Multidimensional K-Anonymity

Kristen LeFevre David J. DeWitt Raghu Ramakrishnan
University of Wisconsin, Madison
Department of Computer Sciences Technical Report 1521
Revised June 22, 2005

Abstract

K-Anonymity has been proposed as a mechanism for privacy protection in microdata publishing, and numerous recoding "models" have been considered for achieving k-anonymity. This paper proposes a new multidimensional model, which provides an additional degree of flexibility not seen in previous (single-dimensional) approaches. Often this flexibility leads to higher-quality anonymizations, as measured both by general-purpose metrics, as well as more specific notions of query answerability.

In this paper, we prove that optimal multidimensional anonymization is NP-hard (like previous k-anonymity models). However, we introduce a simple, scalable, greedy algorithm that produces anonymizations that are a constant-factor approximation of optimal. Experimental results show that this greedy algorithm frequently leads to more desirable anonymizations than two optimal exhaustive-search algorithms for single-dimensional models.

1. Introduction

A number of organizations publish microdata for purposes such as demographic and public health research. In order to maintain individual privacy, the released data must be "de-identified" by removing attributes known to uniquely identify individuals, such as Name and Social Security Number. In addition, this process must account for the possibility of combining certain other attributes with external data to uniquely identify individuals [15]. For example, an individual might be "re-identified" by joining the released data with another (publicly available) database on Age, Sex, and Zipcode. Figure 1 shows such an attack, where Ahmed's medical information is determined by joining a released table of patient data with a public voter registration list.

K-anonymity has been proposed to reduce the risk of this type of attack [12, 13, 15]. The primary goal of k-anonymization is to protect the privacy of the individuals to whom the data pertains. However, subject to this constraint, it is important that the released data remain as "useful" as possible. Numerous recoding *models* have been proposed in the literature for k-anonymization [8, 9, 13, 17, 10]. Often the "quality," or utility, of the published data is dictated by the model that is used. The main contributions of this paper are a new multidimensional recoding model and a greedy algorithm for k-anonymization, an approach with several important advantages:

Voter Registration Data

Name	Age	Sex	Zipcode
Ahmed	25	Male	53711
Brooke	28	Female	55410
Casey	31	Female	90210
Dave	19	Male	02174
Evelyn	40	Female	02237

Patient Data

Age	Sex	Zipcode	Disease
25	Male	53711	Flu
25	Female	53712	Hepatitis
26	Male	53711	Brochitis
27	Male	53710	Broken Arm
27	Female	53712	AIDS
28	Male	53711	Hang Nail

Figure 1. Tables vulnerable to a joining attack

- The greedy algorithm is substantially *more efficient* than optimal k-anonymization algorithms that have been proposed for single-dimensional models [2, 9, 12]. The time complexity of the greedy algorithm is $O(nlogn)$, whereas the exhaustive-search algorithms are exponential in the worst case.
- The greedy multidimensional algorithm often produces *better quality* results than optimal single-dimensional algorithms, thus producing better results than the many existing single-dimensional heuristic [6, 14, 16] and stochastic search [8, 18] algorithms.

1.1. Basic Definitions

Quasi-Identifier Attribute Set A quasi-identifer is a minimal set of attributes $X_1, ..., X_d$ in table T that can be joined with external information to re-identify individual records. We assume that the quasi-identifier is well-understood based on knowledge of the domain.

Equivalence Class A table T consists of a multiset of tuples. An equivalence class for T with respect to attributes $X_1, ..., X_d$ is the set of all tuples in T containing identical values $(x_1, ..., x_d)$ for $X_1, ..., X_d$. In SQL, this is equivalent to the results of a GROUP BY query on attributes $X_1, ..., X_d$.

K-Anonymity Property Table T is said to satisfy k-anonymity (or to be *k-anonymous*) with respect to attributes $X_1, ..., X_d$ if every unique tuple $x_1, ..., x_d$ in the (multiset) projection of T on $X_1, ..., X_d$ occurs at least k times. That is, the size of each equivalence class in T with respect to $X_1, ..., X_d$ is at least k.

K-Anonymization A view V of relation T is said to be a k-anonymization of T if the view modifies or generalizes the data of T according to some **model** such that V satisfies the k-anonymity property with respect to the quasi-identifier.

1.2. General-Purpose Quality Metrics

There are a number of notions of k-anonymization quality [2, 6, 8, 10, 12, 13, 14, 15, 16], but intuitively, anonymization should generalize or perturb the data as little as is necessary to satisfy the k-anonymity constraint. Here we consider some simple general-purpose quality metrics, but a more targeted approach to quality measurement based on query workload is described in Section 5.

The simplest kind of quality measure for k-anonymization V is based on the size of the equivalence classes E in V. The *discernability metric*, described in [2], defines the cost of an anonymization as follows:

$C_{DM} = \sum_{EquivClasses\ E} |E|^2$

Intuitively, this metric assigns to each tuple t in V a penalty, which is determined by the size of the equivalence class containing t.

As an alternative, we also propose the *normalized average equivalence class size metric*:

$C_{AVG} = (\frac{total\_records}{total\_equiv\_classes})/(k)$

This metric has a slightly more intuitive meaning, but it also measures the quality of an anonymization based on the size of the equivalence classes contained therein.

1.3. Paper Overview and Contributions

The first contribution of this paper is a new multidimensional partitioning model for k-anonymization, described in Section 2. Like previous k-anonymity problems [1, 10], optimal k-anonymization using this new model is NP-hard. For this reason, we consider the worst-case maximum size of equivalence classes, and we find that this upper bound is $O(k)$ in the multidimensional case, while in the single-dimensional model, this bound can grow linearly with the number of records. A simple variation of the multidimensional model, described in Section 3 has a maximum upper-bound of $2k$.

Following these results, we introduce a simple greedy algorithm for multidimensional anonymization in Section 4. This algorithm is scalable to large data sets, and for the general-purpose quality metrics described in Section 1.2, the results are a constant-factor approximation of optimal.

General-purpose quality metrics are a good starting point when the ultimate use of the published data is unknown. However, in some cases, the data publisher might want to "optimize" for a particular purpose (while maintaining the k-anonymity constraint). Section 5 introduces a more sophisticated notion of quality measurement, based on a workload of aggregate queries.

In Section 6 we describe our experimental evaluation, which compares the quality of anonymizations obtained by our greedy algorithm with those obtained using exhaustive optimal algorithms for two proposed single-dimensional models. Our results indicate that the greedy algorithm often produces better quality results, as measured both by general-purpose cost metrics and a simple query workload.

Finally, discussions of related and future work are provided in Sections 7 and 8.

2. Multidimensional Global Recoding

In a relational database, there is some domain of values associated with each attribute. We use the notation D_X to denote the domain of attribute X. A *global recoding* seeks to achieve k-anonymity by mapping the domains of the quasi-identifier attributes to generalized or altered values [17].

Global recoding can be further broken down into two sub-classes [9]. A *single-dimensional global recoding* is defined by a function $\phi_i : D_{X_i} \rightarrow D'$ *for each attribute* X_i of the quasi-identifier. An anonymization V is obtained by applying each ϕ_i to the values of X_i in each tuple of T.

Alternatively, a *multidimensional global recoding* is defined by a *single* function $\phi : D_{X_1} \times ... \times D_{X_n} \rightarrow D'$, which is used to recode the domain of value vectors associated with the set of quasi-identifier attributes. Under this model, an anonymization V is obtained by applying ϕ to the vector of quasi-identifier values in each tuple of T.

Partitioning models have been considered in the literature for defining recoding functions for totally-ordered domains [2, 8], such as numeric attributes. However, these previous proposals have considered only single-dimensional recoding. A *single-dimensional interval* is defined by a pair of endpoints $p, v \in D_{X_i}$ such that $p \leq v$. (The endpoints of such an interval may be open or closed, in order to handle continuous domains.)

Single-dimensional Partitioning Assume there is a total order associated with the domain of each quasi-identifier

Age	Sex	Zipcode	Disease
[25-28]	Male	[53710-53711]	Flu
[25-28]	Female	53712	Hepatitis
[25-28]	Male	[53710-53711]	Brochitis
[25-28]	Male	[53710-53711]	Broken Arm
[25-28]	Female	53712	AIDS
[25-28]	Male	[53710-53711]	Hang Nail

Figure 2. A single-dimensional anonymization of Patients

Age	Sex	Zipcode	Disease
[25-26]	Male	53711	Flu
[25-27]	Female	53712	Hepatitis
[25-26]	Male	53711	Brochitis
[27-28]	Male	[53710-53711]	Broken Arm
[25-27]	Female	53712	AIDS
[27-28]	Male	[53710-53711]	Hang Nail

Figure 3. A multidimensional anonymization of Patients

attribute X_i. A single-dimensional partitioning defines, for each X_i, a set of *non-overlapping* single-dimensional intervals that cover D_{X_i}. ϕ_i maps each $q \in D_{X_i}$ to some *summary statistic* for the interval in which it is contained.

The released data will include simple statistics that summarize the intervals they replace. For now, we assume that these summary statistics are min-max ranges, but we discuss some other possibilities in Section 5.

One of the main contributions of this paper is to extend partitioning-based anonymization to multidimensional recoding. Again, assume a total order for each D_{X_i}. A *multidimensional region* is defined by a pair of d-tuples $(p_1, ..., p_d)$, $(v_1, ..., v_d) \in D_{X_1} \times ... \times D_{X_d}$ such that $\forall i, p_i \leq v_i$. Conceptually, each region is bounded by a d-dimensional rectangular box, and each edge and vertex of this box may be either open or closed to provide flexibility for continuous domains.

Strict Multidimensional Partitioning A strict multidimensional partitioning defines a set of non-overlapping multidimensional regions that cover $D_{X_1} \times ... \times D_{X_d}$. ϕ maps each tuple $(x_1, ..., x_d) \in D_{X_1} \times ... \times D_{X_d}$ to a summary statistic for the region in which it is contained.

When ϕ is applied to table T (assuming each region is mapped to a unique vector of summary statistics), each non-empty region is an equivalence class in V. For simplicity, we again assume that these summary statistics are ranges, and further discussion is provided in Section 5.

Sample 2-anonymizations of Patients, using single-dimensional and multidimensional partitioning are shown in Figures 2 and 3, respectively. Notice that the anonymization obtained using the multidimensional model is not permissible under the single-dimensional model because the domains of Age and Zipcode are not recoded to a single set of intervals (e.g., Age 25 is mapped to either [25-26] or [25-27], depending on the values of Zipcode and Sex). However, the single-dimensional generalization is also valid under the multidimensional model.

Proposition 1 *Every single-dimensional partitioning for quasi-identifier attribute sets $X_1, ..., X_d$ can be expressed as a strict multidimensional partitioning. However, when $d \geq 2$ and $\forall i, |D_{X_i}| \geq 2$, there exists a strict multidimensional partitioning that cannot be expressed as a single-dimensional partitioning.*

Proof Sketch The single-attribute intervals that define a single-dimensional partitioning induce a set of multidimensional regions, which in turn define a multidimensional partitioning. However, when there is more that one quasi-identifier attribute, and each quasi-identifier attribute has a domain of size at least 2, we can construct a multidimensional partitioning that does not recode one of the attributes to a single set of intervals, and is thus not a valid single-dimensional partitioning. □

This indicates that the optimal strict multidimensional partitioning must be at least as good as the optimal single-dimensional partitioning. However, in Section 2.2 we prove that the optimal k-anonymous multidimensional partitioning problem is NP-hard. For this reason, we also consider the worst-case bounds on partition size, and we show that the maximum size of a region resulting from a minimal multidimensional partitioning is $O(k)$, for a constant-sized quasi-identifier. In Section 2.4, we show that for the single-dimensional model this bound may be linear in the number of tuples in T. These results provide some useful guidelines, though they do not indicate that either model would necessarily lead to better results for any specific algorithm or database.

2.1. Spatial Representation

Throughout the rest of this paper, it is convenient to represent quasi-identifier attribute sets using a spatial framework. Consider table T with quasi-identifier attributes, $X_1, ..., X_d$, and assume that there exists some total ordering for each domain D_{X_i}. The (multiset) projection of $X_1, ..., X_d$ on T can be represented in d-dimensional space, and each record in T is a *point* in this space. For example, Figure 4(a) shows the two-dimensional representation of Patients from Figure 1, for quasi-identifier attributes *Age* and *Zipcode*.

Similar models have been considered for rectangular partitioning in 2 dimensions [11]. In this context, the single-dimensional and multidimensional partitioning models are analogous to the "$p \times p$" and "arbitrary" classes of tilings, respectively. However, to the best of our knowledge, none of the previous optimal tiling problems have included constraints requiring minimal partition occupancy.

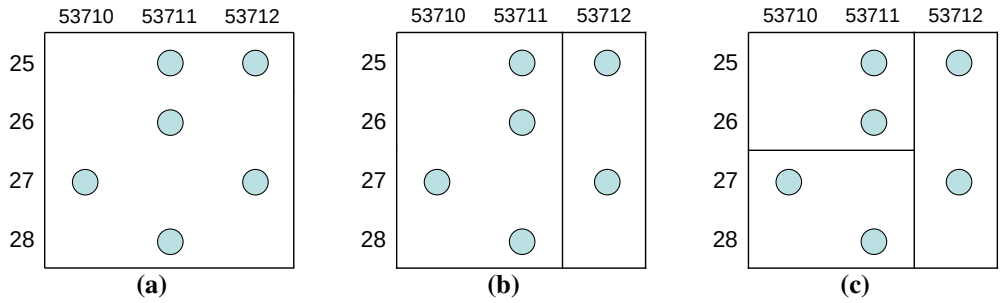

Figure 4. Spatial representation of (a) Patients table, (b) a single-dimensional partitioning, and (c) a strict multidimensional partitioning, for quasi-identifier attributes *Zipcode* and *Age*

2.2. Optimal Strict Multidimensional Partitioning is NP-Hard

There have been several previous hardness results for optimal k-anonymization under other recoding models, including minimum attribute- and cell-suppression [1, 10]. The problem of *optimal strict k-anonymous multidimensional partitioning* (finding the strict k-anonymous multidimensional partitioning with the smallest C_{DM} or C_{AVG}) is also NP-hard, but this result does not follow directly from the previous results. We formulate the following decision problem for strict multidimensional partitioning using C_{AVG} [1]:

K-Anonymous Strict Multidimensional Partitioning
For multiset of points P in d-dimensional space, is there a strict multidimensional partitioning such that every resulting multidimensional region P_i contains either $|P_i| \geq k$ or $|P_i| = 0$ points and $C_{AVG} \leq$ positive constant c?

Our proof is based on a straightforward reduction from integer partitioning [7]:

Integer Partitioning Consider a set A of n positive integers $\{a_1, ..., a_n\}$. Is there some $A' \subseteq A$, such that

$$\sum_{a_i \in A'} a_i = \sum_{a_j \in A - A'} a_j \ ?$$

Theorem 1 *The k-anonymous strict multidimensional partitioning decision problem is NP-complete.*

Proof The proof is by reduction from integer partitioning. For each $a_i \in A$, construct $\{p_i\}$ containing a_i identical copies of the point $(0, ..., 0, 1_i, 0, ..., 0)$ (the i^{th} coordinate is 1, and all other coordinates are 0). Let multiset $P = \bigcup\{p_i\}$. P resides in an n-dimensional unit-hypercube.

We claim that the integer partitioning problem for A can be reduced to the following: *Let* $k = \frac{\sum a_i}{2}$. *Is there a k-anonymous strict multidimensional partitioning for* P *such that* $C_{AVG} \leq 1$*?* To prove this claim, we show that there is a solution to the k-anonymous multidimensional partitioning problem for P if and only if there is a solution to the integer partitioning problem for A.

Suppose there exists a k-anonymous multidimensional partitioning for P. This partitioning must define two multidimensional regions containing precisely $k = \frac{\sum a_i}{2}$ points each, and possibly some number of empty regions. By the strictness property, these regions must not overlap. Thus, the total number of points in each of the two non-empty regions constitute the sum of integers in two disjoint complementary subsets of A, and we have a partitioning of A.

In the other direction, suppose there is an integer partitioning of A. For each binary partitioning of A into disjoint complementary subsets A_1 and A_2, there is a multidimensional partitioning of P into regions $P_1, ..., P_m$ such that $|P_1| = \sum_{a_i \in A_1} a_i$, $|P_2| = \sum_{a_i \in A_2} a_i$, and all other P_i are empty: P_1 is defined by two points, the origin and the point p having i^{th} coordinate 1 when $a_i \in A_1$ and 0 otherwise. The bounding box for P_1 is closed at all edges and vertices. P_2 is defined by the origin and the point p having i^{th} coordinate $= 1$ when $a_i \in A_2$, and 0 otherwise. The bounding box for P_2 is open at the origin, but closed on all other edges and vertices. C_{AVG} is the average number of points in the non-empty regions, divided by k. In this construction, $C_{AVG} = 1$, and $P_1, ..., P_m$ is a k-anonymous multidimensional partitioning of P.

Finally, a given a solution to the k-anonymous multidimensional partitioning problem can be verified in polynomial time by simply scanning the input set of points P and maintaining a count for each region.□

2.3. Worst-Case Bound for Strict Multidimensional Partitioning

We define a *multidimensional cut* for a multiset P of points in d-dimensional space to be an axis-parallel binary cut producing two disjoint multisets of points.

Allowable Multidimensional Cut Consider multiset P of points in d-dimensional space. A cut perpendicular to dimension X_i at x_i is allowable if and only if $Count(P.X_i > x_i) \geq k$ and $Count(P.X_i \leq x_i) \geq k$.

[1] Although the following results are stated and proven for C_{AVG}, the construction is similar for C_{DM}.

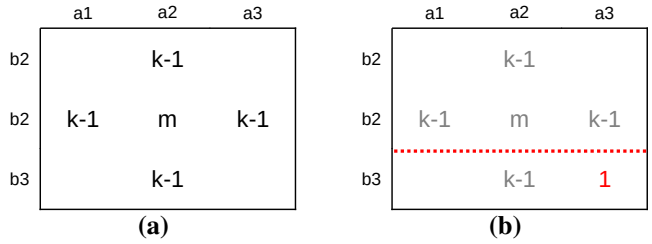

Figure 5. (a)A set of points in 2-dimensions for which there is no allowable cut, (b) Adding a single point produces an allowable cut.

Minimal Strict Multidimensional Partitioning Let $R_1, ..., R_n$ denote a set of regions induced by a strict multidimensional partitioning, and let each region R_i contain multiset P_i of points. This multidimensional partitioning is minimal if and only if, $\forall i$, $|P_i| \geq k$ and there exists no allowable multidimensional cut for P_i.

In this section, we prove that there is a worst-case upper-bound on the number of points contained in a region defined by a minimal multidimensional partitioning, independent of the total number of tuples in T. We represent T as a multiset of points P in d-dimensional space.

Lemma 1 *There exists a multiset P of points in d-dimensional space such that $|P| = 2d(k-1)+m$, where m is the maximum number of copies of any distinct point in P, and there is no allowable multidimensional cut for P.*

Proof Construct a multiset of points P such that $|P| = 2d(k-1)+m$, but there exists no allowable cut for P. Let $\widehat{x}_i$ denote some value on axis X_i such that $\widehat{x}_i - 1$ and $\widehat{x}_i + 1$ are also values on axis X_i, and let P initially contain m copies of the point $\langle \widehat{x}_1, \widehat{x}_2, ..., \widehat{x}_d \rangle$. Add to P $k-1$ copies each of the following points:

$(\widehat{x}_1 - 1, \widehat{x}_2, ..., \widehat{x}_d)$, $(\widehat{x}_1 + 1, \widehat{x}_2, ..., \widehat{x}_d)$,
$(\widehat{x}_1, \widehat{x}_2 - 1, ..., \widehat{x}_d)$, $(\widehat{x}_1, \widehat{x}_2 + 1, ..., \widehat{x}_d)$,
...
$(\widehat{x}_1, \widehat{x}_2, ..., \widehat{x}_d - 1)$, $(\widehat{x}_1, \widehat{x}_2, ..., \widehat{x}_d + 1)$

For example, Figure 5 shows P in 2 dimensions. By addition, $|P| = 2d(k-1)+m$, and by projecting P onto any X_i we obtain the following point counts:

$$Count(X_i) = \begin{cases} k-1, & X_i = \widehat{x}_i - 1 \\ m + 2(d-1)(k-1), & X_i = \widehat{x}_i \\ k-1, & X_i = \widehat{x}_i + 1 \\ 0, & \text{otherwise} \end{cases}$$

Based on these counts, it is clear that any binary cut perpendicular to axis X_i would result in some partition containing fewer than k points. □

Lemma 2 *For any multiset of points P in d-dimensional space such that $|P| > 2d(k-1)+m$, where m is the maximum number of copies of any distinct point in P, there exists an allowable multidimensional cut for P.*

Proof Consider an arbitrary P in d-dimensional space, such that $|P| = 2d(k-1)+m+1$, and let $\widehat{x}_i$ denote the median value of P projected on axis X_i. If there is no allowable cut for P, we claim that there exist at least $m+1$ copies of point $(\widehat{x}_1, \widehat{x}_d, ..., \widehat{x}_d)$ in P, contradicting the definition of m.

For every dimension $i = 1, ..., d$, if there is no allowable cut perpendicular to axis X_i, then $Count(X_i < \widehat{x}_i) \leq k-1$ and $Count(X_i > \widehat{x}_i) \leq k-1$. Then, $Count(X_i = \widehat{x}_i) \geq 2(d-1)(k-1)+m+1$. Thus, over d dimensions, we find $Count(X_1 = \widehat{x}_1 \wedge ... \wedge X_d = \widehat{x}_d) \geq m+1$. □

Theorem 2 *If $R_1, ..., R_n$ denotes the set of regions induced by a minimal strict multidimensional partitioning, the number of points contained in any R_i is no more than $2d(k-1)+m$.*

Proof The proof follows from Lemmas 1 and 2. □

Notice that recursive allowable d-dimensional cuts will result in a k-anonymous multidimensional partitioning for T. (Though not all possible multidimensional partitionings can be obtained in this way.) For example, Figure 4(c) shows a multidimensional partitioning. The first cut in this example occurs on the *Zipcode* dimension at 53711. Then, the left-hand side is cut again on the *Age* dimension, at 26. Both of these cuts are allowable because they do not produce any regions containing fewer than k points.

2.4. Single-dimensional Partitioning Bound

Single-dimensional partitioning can also be represented spatially, and a *single-dimensional cut* is also axis-parallel. However, we must consider all regions in the space when determining whether a cut is allowable.

Allowable Single-Dimensional Cut Consider a multiset P of points in d-dimensional space, and suppose we have already made S single-dimensional cuts, thereby separating the space into disjoint regions $R_1, ..., R_m$. A single-dimensional cut perpendicular to X_i at x_i is allowable, given S, if and only if $\forall R_j$ overlapping line $X_i = x_i$, $Count(R_j.X_i \leq x_i) \geq k$ and $Count(R_j.X_i > x_i) \geq k$.

Minimal Single-Dimensional Partitioning A set S of allowable single-dimensional cuts is a minimal single-dimensional partitioning for P if and only if there does not exist an allowable single-dimensional cut for P given S.

In the previous section, we showed that the worst-case upper bound on minimal partition size is independent of the total number of tuples in T under the strict multidimensional partitioning model. However, under single-dimensional partitioning, this worst-case upper bound can degrade linearly with the number of tuples in T.

Theorem 3 *The maximum size of a region R resulting from a minimal single-dimensional partitioning of a multiset of points P in d-dimensional space, for constant $d \geq 2$, can be $O(|P|)$.*

Proof We construct P and a minimal single-dimensional partitioning for P such that the greatest number of points in a resulting region is $O(|P|)$. For some quasi-identifier attribute X_i, consider finite set $V_{X_i} \subseteq D_{X_i}$ such that $|V_{X_i}| = m$, and let $\widehat{x}_i \in V_{X_i}$. Initially, let P contain precisely $2k-1$ points where $X_i = \widehat{x}_i$. Then add to P an arbitrarily large number of points, each defined by some tuple $(p_1, ..., p_d)$, where $p_i \in V_{X_i}$, but $p_i \neq \widehat{x}_i$, and such that there are at least k points in the resulting P having $p_i = x_i$ for each $x_i \in V_{X_i}$.

By construction, there are m allowable single-dimensional cuts for P perpendicular to X_i (at each point in V_{X_i}), and we denote this set of cuts S. However, there are no allowable single-dimensional cuts for P given S. Thus, S is a minimal single-dimensional partitioning, and the size of the largest resulting region is $O(|P|)$. □

By partitioning the space with allowable single-dimensional cuts, we obtain a k-anonymous single-dimensional partitioning for T. For example, Figure 4(b) shows the spatial representation of a single-dimensional partitioning. The first cut occurs on the *Zipcode* dimension at 53711. Once this cut has been made, there are no other allowable single-dimensional cuts. Notice that any cut perpendicular to the Age axis would result in a region on the right containing fewer than k points.

3. Multidimensional Local Recoding

In contrast to the global recoding models described in the previous section, *local recoding* models seek to achieve k-anonymity by mapping individual instances of data items to generalized values [17]. Formally, a local recoding function, which we will denote ϕ^* to distinguish it from global recoding functions, maps each (non-distinct) tuple $t \in T$ to some recoded tuple t'. Anonymization V is obtained by replacing each tuple $t \in T$ with $\phi^*(t)$. Several local recoding models have been considered in the literature, some of which are outlined in [9]. In this section, we describe one such model that relaxes the requirements of strict multidimensional partitioning.

Relaxed Multidimensional Partitioning A relaxed multidimensional partitioning for relation T defines a set of (potentially *overlapping*) distinct multidimensional regions that cover $D_{X_1} \times ... \times D_{X_d}$. Local recoding function ϕ^* maps each tuple $(x_1, ..., x_d) \in T$ to a summary statistic for *one* of the regions in which it is contained.

This relaxation offers flexibility. For example, consider generating a 3-anonymization of our Patients table, and suppose *Zipcode* is the single quasi-identifier attribute. Using the strict model, we would need to recode the Zipcode value in each tuple to [53710-53712]. However, under the relaxed model, this recoding can be performed on a tuple-by-tuple basis, and Figure 6 shows a possible anonymization.

Age	Sex	Zipcode	Disease
25	Male	[53710-53711]	Flu
25	Female	[53710-53711]	Hepatitis
26	Male	[53711-53712]	Brochitis
27	Male	[53710-53711]	Broken Arm
27	Female	[53711-53712]	AIDS
28	Male	[53711-53712]	Hang Nail

Figure 6. Relaxed partitioning for single quasi-identifier attribute Zipcode

Proposition 2 *Every strict multidimensional partitioning can be expressed as a relaxed multidimensional partitioning. However, when there are at least two tuples in table T having the same vector of quasi-identifier values, there exists a relaxed multidimensional partitioning that cannot be expressed as a strict multidimensional partitioning.*

Proof By definition, every strict partitioning is equivalent to some relaxed partitioning.

When there are at least at least two tuples, a and b, in T having the same vector of quasi-identifier values, $(x_1, ..., x_n)$, we can construct a relaxed multidimensional partitioning that is not equivalent to any strict multidimensional partitioning by constructing multidimensional regions P_1 and P_2 that overlap at $(x_1, ..., x_n)$ and ϕ^* that maps a to P_1 and b to P_2. □

This relaxed model can also be represented in d-dimensional space. However, a partitioning is not necessarily defined by binary cuts. Instead, a set of points is partitioned by defining two (possibly overlapping) multidimensional regions P_1 and P_2, and then mapping each point to either P_1 or P_2 (but not both). In this model, the upper-bound on partition size is $2k-1$.

Minimal Relaxed Multidimensional Partition We say that a multiset of points P is minimal according the the relaxed muldimensional partitioning model if and only if $|P| \geq k$ and it is not possible to divide P into two disjoint sets of points, P_1 and P_2, such that $|P_1| \geq k$ and $|P_2| \geq k$.

Proposition 3 *The maximum size of a minimal partition P, under the relaxed multidimensional partitioning model, is $2k-1$.*

Proof This follows from the definition of minimality.□

4. A Greedy Partitioning Algorithm

Using multidimensional partitioning, a k-anonymization is generated in two steps. In the first step, multidimensional regions are defined that cover the domain space, and in the

```
Anonymize(partition)
 if (no allowable multidimensional cut for partition)
  return partition
 else
  dim ← choose_dimension()
  fs ← frequency_set(partition, dim)
  splitVal ← find_median(fs)
  lhs ← {t ∈ partition : t.dim ≤ splitVal}
  rhs ← {t ∈ partition : t.dim > splitVal}
  Anonymize(rhs)
  Anonymize(lhs)
```

Figure 7. Top-down greedy algorithm for strict multidimensional partitioning

second step, recoding functions are constructed using summary statistics from each region. In the previous sections, we alluded to a recursive algorithm for the first step. In this section we outline a simple scalable algorithm, reminiscent of algorithms used to construct kd-trees [5], that can be adapted to either strict or relaxed partitioning.

The strict partitioning algorithm is shown in Figure 7. Each iteration must choose the dimension, as well as a value on this dimension, about which to partition. In the literature about kd-trees, one strategy used for obtaining uniform occupancy was median-partitioning [5]. In Figure 7, this means choosing a split value that is the median of $partition$ projected on dim. Like kd-tree construction, the time complexity is $O(nlogn)$, where n is the number of tuples in table T.

We have some flexibility in choosing the dimension on which to partition. As long as we make an allowable cut when one exists, this choice does not affect the partition-size upper-bound. However, a simple heuristic chooses the dimension with the widest (normalized) range of values [5], and we use this heuristic in our implementation. Alternatively, it may be possible to choose a dimension based on knowledge of an ancticipated workload.

Theorem 4 *The greedy median-partitioning algorithm for strict multidimensional partitioning generates a set of multidimensional regions, each containing at least k points, but no more than* $2d(k-1)+m$*, where* m *is the maximum number of copies of any distinct point.*

Proof Observe that the algorithm produces a strict multidimensional partitioning of the space, and each of the resulting regions contains at least k points. If there exists an allowable multidimensional cut for a partition P, perpendicular to X_i, then the cut perpendicular to X_i at the median is allowable. Given these observations, the upper bound on partition size follows from Theorem 2. □

The partitioning algorithm in Figure 7 is easily adapted for relaxed partitioning. Specifically, the points falling at the median (where $t.dim = splitVal$) are divided evenly between lhs_child and rhs_child such that $|lhs\_child| = |rhs\_child|$ ($+1$ when $|partition|$ is odd). In this case, there is a *2k-1* worst-case upper-bound on partition size.

Theorem 5 *The greedy median-partitioning algorithm for relaxed multidimensional partitioning produces a set of multidimensional regions, each containing at least k, but no more than* $2k-1$*, points.*

Proof This follows from Proposition 3. □

Following partitioning, the second step of the algorithm computes one or more summary statistics for the tuples contained in each region. A recoding function is then constructed, mapping each of the original tuples contained in a region to the summary statistics for that region. This process is described in more detail in Section 5.

4.1. Bounds on Quality

It is easy to show that the greedy partitioning algorithm produces anonymizations that are a constant factor approximation of optimal, as measured by the general-purpose metrics described in Section 1.2.

By definition, k-anonymity requires that every equivalence class contain at least k records. For this reason, the optimal achievable value of $C_{DM} \geq k*total\_records$, and the optimal value of $C_{AVG} \geq 1$.

Using our worst-case bounds on partition size, and assuming that the points in each distinct partition are mapped to a unique vector of summary statistics, we compute bounds for these cost metrics. Specifically, for strict multidimensional partitioning, the size of each partition $P \leq 2d(k-1)+m$. So, $C_{DM} \leq (2d(k-1)+m)*total\_records$ and $C_{AVG} \leq (2d(k-1)+m)/k$, where m is the maximum number of copies of any distinct point.

Similarly, for relaxed multidimensional partitioning the size of each partition $P \leq 2k$. So, $C_{DM} \leq 2k * total\_records$, and $C_{AVG} \leq 2$.

These simple observations have important implications. For constant d, it is guaranteed that the greedy algorithm for the strict model will generate an $O(k)$ approximation of the optimal solution. Further, the algorithm for relaxed partitioning results in a 2-approximation.

4.2. Scalability

When the table T to be anonymized is larger than the available memory, the main scalability issue to be addressed is finding the median value within a given partition of a selected attribute about which to make the next recursive cut.

We propose a solution to this problem based on the idea of a *frequency set*. The frequency set of attribute A for partition P is the set of unique values of A in P, each paired with an integer indicating the number of times it appears in

P. Given the frequency set of A for P, we can select the median value using a standard median-finding algorithm.

Because individual frequency sets contain just one entry per value in the domain of a particular attribute, and are much smaller than the size of the data itself, it is reasonable to assume that a single frequency set will fit in memory. For this reason, in the worst case, we must sequentially scan the database at most twice, and write once, per level of the recursive partitioning "tree." The data is first scanned once to find the median, and then scanned and written once to re-partition the data into two "runs" (lhs and rhs) on disk.

It is worth noting that in some cases the algorithm presented in Figure 7 could be further optimized to take advantage of available memory because, in practice, the frequency sets for multiple attributes may fit in memory.

5. Workload-Driven Quality Measurement

The general-purpose quality metrics in Section 1.2 are a good place to start when the ultimate use of the anonymized data is unknown. However, in some cases, the publisher may want to consider an anticipated workload, such as building a data mining model [6, 8, 16], or answering a set of aggregate queries. This section introduces the latter problem, including examples where the multidimensional model provides flexibility.

Consider generating an anonymization that is useful for answering a set of queries, drawn from the class of queries with a selection predicate (equality or range) of the form `attribute <oper> constant` and an aggregate function. Here we consider the most common aggregates (COUNT, SUM, AVG, MIN, and MAX). Our ability to answer these types of queries depends on two factors: the type of *summary statistic(s)* released for each attribute, and the degree to which the selection predicates in the workload *match* the range boundaries in the anonymous data.

The choice of *summary statistics* influences our ability to compute various aggregate functions.[2] In this paper, we consider releasing two summary statistics for each attribute A and equivalence class E:

- **Range statistic (R)** So far, all of our examples have considered a single summary statistic defined by the range of values for A appearing in E, which allows for easy computation of MIN and MAX aggregates.
- **Mean Statistic (M)** Now, we also consider a summary statistic defined by the mean value of A appearing in E, which allows for the computation of AVG and SUM aggregates.

When choosing which summary statistics to release, it is important to consider potential avenues for inference.

[2]Certain types of aggregate functions (e.g., MEDIAN) are ill-suited to this type of computation. We do not know of any way to compute such functions from these summary statistics.

Age(R)	Age(M)	Sex(R)	Zipcode(R)	Disease
$[25-26]$	25.5	Male	53711	Flu
$[25-27]$	26	Female	53712	Hepatitis
$[25-26]$	25.5	Male	53711	Brochitis
$[27-28]$	27.5	Male	$[53710-53711]$	Broken Arm
$[25-27]$	26	Female	53712	AIDS
$[27-28]$	27.5	Male	$[53710-53711]$	Hang Nail

Figure 8. A 2-anonymization with multiple summary statistics

Notice that in some cases simply releasing the minimum-maximum range allows for some inferences about the distribution of values within an equivalence class. For example, consider an attribute A, and let $k = 2$. Suppose that an equivalence class of the released anonymization contains two tuples, and A is summarized by the range $[0-1]$. It is easy to infer that in one of the original tuples $A = 0$, and in the other $A = 1$.

The presence of this type of inference is not likely to represent a problem in preventing joining attacks because, without background knowledge, it is still impossible for an adversary to distinguish the tuples within an equivalence class from one another, even if the adversary has information about their distribution. This type of inference may also arise in single-dimensional partitioning models. Nonetheless, it is an important issue to be aware of when designing an anonymization scheme.

The second factor influencing our ability to answer aggregate queries is the degree to which the selection predicates in the given workload "match" the boundaries of the range statistics in the released anonymization. In many ways, this is analogous to matching indices and selection predicates in traditional query processing.

Predicate-Range Matching If a query contains a selection predicate P, P conceptually divides the original table T into two sets of tuples, T^T and T^F (those that satisfy the predicate and those that do not). When range statistics are published, we say that an anonymization V *matches* a boolean predicate P if every tuple $t \in T^T$ is mapped to an equivalence class in V containing no tuples from T^F.

As a simple example illustrating these two ideas, consider a workload consisting of two queries:

```
SELECT AVG(Age)
FROM Patients
WHERE Sex = 'Male'
```

```
SELECT COUNT(*)
FROM Patients
WHERE Sex = 'Male'
AND Age ≤ 26
```

A strict multidimensional anonymization of Patients is given in Figure 8, including two summary statistics (range and median) for the Age attribute. Notice that the mean statistic allows us to answer the first query precisely and accurately. The second query can also be answered precisely because the range referenced by the predicate matches a single

Distribution	(Discrete Uniform, Discrete Skewed)
Attributes	Total quasi-identifier attributes
Cardinality	Distinct values per attribute
Tuples	Total tuples in table
Std. Dev. (σ)	With respect to standard normal (Skewed only)
Mean (μ)	(Skewed only)

Figure 9. Parameters of synthetic generator

equivalence class in the anonymization, which contains exactly 2 tuples. Comparing this with the single-dimensional recoding shown in Figure 2, notice that it would be impossible to answer the second query precisely using the single-dimensional recoding.

When a workload consists of many queries, even a multidimensional anonymization might not match every selection predicate. An exhaustive discussion of query processing over imprecise data is beyond the scope of this paper. However, one approach is to assume a uniform distribution of values for each attribute within each equivalence class, and compute the aggregate function based on this assumption. The effects of multidimensional versus single-dimensional recoding, with respect to a specific query workload, are explored empirically in Section 6.3.

Our work on workload-driven anonymization is preliminary, and there are a number of important future directions. One of the most important directions is directly integrating knowledge of an anticipated workload into the anonymization algorithm. Formally, a query workload can be expressed as a set of *(multdimensional region, aggregate, weight)* triples, where the boundaries of each region are determined by the selection predicates in the workload. Each query is also assigned a weight indicating its importance with respect to the rest of the workload. When a selection predicate in the workload does not exactly match the boundaries of one or more equivalence classes, evaluating this query over the anonymized data will incur some *error*. This error can be defined as the normalized difference between the result of evaluating the query on the anonymous data, and the result on the original data. Intuitively, the task of a workload-driven algorithm is to generate an anonymization that minimizes the weighted sum of such errors.

6. Experimental Evaluation

The main goal of our experiments was to evaluate the quality of the anonymizations produced by our greedy algorithm for multidimensional partitioning. In particular, we compared these anonymizations with those produced by optimal algorithms for two other models: full-domain generalization [9, 12], and single-dimensional partitioning [2, 8]. The specific algorithms used in the comparison were Incognito [9] and K-Optimize [2], respectively. We chose these algorithms for efficiency, but any exhaustive algorithm for these models would yield the same result. From a performance

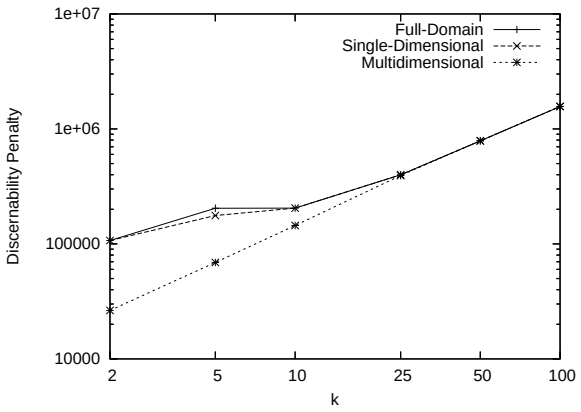

Figure 10. Anonymization quality on 5-attribute uniform distribution

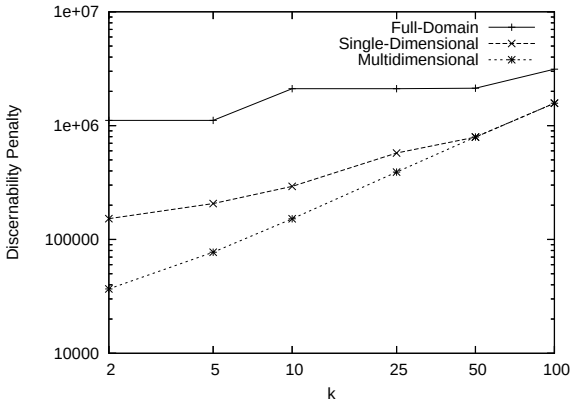

Figure 11. Anonymization quality on 5-attribute discrete skewed distribution ($\sigma = .2$)

perspective, these experiments do not represent a fair comparison because the exhaustive algorithms are exponential in the worst case, and they run many times slower than our greedy algorithm. Nonetheless, the quality of the results obtained by the latter is often superior.

For these experiments, we used both synthetic and real-world data. We compared quality, using general-purpose quality metrics described in Section 1.2, and also with respect to a simple query workload, as outlined in Section 5.

6.1. Experimental Data

For some experiments, we used a synthetic data generator, which produced two discrete joint distributions: *discrete uniform* and *discrete skewed*. We limited the evaluation to discrete distributions so that the exhaustive algorithms would be tractable without pre-generalizing the data. To generate the discrete skewed distribution, we first generated the multivariate normal distribution, and then discretized the values of each attribute into equal-width ranges. The parameters are described in Figure 9.

In addition to synthetic data, we also used the Adults

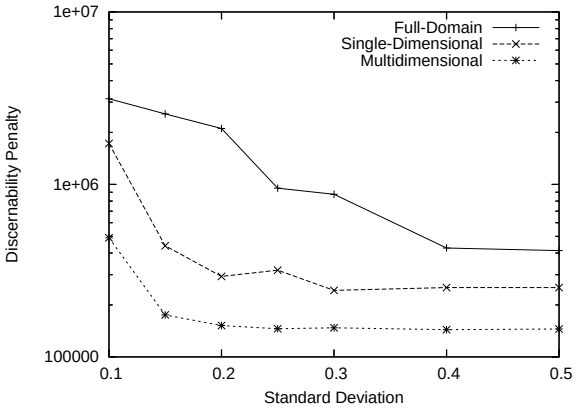

Figure 12. Anonymization quality on 5-attribute skewed distribution ($k = 10$)

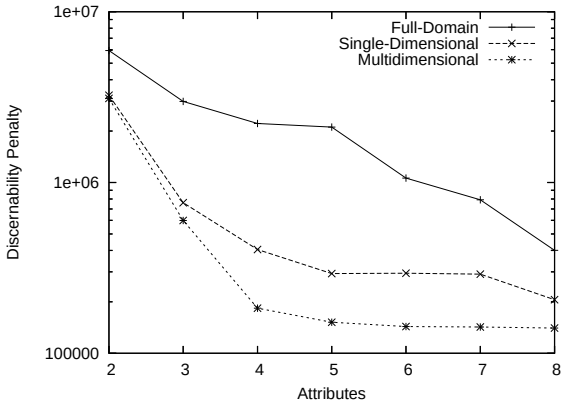

Figure 13. Anonymization quality on discrete skewed distribution ($k = 10, \sigma = .2$)

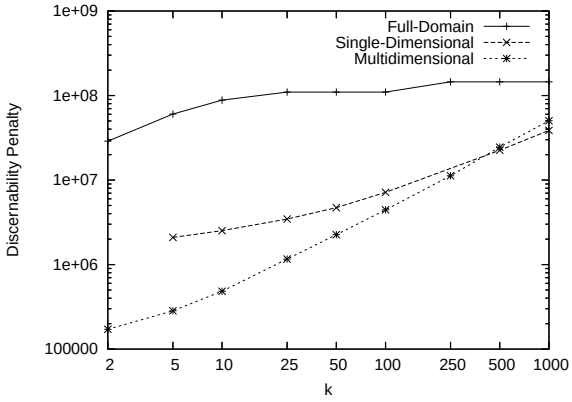

Figure 14. Anonymization quality on Adults database

database from the UC Irvine Machine Learning Repository [3], which contains data from the US Census and has become a de facto benchmark for k-anonymity. We configured this data set as it was configured for the experiments reported in [2], using eight regular attributes, and removing tuples with missing values. The resulting database consisted of 30,162 records. For the (single- and multidimensional) partitioning experiments, we imposed an intuitive ordering on each attribute, but unlike [2], we eliminated all hierarchical constraints for both models. For the full-domain experiments, we used the same generalization hierarchies that were used in [9].

6.2. General-Purpose Metrics

In this section, we describe some simple experiments that use general-purpose metrics to compare the quality of anonymizations generated by greedy (strict) multidimensional partitioning with those resulting from optimal algorithms for single-dimensional partitioning and full-domain generalization. Here we report results for the discernability metric [2], but the comparisons are similar for the average equivalence class size metric.

The first experiment compared the three models for varied values of k. We fixed the number of tuples at 10,000, the per-attribute cardinality at 8, and the number of attributes at 5. For the full-domain generalization model, we constructed generalization hierarchies using binary trees. The results for the Uniform distribution are shown in Figure 10. Results for the Discrete Skewed distribution ($\mu = 3.5, \sigma = .2$) are given in Figure 11. We found that greedy multidimensional partitioning produced "better" generalizations than the other algorithms in both cases. However, the magnitude of this difference was much more pronounced for the skewed distribution.

Following this observation, the second experiment compared quality using the same three models, but varied the standard deviation (σ) of the synthetic data. (Small values of σ indicate a high degree of skew.) The number of attributes was again fixed at 5, and k was fixed at 10. The results (Figure 12) show that the difference in quality between multidimensional and the others is most pronounced for highly-skewed distributions.

The next experiment measured quality for varied quasi-identifier size, with $\sigma = .2$ and $k = 10$. As the number of attributes increases, the observed discernability penalty decreases for each of the three models (Figure 13). At first glance, this result is counter-intuitive. However, this decrease is due to the sparsity of the original data, which contains fewer duplicate tuples as the number of attributes increases.

In addition to the synthetic data, we compared the three algorithms using the Adults database (Figure 14). Again, we found that greedy multidimensional partitioning produced the best results. This difference is most pronounced for small k; as k increases, the results become comparable.

6.3. Workload-Based Quality

We also ran several experiments to compare the single- and multidimensional partitioning models with respect to a sim-

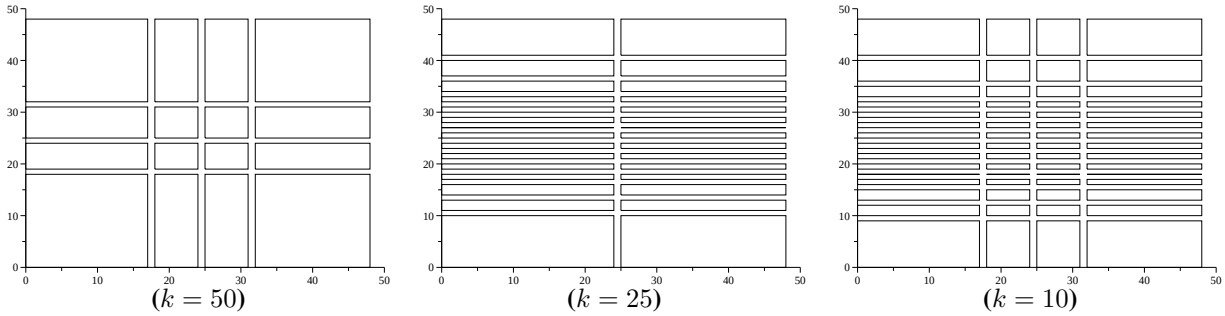

Figure 15. Optimal single-dimensional partitioning for two quasi-identifier attributes with a discrete skewed distribution ($\mu = 25$, $\sigma = .2$)

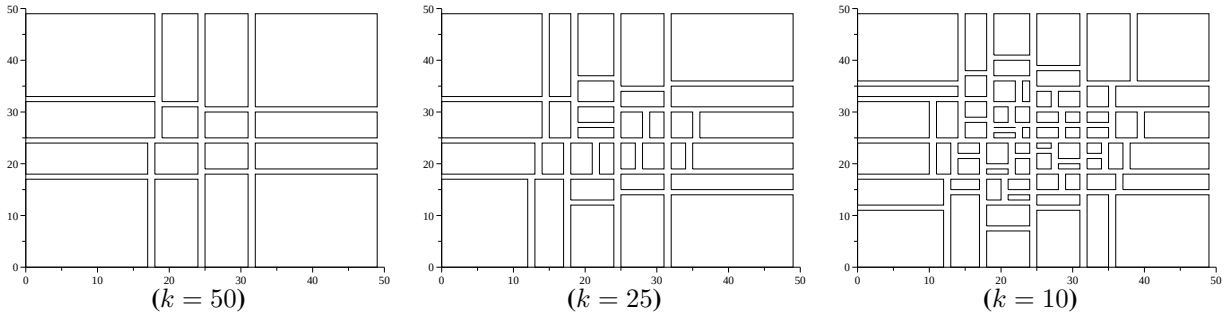

Figure 16. Strict multidimensional partitioning for two quasi-identifier attributes with a discrete skewed distribution ($\mu = 25$, $\sigma = .2$).

Predicate on X

k	Model	Mean Error	Std. Dev.
10	Single	7.73	5.94
10	Multi	4.66	3.26
25	Single	12.68	7.17
25	Multi	5.69	3.86
50	Single	7.73	5.94
50	Multi	7.94	5.87

Predicate on Y

k	Model	Mean Error	Std. Dev.
10	Single	3.18	2.56
10	Multi	4.03	3.44
25	Single	5.06	4.17
25	Multi	5.67	3.80
50	Single	8.25	6.15
50	Multi	8.06	5.58

Figure 17. Error for count queries with single-attribute selection predicates

ple query workload. We considered a synthetic data set containing 1000 tuples, with two quasi-identifier attributes (discrete skewed, each with cardinality 50, $\mu = 25$, $\sigma = .2$), and we generated k-anonymizations using the greedy multidimensional partitioning and optimal single-dimensional partitioning algorithms. Visual representations of the resulting partitionings are given in Figures 16 and 15.

The multidimensional partitioning does an excellent job at capturing the underlying multivariate distribution. In contrast, we observed that for small k, single-dimensional partitioning tends to reflect the distribution of just one attribute. However, the optimal single-dimensional anonymization is quite sensitive to the underlying data, and a small change to the synthetic data set often changes the resulting anonymization.

This potential "linearization" of attributes has an impact on query processing over the anonymized data. We considered a simple workload for this two-attribute data set, consisting of queries of the form "SELECT COUNT(*) WHERE $\{X, Y\}$ = $value$", where X and Y are the two quasi-identifier attributes, and $value$ consists of integers between 0 and 49. (In Figures 15 and 16, X and Y are displayed on the horizontal and vertical axes.) We evaluated these queries over each anonymization, as well as the original data set. On the anonymized data, when a predicate did not match any partition, we assumed a uniform distribution within each partition.

For each anonymization, we computed the mean and standard deviation of the absolute error over the set of queries in the workload. These results are presented in Figure 17. As is apparent from Figures 15 and 16, and from the error measurements, queries with predicates on Y are more accurately answered from the single-dimensional anonymization than are queries with predicates on X. The observed error is more consistent across queries using the multidimensional anonymization.

7. Related Work

The bulk of previous work on k-anonymity has involved user-defined value generalization hierarchies [6, 8, 9, 12,

14, 16]. Recently, partitioning models have been proposed to automatically generate generalization hierarchies[2, 8]. Such models are particularly well-suited for continuous or numeric data, but all of the previously-proposed anonymization techniques have been single-dimensional.

User-defined hierarchies impose additional constraints over partitioning models. For numeric values, these constraints may be unnecessary and reduce flexibility. However, for other types of data (e.g., categorical), user-defined hierarchies may lead to more intuitive anonymizations.

Another simpler model of anonymization has also been considered in the literature, and this model considers suppressing individual cells in a relation in order to achieve k-anonymity. Approximation algorithms have also been proposed for the problem of finding the k-anonymization suppressing the fewest cells [1, 10].

In other related work, private histograms have also been proposed for data publishing [4]. Chawla et al [4] present an algorithm that also considers the domain of sensitive attributes in a multidimensional space. However, it does not view minimal partition-occupancy to be an absolute constraint, and for this reason, the resulting partitions may contain fewer than k points.

8. Conclusion and Future Work

In this paper, we introduced a multidimensional recoding model for k-anonymity. Although the problem of finding the optimal anonymization is NP-hard, we provide a simple, scalable, and efficient greedy constant-factor approximation algorithm for several general-purpose quality metrics. An experimental evaluation indicates that often the results of this algorithm are actually *better* than those produced by more expensive optimal algorithms using other recoding models.

The second main contribution of this paper is a more targeted notion of quality measurement, based on a workload of aggregate queries. The second part of our experimental evaluation showed that, for workloads involving predicates on multiple attributes, the multidimensional recoding model often leads to more desirable results.

There are a number of promising areas for future work. In particular, we would like to extend the greedy algorithm presented in this paper to multidimensional models involving user-defined hierarchies. Also, as mentioned in Section 5, we are considering ways of integrating knowledge of an anticipated query workload directly into the anonymization algorithms. Finally, we suspect that multidimensional recoding would lend itself to creating anonymizations that are useful for building data mining modes, such as decision trees [6, 8, 16].

References

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *Proc. of the 10th Int'l Conf. on Database Theory*, January 2005.

[2] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymity. In *Proc. of the 21st Int'l Conf. on Data Engineering*, April 2005.

[3] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[4] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Proc. of the 2nd Theory of Cryptography Conf.*, February 2005.

[5] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic time. *ACM Trans. on Mathematical Software*, 3(3), September 1977.

[6] B. Fung, K. Wang, and P. Yu. Top-down specialization for information and privacy preservation. In *Proc. of the 21st Int'l Conf. on Data Engineering*, April 2005.

[7] M. Garey and D. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman, 1979.

[8] V. Iyengar. Transforming data to satisfy privacy constraints. In *Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, August 2002.

[9] K. LeFevre, D.DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, June 2005.

[10] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proc. of the 23rd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, June 2004.

[11] S. Muthakrishnan, V. Poosala, and T. Suel. On rectangular partitionings in two dimensions: Algorithms, complexity, and applications. In *Proc. of the 7th Int'l Conf. on Database Theory*, 1998.

[12] P. Samarati. Protecting respondants' identities in microdata release. *IEEE Trans. on Knowledge and Data Engineering*, 13(6), November/December 2001.

[13] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.

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

[15] L. Sweeney. K-anonymity: A model for protecting privacy. *Int'l Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5):557–570, 2002.

[16] K. Wang, P. Yu, and S. Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *Proc. of the 4th IEEE Int'l Conf. on Data Mining*, November 2004.

[17] L. Willenborg and T. deWaal. *Elements of Statistical Disclosure Control*. Springer Verlag, 2000.

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