Achieving Multidimensional K-Anonymity by a Greedy Approach

G.Narasimha Murthy (M.Tech),
Sri Sai Aditya Institute of Science And Technology,JNTUK,Kakinada.

R.Srinivas M.Tech(PhD),
HOD,Department Of Computer science
Sri Sai Aditya Institute of Science And Technology,JNTUK,Kakinada.

ABSTRACT
Protecting privacy in microdata publishing is K-Anonymity, Here recoding "models" have been considered for achieving k anonymity[1,2]. We proposes a new multidimensional model, which gives high flexibility. Often this flexibility leads to higher-quality anonymizations, as measured both by general-purpose metrics and more specific notions of query answerability. Like previous multidimensional models anonymization is NP-hard. However, we introduce a simple greedy approximation algorithm, It leads to more desirable anonymizations than single-dimensional models.

1. INTRODUCTION
Many organizations publish microdata for research[6]. To protect individual privacy, known identifiers (e.g., Name and Social Security Number) must be removed[7]. In addition, this process must account for the possibility of combining certain other attributes with external data to uniquely identify individuals. For example, an individual might be "re-identified" by joining the released data with another (public) database on Age, Sex, and Zipcode. Table 1 shows such an attack, where Lee's medical information is determined by joining the released patient data with a public voter registration list.

To reduce the risk of this type of attack by k anonymity. The primary goal of k-anonymization is to protect the privacy of the individuals to whom the data pertains[1]. It is important that the released data remain as "useful" as possible. Numerous recoding *models* have been proposed in the literature for k-anonymization, and often the "quality" of the published data is dictated by the model that is used.

1.1 Basic Definitions
Quasi-Identifier Attribute Set A quasi-identifier is a minimal set of attributes*X*1, ...,*Xd* in table *T* that can be joined with external information to re-identify individual records.

Equivalence Class A table *T* consists of a multiset of tuples. An equivalence class for *T* with respect to attributes *X*1, ...,*Xd* is the set of all tuples in *T* containing identical values (*x*1, ..., *xd*) for *X*1, ...,*Xd*. In SQL, this is like a GROUP BY query on *X*1, ...,*Xd*.

Table 1. A joining attack
Voter Registration data

Name	Age	Sex	Zipcode
Lee	35	Male	533449
zan	38	Female	533448
dum	41	Female	533447
wry	29	Male	533446
su	51	Female	533445

Patient data

Age	Sex	Zipcode	Disease
35	Male	533449	Flu
35	Female	533441	Cancer
36	Male	533449	Aids
37	Male	533448	Hang Nail
37	Female	533448	Hepatites
38	Male	533449	Broken harm

K-Anonymity Property Table *T* is *k-anonymous* with respect to attributes *X*1, ...,*Xd* if every unique tuple (*x*1, ..., *xd*) in the (multiset) projection of *T* on *X*1, ...,*Xd* occurs at least *k* times. That is, the size of each equivalence class in *T* with respect to *X*1, ...,*Xd* is at least *k*.

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

2. OVERVIEW OF THE CONCEPTS
2.1 General-Purpose Quality Metrics
There are a number of notions of microdata quality , but intuitively, the anonymization process should generalize or perturb the original data as little as is necessary to satisfy the k-anonymity constraint[3]. Here we consider some simple general-purpose quality metrics, but a more targeted approach to quality measurement based on query answerability is described later.
The simplest kind of quality measure is based on the size of the equivalence classes *E* in *V* . Intuitively, the discern ability metric (*CDM*), described, assigns to each tuple *t* in *V* a penalty, which is determined by the size of the equivalence class containing *t*.
$CDM = \sum \text{EquivClasses } E\ |E|2$
As an alternative, we also propose the normalized average equivalence class size metric (CAV G).

$$CAV\ G = \frac{(\text{total records})}{(\text{total equiv classes})/(k)}$$

2.2 Multidimensional Global Recoding
Each attribute has some domain of values. We use the notation *DX* to denote the domain of attribute *X*. A global

recoding achieves anonymity by mapping the domains of the quasi-identifier attributes to generalized or altered values .

Global recoding can be further broken down into two sub-classes. A single-dimensional global recoding is defined by a function φi : DXi → D'for each attribute Xi of the quasi-identifier. An anonymization *V* is obtained by applying each *φi* to the values of *Xi* in each tuple of *T*.

A multidimensional global recoding is defined by a *single* function φ : DX1 × ... × DXn → D', which is used to recode the domain of value *vectors* associated with the set of quasi-identifier attributes. Under this model, *V* is obtained by applying *φ* to the vector of quasi identifier values in each tuple of *T*.

Multidimensional recoding can be applied to categorical data (in the presence of user-defined generalization hierarchies) or to numeric data. For numeric data, and other totally-ordered domains, (single-dimensional) "partitioning" models have been proposed . A *single dimensional interval* is defined by a pair of endpoints *p, v*€ *DXi* such that *p* ≤ *v*. (The endpoints of such an interval may be open or closed to handle continuous domains.)

Single-dimensional Partitioning Assume there is a total order associated with the domain of each quasi-identifier attribute *Xi*. A single-dimensional partitioning defines, for each *Xi*, a set of *non-overlapping* single-dimensional intervals that cover *DXi* . *φi* maps each *x* € *DXi* 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 later section .

This partitioning model is easily extended to multidimensional recoding. Again, assume a total order for each *DXi*. A *multidimensional region* is defined by a pair of *d*tuples (*p*1, ..., *pd*), (*v*1, ..., *vd*) €*DX*1 ×...×*DXd* such that ¥*i*, *pi* ≤ *vi*. 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.

Table 2 . Single dimensional anonymization

Age	Sex	Zipcode	Disease
[35-38]	Male	[533448-533449]	Flu
[35-38]	Female	533441	Cancer
[35-38]	Male	[533448-533449]	Aids
[35-38]	Male	[533448-533449]	Hang Nail
[35-38]	Female	533449	Hepatites
[35-38]	Male	[533448-533449]	Broken harm

Table 3. Multidimensional anonymization

Age	Sex	Zipcode	Disease
[35-36]	Male	53711	Flu
[35-37]	Female	53712	Cancer
[35-36]	Male	53711	Aids
[37-38]	Male	[53710-53711]	Hang Nail
[35-37]	Female	53712	Hepatites
[37-38]	Male	[53710-53711]	Broken harm

Strict Multidimensional Partitioning A strict multidimensional partitioning defines a set of non-overlapping multidimensional regions that cover *DX*1 × ... × *DXd* . *φ* maps each tuple (*x*1, ..., *xd*) € *DX*1 × ... × *DXd* 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), the tuple set in each non-empty region forms an equivalence class in *V* . For simplicity, we again assume that these summary statistics are ranges, and further discussion is provided in later section .

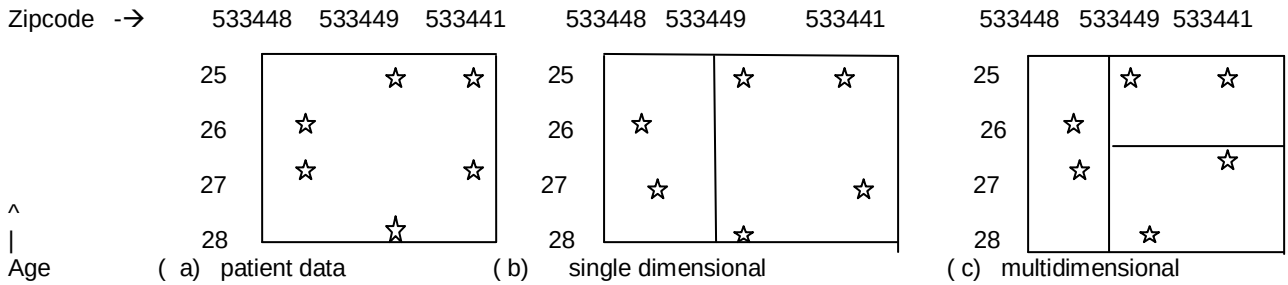
Notice that the anonymization obtained using the multidimensional model is not permissible under the single-dimensional model because the domains of Age and Zipcode are not recoded to a single set of intervals (e.g., Age 35 is mapped to either [35-36] or [35-37], depending on the values of Zipcode and Sex). However, the single-dimensional recoding is also valid under the multidimensional model.

2.2.1 Spatial Representation

We use a convenient spatial representation for quasi-identifiers. Consider table *T* with quasi identifier attributes *X*1, ...,*Xd*, and assume that there is a total ordering for each domain. The (multiset) projections of *T* on *X*1, ...,*Xd* can then be represented as a multiset of points in *d*-dimensional space. For example, Figure 1(a) shows the two-dimensional representation of Patients from table 1, for quasi-identifier attributes *Age* and *Zipcode*.

Similar models have been considered for rectangular partitioning in 2 dimensions. In this context, the single dimensional and multidimensional partitioning models are analogous to the "*p***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 minimum occupancy.

Figure 1. Spatial representation of Patients and partitionings (quasi-identifiers Zipcode and Age)



Bounds on Partition Size

It is also interesting to consider worst-case upper bounds on the size of partitions resulting from single-dimensional and multidimensional partitioning. This section presents two results, the first of which indicates that for a constant-sized quasi-identifier, this upper bound depends only on *k* and the maximum number of duplicate copies of a single point.

This is in contrast to the second, which indicates that for single-dimensional partitioning, this bound may grow linearly with the total number of points.

In order to state these results, we first define some terminology. A *multidimensional cut* for a multiset of points is an axis-parallel binary cut producing two disjoint multisets of points. Intuitively, such a cut is allowable if it does not cause a violation of k-anonymity.

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

A *single-dimensional cut* is also axis-parallel, but considers all regions in the space to determine allowability. 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, ...,*Rm*. A singledimensional cut perpendicular to *Xi* at *xi* is allowable,given *S*, if *Rj* overlapping line *Xi* = *xi*, $Count(Rj.Xi > xi) \geq k$ and $Count(Rj.Xi \leq xi) \geq k$.

Notice that recursive allowable multidimensional cuts will result in a *k*-anonymous strict multidimensional partitioning for *P* (although not all strict multidimensional partitionings can be obtained in this way), and a *k*-anonymous single-dimensional partitioning for *P* is obtained through successive allowable single-dimensional cuts.

For example, in Figures 1(b) and (c), the first cut occurs on the *Zipcode* dimension at 533449. In the multidimensional case, the left-hand side is cut again on the *Age* dimension, which is allowable because it does not produce a region containing fewer than *k* points. In the single dimensional case, however, once the first cut is made, there are no remaining allowable single-dimensional cuts. (Any cut perpendicular to the Age axis would result in a region on the right containing fewer than *k* points.) Intuitively, a partitioning is considered minimal when there are no remaining allowable cuts.

2.3 A Greedy Partitioning Algorithm

A k-anonymization is generated Using multidimensional partitioning in two steps. In the first step, multidimensional regions are defined that cover the domain space, and in the second step, recoding functions are constructed using summary statistics from each region reminiscent of those used to construct *kd*-trees , that can be adapted to either strict or relaxed partitioning. The strict partitioning algorithm is shown in Figure 2. Each iteration must choose the dimension and value about which to partition. In the *kd*-trees,

Figure 2. Top-down greedy algorithm for strict multidimensional partitioning

Anonymize(*partition*)
if (no allowable multidimensional cut for *partition*)
return φ : *partition* → *summary*
else
dim ←choose dimension()
fs ←frequency set(*partition*, *dim*)
splitV al ←find median(*fs*)
lhs ←{*t* ∈ *partition* : *t.dim* ≤ *splitV al*}
rhs ←{*t* ∈ *partition* : *t.dim* > *splitV al*} return Anonymize(*rhs*)
ụ Anonymize(*lhs*) one strategy used for obtaining uniform occupancy was median partitioning. In Figure 2, the split value is the median of *partition* projected on *dim*. Like *kd*-tree construction, the time complexity is $O(nlogn)$, where $n = |T|$. If there exists an allowable multidimensional cut for partition *P* perpendicular to some axis *Xi*, then the cut perpendicular to *Xi* at the median is allowable. The greedy (strict) median-partitioning algorithm results in a set of multidimensional regions, each containing between *k* and $2d(k-1)+m$ points , where *m* is the maximum number of copies of any distinct point.

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. One heuristic, used in our implementation, chooses the dimension with the widest (normalized) range of values. Alternatively, it may be possible to choose a dimension based on an anticipated workload.

The partitioning algorithm in Figure 1 is easily adapted for relaxed partitioning. Specifically, the points falling at the median (where *t.dim = split Val*) 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 $2k - 1$ upper-bound on partition size.Finally, a similar greedy multidimensional partitioning strategy can be used for categorical attributes in the presence

of user-defined generalization hierarchies. However, our quality upper-bounds do not hold in this case.

2.3.1 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 of a selected attribute within a given partition.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*, the median value is found 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.

2.4 Workload-Driven Quality Measurement

Here we want to consider an anticipated workload, such as building a data mining model, or answering a set of aggregate queries. This section introduces the latter problem, including examples where multidimensional recoding provides needed flexibility.

Consider a set of queries with selection predicates (equality or range) of the form attribute <oper> constant and an aggregate function (COUNT, SUM, AVG, MIN, and MAX). Our ability to answer this type of queries from anonymized data depends on two main 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. we consider releasing two summary statistics for each attribute *A* and equivalence class *E*:

• Range statistic (R) So far, all of our examples have involved 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) 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.

When choosing summary statistics, it is important to consider potential avenues for inference. 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 $A = 1$. This type of inference about distribution (which may also arise in single-dimensional recoding) is not likely to pose a problem in preventing joining attacks because, without background knowledge, it is still m possible for an adversary to distinguish the tuples within an equivalence class from one another.

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 A selection predicate *Pred* conceptually divides the original table *T* into two sets of tuples, *TT Pred* and *TF Pred* (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 *Pred* if every tuple $t \varepsilon$ *TT Pred* is mapped to an equivalence class in *V* containing no tuples from *TF Pred*.

To illustrate these ideas, consider a workload containing two queries:

SELECT AVG (Age) FROM Patients WHER Sex = 'Male'.

SELECT COUNT (*) FROM Patients WHERE Sex = 'Male' AND Age ≤ 36.

Table 4. A 2-anonymization with multiple summary statistics

Age(R)	Age(M)	Age	Zipcode	Disease
[35-36]	35.5	Male	533449	Flu
[35-37]	36	Female	533441	Cancer
[35-36]	35.5	Male	533449	Aids
[37-38]	36	Male	[533448-533449]	Hnag Nail
[35-37]	36	Female	533441	Hepatites

A strict multidimensional anonymization of Patients is including two summary statistics (range and mean) for Age. Notice that the mean allows us to answer the first query precisely and accurately. The second query can also be answered precisely because the predicate matches a single equivalence class in the anonymization. Comparing this with the single- dimensional recoding shown in Table 3, 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, when no additional distribution information is available, a simple approach assumes a uniform distribution of values for each attribute within each equivalence class.

Our work on workload-driven anonymization is preliminary, and in this paper, the workload is primarily used as an evaluation tool. One of the most important future directions is directly integrating knowledge of an anticipated workload into the anonymization algorithm. Formally, a query workload can be expressed as a set of *(multidimensional 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.

5 CONCLUSION

Achieving Multidimentional k-anonymity by introducing a multidimensional recoding model and provide a simple and efficient greedy approximation algorithm. And Here Uses several general purpose quality metrics, The more targeted notion of this paper is quality measurement, based on a workload of aggregate queries. The second part describes for workloads involving predicates on multiple attributes, the multidimensional recoding model often produces more desirable results. Although optimal multidimensional partitioning is NP-hard. There are a number of promising areas for future work.

6 REFERENCES

[1] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE*, 2005.

[2] B. Fung, K. Wang, and P. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, 2005.

[3] V. Iyengar. Transforming data to satisfy privacy constraints. In *ACM SIGKDD*, 2002.

[4] K. LeFevre, D.DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACMSIGMOD*, 2005.

[5] A. Meyerson and R.Williams. On the complexity of optimal k-anonymity. In *PODS*, 2004.

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

[7] 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.

[8] 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.

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

[10] K. Wang, P. Yu, and S. Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *ICDM*, 2004.

[11] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, 2005.

[12] S. Muthakrishnan, V. Poosala, and T. Suel. On rectangular partitionings in two dimensions: Algorithms, complexity, and applications. In *ICDT*, 1998.