms a quasi-identifier. However there may be false positives. A set of columns signalled as a probable α quasi-identifier may only be a β quasi-identifier for some $\beta < \alpha$.

2. We do not assume anything about the distribution on the values taken by the quasi-identifier. The expected number of singletons is bounded by the expression provided in this section for all possible distributions over the values taken by the quasi-identifier.

3. When a set of columns is declared not to be a quasi-identifier by the test in this section, the set of columns is almost certainly not a quasi-identifier, that is, there is a minuscule chance of false negatives.

3. PROBABILISTIC ANONYMITY

In Sweeney's anonymity model [33], every row of the dataset is required to be identical with k other rows in the dataset along Q. In the following notion of anonymity, we insist that each row of the anonymized dataset should match with at least k or more rows of the universal table $\mathcal{U}$ along Q. Since $\mathcal{U}$ is represented in a probabilistic fashion, we want this event to happen with high probability.

DEFINITION 4. *A dataset is said to be probabilistically* $(1-\beta, k)$*- anonymized along a quasi-identifier set Q, if each row matches with at least k rows in the universal table* $\mathcal{U}$ *along Q with probability greater than* $(1-\beta)$*.*

Our notion of anonymity is similar to that of [33] for an adversary who is *oblivious*, that is, she is not really looking for some *particular* individuals, but is trying to do a join on Q and checking if she is "lucky". This kind of attack is quite a possibility in today's outsourcing scenarios where in an attacker, say, from a call center, would want to know identities in her client's data without really knowing whom to look for. If an adversary is looking for a *particular* individual in the anonymized dataset, then Sweeney's model would generally provide better privacy than our model for it would always yield k matches. For our model to work well against such an adversary, we need to declare the original dataset itself as the universal table $\mathcal{U}$ and carry out anonymization.

In what follows, we build on the strong connection between the number of distinct values assumed by a set of attributes Q and its identity revealing potential that was discovered in Section 2. Intuitively, it is clear from Theorems 1, 2 and 3 that the potency of Q as a quasi-identifier would decrease if we reduce the number of distinct values assumed by Q. This is to be done with appropriate *generalization*. We borrow the following definition of generalization from [33] which has an excellent discussion on this topic.

DEFINITION 5. *Generalization involves replacing (or recoding) a value with a less specific but semantically consistent value.*

EXAMPLE 5. *The original ZIP codes* $\{02138, 02139\}$ *can be generalized to 0213*, thereby stripping the rightmost digit and semantically indicating a larger geographical area.*

One way of looking at generalization is creating $<< D$ partitions of the space of D distinct values and choosing a representative for each partition. In fact, it would give us k-anonymity if we could ensure that most of these partitions are represented by k or more of their own members in the universal table $\mathcal{U}$ with high probability. To make this work, let us suppose that we have got a D'-partition of original D size space such that each partition has probability $1/D'$ (or $O(1/D')$ to be precise). Given a $< p_1, p_2, \ldots, p_D >$ probabilities of the original D size space, such partitioning is certainly possible using techniques we show in Section 4 for a single dimension. Now, we analyze below the bound on D' that is necessary is order to ensure that most of these partitions are represented k or more times in $\mathcal{U}$ with high probability. Please recall that $\mathcal{U}$ has size n and it is built by sampling with replacement.

THEOREM 4. *A data set is probabilistically* $(1-\beta, k)$*-anonymized with respect to a universal table* $\mathcal{U}$ *of size n along the quasi-identifier Q if the number of distinct values along Q,* $D' < \frac{n}{k}(1-c)$ *for some small constant c.*

Before we proceed with the proof, please note that Theorem 4 provides a recommendation for D', the number of partitions of D size space of Q. If the probabilities $< p_1, p_2, \ldots, p_D >$ are known, then as per our earlier assumption, one could cluster these probabilities such that D' equi-probable partitions are created. This concretizes generalization which could be used by any data-holder for anoymizing its data before release.

PROOF. Please refer to the Appendix A for a detailed proof. □

EXAMPLE 6. *Let* $\mathcal{U}$ *be the U.S. Census Table of size* $n = 3 * 10^8$*. Consider the columns* $Q =$ *(*`Gender`*,* `Date of Birth`*,* `Zipcode`*). By Example 2,* $D = 4 * 10^9$*. According to Theorem 4, a dataset is* $(0.9, 100)$ *anonymized along Q with respect to* $\mathcal{U}$ *if we make D′ partitions (or generalizations) of the D size space where*

$$D' \leq \frac{n}{125} = 2.4 * 10^6.$$

Thus, we have to reduce the number of possibilities for Q by a factor of $D/D' < 1700$*. Consider the following generalization (*`Gender`*,* `Half-year of Birth`*,* `First Four Digits of Zipcode`*). Now* $D' = d'_1 * d'_2 * d'_3$*.* d'_1*, the number of distinct values of* `Gender`*, is 2.* d'_2 *is* $60 * 2 = 120$*, and* $d'_3 = 10^4$*. Therefore,* $D' = 2.4 * 10^6$*. This should be good enough to make each row* 100*-anonymous with probability at least* 0.9*.*

3.1 Privacy vs Utility

Note that (`Gender`, `Half-year of Birth`, `First Four Digits of Zipcode`) was just one of many different ways we could have compressed the D size space in Example 6 by factor 1700. Ideally, we would like to devise this generalization such that there is little or no loss in the *data utility*. We frame this problem as an optimization problem below where the goal is to retain maximum utility given privacy constraints.

Let there be m columns $< C_1, C_2, \ldots, C_m >$ that need generalization and $w_1, w_2, \ldots, w_m$ be their respective weights giving their relative importance. We aim to anonymize this multi-column database so that maximum utility is retained in the probabilistically k-anonymized output.

Let $d'_1, d'_2, \ldots, d'_m$ be the number of distinct values along columns $C_1, C_2, \ldots, C_m$ after probabilistic k-anonymization. Then, by Theorem 4,

$$\prod_{i=1}^{m} d'_i \quad = \quad \frac{n}{k}(1-c) = D'.$$

Let us suppose that the quantile based anonymization from Section 4 is used. Thus, d'_i different quantiles are used along the column C_i. Then, the rank difference of the transformation (from Section 4) is approximately $(\frac{n}{d'_i})^2 \times d'_i = \frac{n^2}{d'_i}$.

The sum of the distortion along all columns weighted by the column weights is, therefore, $n^2(\sum_{i=1}^{m} \frac{w_i}{d'_i})$. Minimizing this is equivalent to minimizing $\sum_{i=1}^{m} \frac{w_i}{d'_i}$ subject to $\prod_{i=1}^{m} d'_i = D'$. For a fixed value of product, the sum of numbers is minimized when all the numbers are equal. Therefore,

$$\frac{w_1}{d'_1} = \frac{w_2}{d'_2} = \ldots \frac{w_m}{d'_m} = \frac{1}{d} \text{ (say).}$$

Therefore, $d'_i = d \times w_i \; \forall 1 \leq i \leq m$. The product condition implies, $\prod_{i=1}^{m} d'_i = d^m \prod_{i=1}^{m} w_i = D'$. Therefore,

$$d = (\frac{D'}{\prod_{i=1}^{m} w_i})^{1/m},$$

$$d'_i = (\frac{D'}{\prod_{i=1}^{m} w_i})^{1/m} \times w_i. \quad (1)$$

Note that if d'_i is less than the number of distinct values in column i initially, say d_i, it suggests applying an approach like quantiles proposed here on column C_i. If d'_i is greater than the number of distinct values in column C_i initially, say d_i, then the column C_i is left untouched. The number of distinct elements for other columns can be recalculated (and increased) after this. That is, if $d'_i > d_i$, then the optimization problem over all other variables is first solved after column C_i is eliminated, i.e. Maximize $\sum_{j=1,j\neq i}^{m} \frac{w_j}{d'_j}$ subject to $\prod_{j=1,j\neq i}^{m} d'_j = D'/d_i$.

EXAMPLE 7. *Suppose that we want to probabilistically* $(0.9, 100)$*-anonymize a dataset with* 3 *columns (*`Gender`*,* `Date of Birth`*,* `Zipcode`*) and all columns are equally important, that is , they have equal weight.*

As worked out in Example 9, each row is given 100*-anonymity with probability at least* 0.9 *if* $D' = 2.4 * 10^6$*. As all* 3 *columns have equal weight, we get* $d'_1 = d'_2 = d'_3 \approx 133$*. However* `Gender` *has only* $2 < d'_1$ *values. This means we have to leave it untouched and work with the remaining two attributes. That gives* $d'_2 * d'_3 = 1.2 * 10^6$*. Since both the columns have equal weight, we get* $d'_2 = d'_3 \approx 1.1 * 10^3$*. As* $d'_2 = 1.1 * 10^3$ *is approximately* 60 *(years)*$*$12 *(number of months per year),* `Date of Birth` *is approximated to the month of birth. Also the number of distinct values of* `Zipcode` *being* $O(10^3)$ *implies that the last two digits of* `Zipcode` *are starred out. Thus the anonymization produced is (*`Gender`*,* `Month of Birth`*,* `First Three Digits of Zipcode`*).*

Note that this anonymization was entirely worked out in constant time in the above example. For general case, where the number of columns is m, it would require $O(m^2)$ time. Previous techniques to provide anonymity were not only *NP*-hard in the input size (that means it took exponential time in the dataset) [26, 3] but even approximations required many passes over the database [3, 4]. [23] required passes to be exponential in the number of columns to be anonymized as the lattice developed there took exponential time to be built.

EXAMPLE 8. *According to HIPAA [19], each person must be anonymized in a crowd of* $k = 20,000 = 2 * 10^4$ *people. Now, suppose we want to anonymize a medical records table with columns (*`Gender`*,* `Age` *(In Years),* `Zipcode`*,* `Disease`*).*

As always, the U.S. Census Table is the universal table $\mathcal{U}$ *with* $n = 3 * 10^8$ *rows. The quasi-identifier is (*`Gender`*,* `Age` *(In Years),* `Zipcode`*). As the number of distinct values of* `Gender` *and* `Age` *are* 2 *and* 100 *respectively, the number of distinct values of* `Zipcode` *allowed is approximately* $3 * 10^8/((2 * 10^4) * 2 * 100) = 75$ *by Theorem 4. Therefore,* `Zipcode` *must be anonymized to its first two digits and should only indicate the State.*

3.2 The Curse of Dimensionality

As the number of dimensions (columns) increase, the number of distinct values per column on anonymization decrease rapidly. For example, consider a database table with 25 columns. The aim is to anonymize the table so that 10-anonymity is achieved for the U.S. population of size $3 * 10^8$. Further suppose that all the columns are given equal weight (importance). Applying Theorem 4 and the Multiple Domain Assumption, the number of distinct values per column can be obtained to be roughly 2. Thus all values in a column are generalized to two intervals or converted to two types of values. This hints at reduced data utility measured by any reasonable metric.

This phenomenon was also observed as the curse of dimensionality on k-anonymity [1]. However, we must notice that the previous analysis should only be applied to columns that are available publicly. For example, in the Adults database [8], columns `capgain`, `caploss`, `fnlwgt` and `income` can be assumed to be sensitive columns that are present only in the database itself and are not available for an external join.

3.3 Distinct Values and Anonymity

In this section, we have provided an interesting connection between the number of distinct values taken by a combination of columns and the anonymity it can offer. The main contributions of this association are as follows.

1. This association between distinct values and anonymity guarantee results in an easy technique to obtain a k-anonymized dataset. Merge similar distinct values taken by a column so that the number of distinct values assumed by the column is reduced. The appropriate reduction in the number of distinct values leads to the conversion of a quasi-identifier into k-anonymous columns. As explained in Section 3.1, this would also help retain much of data utility since it minimally distorts ranks. We shall discuss this angle in more detail in the next section.

2. It also helps in coming up with the right kind of generalization for publicly known attributes so that published database can conform to laws like HIPAA.

4. 1-DIMENSIONAL ANONYMITY

The results of Section 3 provide us with the right amount of generalization for each publicly known attribute in order to achieve probabilistic k-anonymity for the entire m column dataset. From any particular attribute point of view, the suggested generalization tries to create appropriate number of buckets (or partitions) in its distinct values space so that each bucket has $k' \gg k$ individuals from the universal table $\mathcal{U}$. Thus, in nutshell, there are m 1-dimensional Sweeney's k-anonymity problems, of course, each with different value of k. Before we proceed further, we will like the reader to take a note of this strong underlying connection between our notion of probabilistic k-anonymity and Sweeney's notion of k-anonymity.

Now k-anonymity for multiple columns is known to be NP-hard [26, 3, 23]. Thankfully we found that this is not the case for a

single column. In the remainder of this section, we showcase various algorithms that help achieve 1-dimensional k-anonymity while retaining maximum possible data utility.

4.1 Numerical Attributes

We start out with algorithms for numerical attributes. Note that they are also applicable to attributes of type date and `Zipcode`.

DEFINITION 6. **k-Anonymous Transformation** *A k-anonymous transformation is a function, f, from $S = \{s_1, s_2, \ldots s_n\}$ to S such that $\forall s_j : |\{f^{-1}(s_j)\}| \geq k$ or $|\{f^{-1}(s_j)\}| = 0$, that is, at least k elements are mapped to each element (which has some element mapped to it) in the range.*

EXAMPLE 9. *Consider $S = \{1, 12, 4, 7, 3\}$, and a function f given by $f(1) = 3$, $f(3) = 3$, $f(4) = 3$, $f(7) = 7$ and $f(12) = 7$. Then f is a 2-anonymous transformation.*

4.1.1 Dynamic Programming

Our goal is to find a k-anonymous transformation that minimizes, say, the maximum cluster size amongst all clusters [34], or the sum of distances to the cluster centers [22], or the sum over all clusters the radius of the cluster times the number of points in the cluster [4]. All these problems are known to be NP-hard for a general metric space. However, for points in a single dimension, we showcase an optimal polynomial time algorithm based on dynamic programming. The details of the algorithm can be found in the Appendix B.

This algorithm needs input in the sorted order. Therefore, its time complexity has two components: 1. Time taken for sorting the input, and 2. time required for the dynamic programming. For input of size n points, sorting takes $O(n \log n)$ time. The dynamic programming part requires time $O(nk)$ as evaluating ClusterCost$(1 \ldots i)$ takes $O(k)$ time for each i. Thus, overall time complexity is $O(n(k + \log n))$.

4.1.2 Quantiles

The algorithm from previous section requires sorting of the input. For large n, this would entail external sort. It is not very desirable in practice. In this section, we explore efficient algorithms that cluster the data in time required to make 1 or 2 sequential passes over the data and use very little extra memory.

DEFINITION 7. **Rank** *Given a set of distinct elements $S = \{s_1, s_2, \ldots, s_n\}$, the rank of an element s_i is r if s_i is the r^{th} largest element in the set.*

For a multi-set containing duplicates, different occurrences of the same element are given consecutive ranks.

EXAMPLE 10. *Among elements $S = \{1, 12, 4, 7, 3\}$, 7 has rank 4, while 3 has rank 2.*

DEFINITION 8. **Rank difference of a transformation** *Given a set $S = \{s_1, s_2, \ldots, s_n\}$ of n numbers, and a k-anonymous transformation f, let $\pi(s_i)$ represent the rank of element s_i. Then, the rank difference incurred by s_i under the transformation f is defined as $|\pi(f(s_i)) - \pi(s_i)|$. The rank difference of the transformation f is the sum of rank difference over all elements, that is, $\sum_{i=1}^{n} |\pi(f(s_i)) - \pi(s_i)|$.*

EXAMPLE 11. *For set $S = \{1, 12, 4, 7, 3\}$, $\pi(1) = 1$, $\pi(12) = 5$, $\pi(4) = 3$, $\pi(7) = 4$ and $\pi(3) = 2$. For f from Example 9, $\pi(f(1)) = 2$, $\pi(f(12)) = 4$, $\pi(f(4)) = 2$, $\pi(f(7)) = 4$, and $pi(f(3)) = 2$. The rank difference of this transformation is 3.*

DEFINITION 9. **Quantile Transformation** *Suppose that $n = qk + r$, where $0 \leq r < k$. Then, the quantile transformation is a k-anonymous transformation that partitions the elements into q contiguous groups of size $(k + \lfloor r/q \rfloor)$ or $(k + \lceil r/q \rceil)$ each. All elements in a group are mapped to the median element of the group.*

THEOREM 5. *The quantile transformation has the minimum rank difference among all k anonymous transformations.*

PROOF. The proof is by a simple greedy argument. □

4.1.3 Efficient Approximate Quantiles using Samples

It is possible to implement the exact quantile transformation. But finding the exact median(quantile) in p passes over the data requires $n^{1/p}$ memory [27]. Thus, to get the exact quantile transformation in 2 passes, would require $\Omega(\sqrt{n})$ memory.

For those who work with smaller memory and/or look for something easier to implement, we sketch a sampling based approach here. We maintain a uniform sample of size $s = \frac{1}{\epsilon^2} log(\frac{1}{\delta})$ using Vitter's sampling technique [35]. The rank t element in the original set is approximated by the rank st/n element in the sample, where n is the size of the original dataset over which the sample is maintained. This element has rank between $t - (\epsilon n)$ and $t + (\epsilon n)$ in the original data with probability greater than $(1 - \delta)$ if the sample size s is chosen as given above [25]. For example suppose that we maintain a uniform sample of 100 elements out of a total $100,000$ elements. Then the $5,000$th element in sorted order among the $100,000$ elements can be approximated well by the 5th element in sorted order from amongst the sample of 100 elements.

4.2 Categorical Attributes

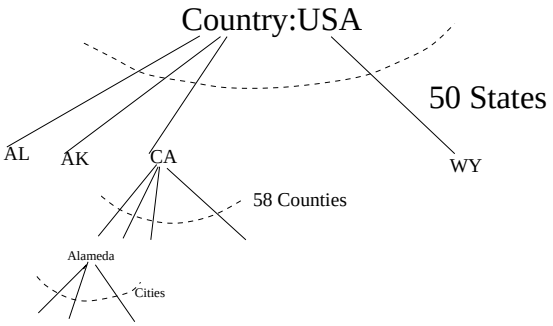

Figure 2: *A Categorical Attribute*

In the previous sub-section, we discussed how to create appropriate buckets or categories for numerical (ordered) attributes. But many a times, there is an attribute with no intrinsic ordering among its value-set. Such an attribute is called as a *categorical attribute*

For categorical attributes we create a layered tree graph as explained. The first layer consists of a node for each category value. The next layer groups together nodes that generalize into one general categorical value, so that they form a single node. This is set to be the parent of the generalized values. This is repeated till there is a single category. Consider for example location information shown in Figure 2. Zipcodes are generalized to cities which are generalized to counties to state and finally to country. The top three levels of the generalization hierarchy are shown. To anonymize this dataset so that there are d distinct values, the generalization is

carried till the level that there are d values. For example, to generalize location so that there are 50 different values, the state information would be retained. However to generalize it to 3000 distinct values, the county information would be retained.

5. EXPERIMENTS

5.1 Quasi-Identifiers

We counted the number of singletons in the Adult Database available from the UCI machine learning repository [8]. The Adult Database has got 32561 rows with 15 attributes, we considered 10 of them and dropped the remaining 5. The dropped attributes are sensitive attributes (not quasi-identifiers): `fnlwgt`, `capgain`, `caploss`, `income` and the attribute `edunum` which is equivalent to the attribute education. In our experiments, we varied the size of the attribute set Q under consideration from 1 to the maximum of 10. The table in Figure 3 shows some of the results that we obtained.

Labels **A1**, **A2**, . . ., **A10** denote the 10 columns of the table. The first row gives the number of distinct values each attribute **A1**, **A2**, . . ., **A10** takes. All other rows (which are labeled with row numbers from 1 to 12) of the table represent publishing the projection of the table along the columns marked 'x'. For example, the row 1 represents publishing the database projected on the `Age` (**A1**) column while the row 12 represents publishing all 10 columns in the database. The column **Size** gives the number of 'x' marks in each row, that is, the number of columns that constitute the quasi-identifier Q under consideration.

The column **S** is the number of rows uniquely identified by the projection of these columns, that is, the number of rows uniquely identified in the published projection. For example, for row 2, where **A1** and **A9** are the attributes of projection, **S** = 986 is returned by the following SQL statement in MS Access:

```
SELECT A1, A9 FROM T
GROUP BY A1, A9
HAVING count(*)=1
```

$\mathbf{F_1}$ is the fraction of rows uniquely identified, given by $\mathbf{S}/32561$ where **S** is the number of singletons while 32561 represents the total number of rows in the database table. For row 2, $\mathbf{F_1} = 0.03$. Some previous definitions of quasi-identifiers [37] measured a quasi-identifier as a set of columns that have a large fraction of unique rows. Thus, $\mathbf{F_1}$ is used as a measure of quasiness. This does not model the external table present with the adversary. For example, by this definition, **A1** and **A9** would together be a 0.03-quasi-identifier.

D is the product of the domain sizes of the attributes marked 'x' in the row. By Multiple Domain Assumption, it is the size of the distinct values space for that combination of columns. For example, for row 3, $\mathbf{D} = 60 * 5 * 2 = 600$.

$\mathbf{F_2}$ captures the notion of quasiness as proposed in Section 2. It is given by $f(D/n)$ shown in Figure 1. Here, D is set to be equal to the value from column **D**, and $n = 3{*}10^8$, the size of US population. Please recall that, by Theorems 1 and 2, $f(D/n) = D/en$ for $D < n$ and $e^{-n/D}$ for $D \geq n$. For all but the last row of the table, $\mathbf{D} < 3{*}10^8$, hence $\mathbf{F_2} = \frac{\mathbf{D}}{2.7*3*10^8}$, for the last row $\mathbf{F_2} = e^{-3*10^8/\mathbf{D}}$.

k-Anon is approximately the probabilistic k-anonymity obtained from the published database. Based on the result of Theorem 4, it is set to $n/\mathbf{D}$, where $n = 3 * 10^8$, the size of the US population. When **D** exceeds n, it is set to 1.

Suppose we are allowed to publish a set of columns with the condition that all 0.2-quasi-identifiers are to be suppressed. If we only consider the entries of the table and look at those projections where at least 0.2 fraction of the rows are unique, then the projections indicated by rows numbered 6, 8, 10, 11 and 12 cannot be published. This is because their $\mathbf{F_1}$ values exceed 0.2.

In fact, our real worry is that > 0.2 fraction of the rows should not get uniquely identified after taking an external join with the universal table $\mathcal{U}$. Then, only row 12 qualifies as a possible 0.2-quasi-identifier as only its $\mathbf{F_2}$ value exceeds 0.2. Note that, from Theorems 1 and 2, there is a minuscule chance of false negatives, that is, rows 1 – 11 are unlikely to be 0.2-quasi-identifiers.

Row 12 needs a closer look since 0.99 is only an upper bound on the expected fraction of unique rows. It may be noticed that many combinations are rare and do not occur. In our example, two attributes **A9** and **A10** are special. **A9** may be represented with only 5 distinct values since the exact hours per week of an individual may not be known and **A10** is not uniformly distributed. Such a case by case analysis of the different attributes may bring down the distinct values, **D**, and hence the fraction of distinct rows. Thus, it can help improve the estimate of quasiness, say, from a 0.99 fraction to (probably) a fraction lower than 0.2. In such a case, row 12 would be a false positive.

5.2 Anonymity Algorithms

We implemented sampling based approximate quantile algorithm (from Section 4.1.3) as a technique in a commercial data masking tool. Our technique required 400 lines of code to be added to the tool. The tool was run on an Oracle database containing 250, 000 rows of a table from a real bank, which was a customer of the tool vendor. The database table was about 1GB in size and had 261 columns. We also repeated our experiments on the public use microdata sample (PUMS) [10] provided by the U.S. Census Bureau. This dataset was given in a flat file format as input to the data masking tool. The experiments were run on a machine with 2.66GHz processor and 504 MB of RAM running Microsoft Windows XP with Service Pack 2.

Scaling with the Dataset Size

We studied how the running time of the quantile algorithm for masking a single column changes as the number of rows in the database table is varied. We measured the time required to mask various fractions of the table, the entirety of which contains 250, 000 rows. The time required to mask this single numeric column with $k = 10,000$ anonymity (so that there are 25 different quantiles to which the data is approximated) increased linearly to a total of about 10 seconds for the entire column. A straight line with almost exactly identical slope and coordinates was obtained for the PUMS [10] dataset.

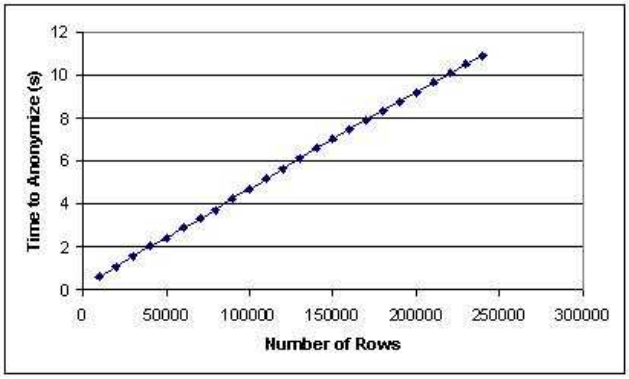

Figure 4: *Time taken for varying number of rows.*

Row	Size	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	S	F_1	D	F_2	k-Anon
		60	8	15	7	14	6	5	2	20	40					
1	1	x										2	$6.1 * 10^{-5}$	60	$7.4 * 10^{-8}$	$5 * 10^6$
2	2	x								x		986	0.03	1200	$1.48 * 10^{-6}$	$2.5 * 10^5$
3	3	x						x	x			65	0.002	600	$7.4 * 10^{-7}$	$5 * 10^5$
4	4	x	x	x		x						5056	0.16	$1 * 10^5$	$1.2 * 10^{-4}$	$3 * 10^3$
5	4	x	x			x					x	3105	0.095	$2.7 * 10^5$	$3.3 * 10^{-4}$	$1.1 * 10^3$
6	4	x				x				x	x	7581	0.23	$6.7 * 10^5$	$8.3 * 10^{-4}$	450
7	4		x	x		x					x	1384	0.043	$6.7 * 10^4$	$8.3 * 10^{-5}$	$4.5 * 10^3$
8	5	x	x	x		x					x	7659	0.235	$4 * 10^6$	$4.9 * 10^{-3}$	75
9	5	x	x		x	x	x					5215	0.16	$2.8 * 10^5$	$3.4 * 10^{-4}$	$1 * 10^3$
10	5	x	x			x	x			x		12870	0.40	$8 * 10^5$	$9.9 * 10^{-4}$	380
11	5	x	x			x				x	x	10402	0.32	$5.4 * 10^6$	$6.7 * 10^{-3}$	55
12	10	x	x	x	x	x	x	x	x	x	x	24802	0.76	$33 * 10^9$	0.99	1

Size = Number of columns that make the quasi-identifier, A1 = Age, A2 = Work class, A3 = Education, A4 = Marital status, A5 = Occupation, A6 = Relationship, A7 = Race, A8 = Sex, A9 = Hours per week, A10 = Native country, S = Number of singletons in the current table, F_1= Fraction of singletons using the table itself = S/32561, F_2=Fraction of singletons using Figure 1 and $n = 3 * 10^8$ for US population, k-Anon= Anonymity parameter for the published database = n/D.

Figure 3: Quasi-Identifiers on the Adult Dataset

Scaling with the Number of Columns Masked

We studied how the running time of the quantile algorithm for masking multiple columns varies as the number of columns to be masked is varied. For this experiment too, we used the table with 250, 000 rows and 261 columns. As each column is independently anonymized, the time taken increases linearly as the number of columns being anonymized increases. Previous algorithms [23] had an exponential increase in the time taken for anonymization as the number of columns increased as the lattice created was exponential in the number of columns being anonymized.

The time taken to anonymize 10 columns of data with 250, 000 rows was approximately 100 seconds. This is almost an order of magnitude improvement over the previous algorithm [23]. The results on the PUMS dataset were similar.

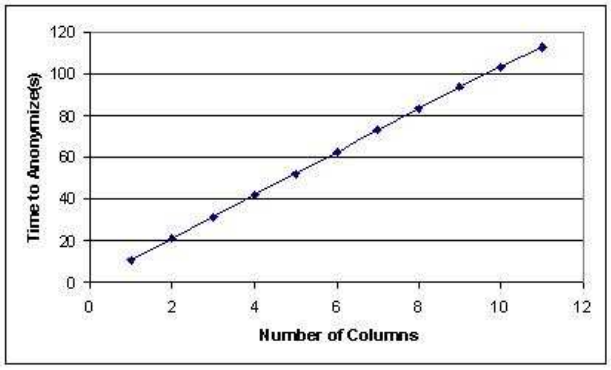

Figure 5: *Time taken for varying number of columns.*

Scaling with the Anonymity Parameter

The implemented algorithm does a binary scan over all buckets to find the bucket closest to each data item. The time required to anonymize a data value, therefore, logarithmically increases as the number of buckets increases (or the value k of anonymity parameter decreases). If b is the number of buckets and n the number of rows, then the time to anonymize is $n\log(b)$. The time taken to read n rows from disk is nC where C is a large constant. The total time taken is, therefore, $n(C + \log b)$ where $C \gg \log(b)$. This explains the shape of the curve in Figure 6. Here $nC \approx 10$ seconds and the $\log(b)$ term explains the slight increase from 0 to 500 buckets.

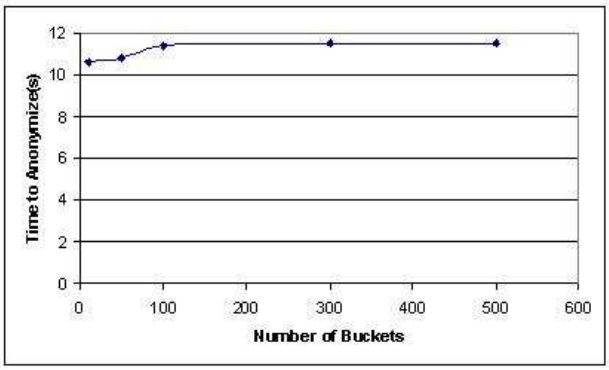

Figure 6: *Time taken for varying number of buckets.*

Tradeoff between Privacy and Utility

We studied how the error introduced in a column as a result of k-anonymization varies with the anonymity parameter k. Let x_i be the original value of the i^{th} row. Let x_i' be its value after k-anonymization. Then $(x_i' - x_i)^2$ is the error introduced for row i as a result of k-anonymization. The total error introduced over n rows is $Error = \sum_{i=1}^{n}(x_i' - x_i)^2$. Let $\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$. If all x_i' are constrained to be identical (corresponding to anonymity with a single bucket), then $\overline{x}$ gives the minimum error according to the above metric, i.e. it gives $MinError = Min_x \sum_{i=1}^{n}(x - x_i)^2 = \sum_{i=1}^{n}(\overline{x} - x_i)^2$. We, therefore, normalize the error as Error/MinError.

The curve is plotted in Figure 7 where the normalized error is plotted on the y-axis while the number of buckets, $b = \frac{n}{k}$, is plotted on the x-axis. An almost identical curve was obtained for the PUMS dataset. The curve very closely follows the curve $\frac{1}{b^2}$. This could be proven analytically.

Thus, for given n and k, we find that the identity disclosure risk is $< 1/k$ (for "join" class of attacks) and the error introduced in data is $\propto k^2/n^2$. We may, therefore, boldly quantify the privacy provided by k-anonymization as $p = 1 - 1/k$ and the utility retained as $u = 1 - k^2/n^2$ implying the following privacy-utility trade-off

equation.

$$(1-p)^2(1-u) = 1/n^2 \text{ (a constant).}$$

Note that, the fact that we used sum square errors, instead of sums of absolute values of errors explains the square term above.

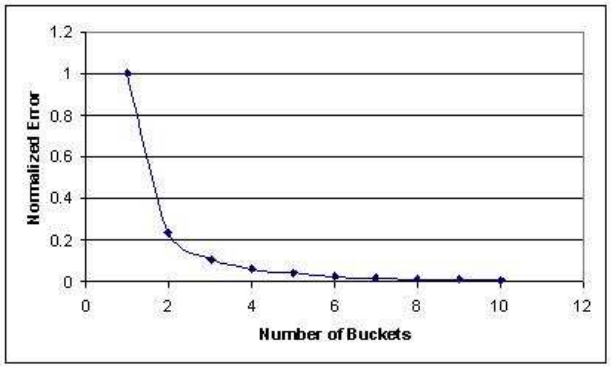

Figure 7: *Tradeoff between privacy and utility.*

6. RELATED WORK

One of the earliest definitions of quasi-identifier can be found in Dalenius [16]. [33, 32] and [23] use a similar definition.

Samarati and Sweeney formulated the *k*-anonymity framework and suggested mechanisms for *k*-anonymization using the ideas of generalization and suppression [29, 33, 32]. Subsequent work has shown some NP-hardness results [26, 2, 4] and that has inspired many interesting heuristics and approximation algorithms [21, 36, 26, 7, 2, 23, 24, 4]. All of this work assumes that quasi-identifier attribute sets are known based on specific knowledge domain.

The basic theme of *k*-anonymity model is to *hide* an individual in a crowd of size *k* or more. A similar intuition is pursued by Chawla et al in [13] who, in fact, manage to convert it into a precise mathematical statement. They not only give definition of privacy and its compromise for statistical databases, but also provide a method for describing and comparing the privacy offered by specific sanitization techniques. They also give a formal definition of an *isolating* adversary whose goal is to single out someone from the crowd with the help of some *auxiliary* information *z*. This work is further extended in [14] where Chawla et al study privacy-preserving histogram transformations that provide substantial utility.

There is a wide consensus that privacy is a corporate responsibility [20]. In order to help and ensure corporations fulfil this responsibility, governments all over the world have passed multiple privacy acts and laws, for example, Gramm-Leach-Bliley (GLB)Act [18], Sarbanes-Oxley (SOX) Act [30], Health Insurance Portability and Accountability Act (HIPAA) [19] are some such well known U.S. privacy acts. In fact, HIPAA recommends the following *safe-harbor* method of de-identification in which it provides clear guidelines for sanitizing quasi-identifiers including date types, `Zipcode`, etc. For $20,000$ anonymity, HIPAA advises to retain essentially only the State information in `Zipcode` and year information in `Date of Birth` which is quite inline with what we concluded in Examples 6, 7 and 8 based on our analysis. The de-identification excerpt from the HIPAA law is provided in Appendix C.

7. CONCLUSIONS

In this paper, we provided the first formalism and a practical technique to identify a quasi-identifier. Along the way we discovered an interesting connection between whether a set of columns forms a quasi-identifier and the number of distinct values assumed by the combination of the columns.

Then we defined a new notion of anonymity called as probabilistic anonymity where in we insist that each row of the anonymized dataset should match with at least *k* or more rows of the universal table $\mathcal{U}$ along a quasi-identifier. We observed that this new notion of anonymity is similar to the existent *k*-anonymity notion in terms of privacy guarantees and is sufficiently strong for many real life scenarios involving oblivious adversaries. Building on our earlier work, we found an interesting connection between the number of distinct values taken by a combination of columns and the anonymity it can offer. This allowed us to find an ideal amount of generalization or suppression to apply to different columns in order to achieve probabilistic anonymity. We worked through many examples and showed that our analysis can be used to make a published database conform to privacy acts like HIPAA.

In order to achieve the probabilistic anonymity, we observed that one needs to solve multiple 1-dimensional *k*-anonymity problems. We proposed many efficient and scalable algorithms for achieving 1-dimensional anonymity. Our algorithms are optimal in a sense that they minimally distort data and retain much of its utility.

8. REFERENCES

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 2005 International Conference on Very Large Data Bases*, pages 901–909, 2005.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *Proceedings of the International Conference on Database Theory*, pages 246–258, 2005.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-Anonymity. *Journal of Privacy Technology*, 20051120001, 2005. Earlier version appeared in Proc. of the Intl. Conf. on Database Theory (ICDT 2005).

[4] G. Aggarwal, T. Feder, K. Kenthapadi, R. Panigrahy, D. Thomas, and A. Zhu. Clustering for privacy. In *Proceedings of the ACM Symposium on Principles of Database Systems*, 2006.

[5] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the International Conference on Very Large Data Bases*, pages 487–499, Santiago, Chile, September 1994.

[6] K. Baum. First estimates from the national crime victimization survey: Identity theft, 2004. *Bureau of Justice Statistics Bulletin*, Apr. 2006. Available from URL: http://www.ojp.usdoj.gov/bjs/pub/pdf/it04.pdf.

[7] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the International Conference on Data Engineering*, pages 217–228, 2005.

[8] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. Available from URL: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[9] M. Brown. Identity theft victim stories: Verbal testimony by michelle brown, July 2000. Privacy Rights ClearingHouse. Available from URL: http://www.privacyrights.org/cases/victim9.htm.

[10] U. C. Bureau. Public use microdata sample (PUMS). http://www.census.gov/acs/www/Products/PUMS/.

[11] U. Census. Accuracy of the US census data. Available from URL: http://www.census.gov/acs/www/UseData/Accuracy/Accuracy1.htm.

[12] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2003.

[13] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward privacy in public databases. In *2nd Theory of Cryptography Conference (TCC)*, pages 363–385, 2005.

[14] S. Chawla, C. Dwork, F. McSherry, and K. Talwar. On the utility of privacy-preserving histograms. In *21st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

[15] H. Chernoff. Asymptotic efficiency for tests based on the sums of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[16] T. Dalenius. Finding a needle in a haystack or identifying anonymous census records. In *Journal of Official Statistics (2)*, pages 329–336, 1986.

[17] P. B. Gibbons. Distinct sampling for highly-accurate answers to distinct values queries and event reports. In *Proceedings of the International Conference on Very Large Data Bases*, pages 541–550, 2001.

[18] GLB. Gramm-Leach-Bliley Act. Available from URL: http://www.ftc.gov/privacy/privacyinitiatives/glbact.html.

[19] HIPAA. Health Information Portability and Accountability Act. Available from URL: http://www.hhs.gov/ocr/hipaa/.

[20] IBM. Privacy is good for business. Available from URL: http://www-306.ibm.com/innovation/us/customerloyalty/harriet_pearson_interview.shtml.

[21] V. Iyengar. Transforming data to satisfy privacy constraints. In *8th ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining*, pages 279–288, 2002.

[22] K. Jain and V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science*, pages 2–13, 1999.

[23] K. Lefevre, D. J. Dewitt, and R. Ramakrishnan. Incognito: efficient full domain k-anonymity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 49–60, 2005.

[24] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the International Conference on Data Engineering*, page 24, 2006.

[25] G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 251–262, 1999.

[26] A. Meyerson and R. Williams. On the complexity of optimal *k*-anonymity. In *Proceedings of the ACM Symposium on Principles of Database Systems*, pages 223–228, June 2004.

[27] I. Munro and M. Paterson. Selection and sorting with limited storage. In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science*, pages 253–258, 1978.

[28] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987.

[29] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of the ACM Symposium on Principles of Database Systems*, page 188, 1998.

[30] SOX. Sarbanes-Oxley Act. Available from URL: http://www.sec.gov/about/laws/soa2002.pdf.

[31] L. Sweeney. Uniqueness of simple demographics in the U.S. population. In *LIDAP-WP4. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA*, 2000.

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

[33] L. Sweeney. *k*-Anonymity: A model for preserving privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[34] V. Vazirani. *Approximation Algorithms*. Springer, 2004.

[35] J. Vitter. Random sampling with a reservoir. *ACM Transaction on Mathematical Software*, pages 37–57, 1985.

[36] W. Winkler. Using simulated annealing for k-anonymity. *Research Report 2002-07, US Census Bureau Statistical Research Division*, November 2002.

[37] Y. Xu and R. Motwani. Random sampling based algorithms for efficient semi-key discovery, 2006. Available from URL: http://theory.stanford.edu/~xuying/papers/minkey_vldb.pdf.

APPENDIX

A. PROOFS

PROOF:[of theorem 1] If $f(x) = xe^{-x}$, $f'(x) = (1-x)e^{-x}$ and $f''(x) = (x-2)e^{-x}$. Thus, the function f has a global maximum at $x = 1$, since $f'(1) = 0$ and $f''(1) < 0$.

Now the expected number of singletons,

$$\sum_{i=1}^{D} x_i e^{-x_i} \leq \sum_{i=1}^{D} e^{-1} = \frac{D}{e}.$$

This expression is a tight upper bound on the expected number of singletons for $D \leq n$. For example, it is almost obtained by setting $x_i = 1$, for $i = 1, 2, \ldots, D-1$, and $x_D = n - D + 1$. □

PROOF:[of theorem 2] If $f(x) = xe^{-x}$, $f'(x) = (1-x)e^{-x}$ and $f''(x) = (x-2)e^{-x}$. The function f has a point of inflection at $x = 2$, since $f''(x) < 0$ for $x < 2$ implying the function is concave here, and $f''(x) > 0$ for $x > 2$ implying the function is convex here.

First we claim that on maximizing $\sum_{i=1}^{D} x_i e^{-x_i}$, no $x_i \geq 2$. Suppose otherwise: after maximizing $\sum_{i=1}^{D} x_i e^{-x_i}$, some $x_a \geq 2$. As $D \geq n$, and $\sum_{i=1}^{D} x_i = n$, some $x_b < 1$. For some small δ, replacing x_a by $x_a - \delta$ and x_b by $x_b + \delta$ we retain $\sum_{i=1}^{D} x_i = n$. As $f(x) = xe^{-x}$ increases towards x=1, $f(x_a - \delta) > f(x_a)$ and $f(x_b + \delta) > f(x_b)$. Thus $\sum_{i=1}^{D} x_i e^{-x_i}$ is increased, contradicting the fact that it was maximized. Thus, $\forall 1 \leq i \leq D$, $x_i < 2$.

Now $f''(x) < 0$ for $0 \leq x < 2$. Since f is concave, we can apply Jensen's inequality [28] [2] to get

$$\begin{aligned}
\sum_{i=1}^{D} x_i e^{-x_i} &= D \sum_{i=1}^{D} \frac{1}{D} x_i e^{-x_i} \\
&\leq D \cdot (\sum_{i=1}^{D} \frac{x_i}{D}) e^{-(\sum_{i=1}^{D} \frac{x_i}{D})} \\
&= n e^{\frac{-n}{D}}.
\end{aligned}$$

Thus, if $D \geq n$, the expected number of singletons is bounded above by $ne^{\frac{-n}{D}}$. □

PROOF OF THEOREM 3. Note that $D > n$. If not, then, by Theorem 1, the maximum expected fraction of rows taking unique values is $D/en \leq 1/e < \alpha$.

From Theorem 2, the maximum expected fraction of rows taking unique values along the columns with D distinct values is $e^{-n/D}$. For the the set of rows to form an α-quasi-identifier, this fraction must be larger than α. Thus, $e^{-n/D} > \alpha$, which implies that $D > \frac{n}{ln(1/\alpha)}$. □

PROOF OF THEOREM 4. Let us suppose that we have got a D'-partition of original D size space of quasi-identifier Q such that each partition has probability $1/D'$. Let X_i denote the indicator variable if $\geq k$ rows in the universal table $\mathcal{U}$ are chosen from the i^{th} partition.

$$\begin{aligned}
P[X_i = 1] &= \sum_{j=k}^{n} \binom{n}{j} (\frac{1}{D'})^j (1 - \frac{1}{D'})^{n-j} \\
&= 1 - \sum_{j=0}^{k-1} \binom{n}{j} (\frac{1}{D'})^j (1 - \frac{1}{D'})^{n-j} \\
&\geq 1 - exp(\frac{-D'(n/D' - (k-1))^2}{2n}) \\
&\quad \text{(by Chernoff bounds [15])} \\
&= 1 - exp(\frac{-(n - (k-1)D')^2}{2nD'}).
\end{aligned}$$

For $1 - \beta$ probability guarantee, we would like to have

$$1 - exp(\frac{-(n - (k-1)D')^2}{2nD'}) \geq 1 - \beta,$$

that is,

$$\frac{-(n - (k-1)D')^2}{2nD'} \leq ln\beta.$$

This is true when,

$$0 \leq D'^2 + \frac{2nD'}{k-1}\left(\frac{ln\beta}{k-1} - 1\right) + \left(\frac{n}{k-1}\right)^2,$$

that is,

$$D' \leq \frac{n}{k-1}(1 + x - \sqrt{x^2 + 2x}),$$

where

$$x = \frac{-ln\beta}{k-1}.$$

This implies that

$$D' \leq \frac{n}{k}(1 - c)$$

is sufficient for some small constant c. □

B. ALGORITHM OF SECTION 4.1.1

If not already sorted, first sort the input and suppose that it is $p_1 < p_2 < \ldots < p_n$. For $1 \leq a < b \leq n$, let Cluster(a, b) be the cost to cluster elements $p_a, \ldots, p_b$.

Consider the optimal clustering of the input points. Note that each cluster in the optimal clustering contains a set of contiguous elements. Moreover, each cluster is of size at least k by the k-anonymity requirement. Since any cluster of size $\geq 2k$ can be broken into two contiguous clusters of size at least k each and that would reduce the clustering cost, the size of a cluster in the optimal clustering will be at most $2k - 1$.

The optimal clustering of the n input points is, therefore, the optimal clustering of points p_1, p_2, p_{n-i} and one single cluster of the points $(p_{n-i+1}, \ldots, p_n)$, where i is the size of the last cluster. Note that $k \leq i < 2k$ by the previous analysis. Therefore we find the optimal clustering by trying out all possible values of $i \in \{k, k+1, \ldots, 2k-1\}$. Now, the dynamic programming recursive equation is given by

ClusterCost$(1, n) = min_{k \leq i < 2k}$ Cost(ClusterCost$(1, n-i)$, Cluster$(n-i+1, n)$).

Here Cost(A, B) is the sum for a metric like the k-median [22] or cellular [4] metric which minimizes the sum of costs over all clusters. It is the maximum function for the k-center metric [34] which minimizes the maximum of cluster sizes amongst all clusters.

ClusterCost(a, b) is initially set to ∞ if $b-a+1 < k$. For $b-a+1 \geq k$, ClusterCost(a, b) is initially set to the cost of clubbing all points into a single cluster, that is, Cluster(a, b).

[2] If f is a concave function, and $\sum_{i=1}^{m} p_i = 1$, with $p_i \geq 0$ $\forall i$, then $\sum_{i=1}^{m} p_i f(x_i) \leq f(\sum_{i=1}^{m} p_i x_i)$.

C. DE-IDENTIFICATION REQUIRED FOR HIPAA

"The following identifiers of the individual or of relatives, employers, or household members of the individual must be removed to achieve the "safe harbor" method of de-identification: (A) Names; (B) All geographic subdivisions smaller than a State, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of a zip code if, according to the current publicly available data from the Bureau of Census (1) the geographic units formed by combining all zip codes with the same three initial digits contains more than 20,000 people; and (2) the initial three digits of a zip code for all such geographic units containing 20,000 or fewer people is changed to 000; (C) All elements of dates (except year) for dates directly related to the individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older; (D) Telephone numbers; (E) Fax numbers; (F) Electronic mail addresses: (G) Social security numbers; (H) Medical record numbers; (I) Health plan beneficiary numbers; (J) Account numbers; (K) Certificate/license numbers; (L) Vehicle identifiers and serial numbers, including license plate numbers; (M) Device identifiers and serial numbers; (N) Web Universal Resource Locators (URLs); (O) Internet Protocol (IP) address numbers; (P) Biometric identifiers, including finger and voice prints; (Q) Full face photographic images and any comparable images; and (R) any other unique identifying number, characteristic, or code, except as permitted for re-identification purposes provided certain conditions are met. In addition to the removal of the above-stated identifiers, the covered entity may not have actual knowledge that the remaining information could be used alone or in combination with any other information to identify an individual who is subject of the information. 45 C.F.R. §164.514(b). "