

On the Complexity of Optimal K-Anonymity

Adam Meyerson^{*}
Department of Computer Science
University of California
Los Angeles, CA
awm@cs.ucla.edu

Ryan Williams[†]
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA
ryanw@cs.cmu.edu

ABSTRACT

The technique of *k-anonymization* has been proposed in the literature as an alternative way to release public information, while ensuring both data privacy and data integrity. We prove that two general versions of optimal *k-anonymization* of relations are *NP*-hard, including the suppression version which amounts to choosing a minimum number of entries to delete from the relation. We also present a polynomial time algorithm for optimal *k-anonymity* that achieves an approximation ratio independent of the size of the database, when *k* is constant. In particular, it is a $O(k \log k)$ -approximation where the constant in the big- O is no more than 4. However, the runtime of the algorithm is exponential in *k*. A slightly more clever algorithm removes this condition, but is a $O(k \log m)$ -approximation, where *m* is the degree of the relation. We believe this algorithm could potentially be quite fast in practice.

1. INTRODUCTION

Data privacy and data mining are quite naturally at odds with each other. In certain scenarios such as tracking epidemics and product marketing, a data miner requires access to large volumes of (possibly) personal information, in order to spot interesting trends or correlations. However, the direct release and study of such data may violate the privacy of individuals. Ideally, one strives for the best of both worlds: to somehow infer overall trends in data without inferring much information about particulars. Many approaches to this fundamental data mining problem have been suggested, implemented, and theoretically studied (*e.g.* [1, 2, 3, 5, 7, 4]). In general, most proposals for privacy-protecting data mining involve perturbing individual data values or perturb-

ing the results of queries, but this is undesirable if one wants to ensure complete data integrity. (One can imagine cases arising where it is vital to be capable of rigorously proving to a judge that a certain trend is indeed occurring; with perturbations in data, only “probably true” inferences may be drawn.)

An alternative approach is to restrict the release of information in some way. We shall focus on the strategy of *k-anonymization*, first proposed by Samarati and Sweeney [9, 10]. To our knowledge, no theoretical results were known concerning the problem prior to this paper. Let $k > 1$ be a fixed integer. Suppose we want to release a table of private data to the public, and we have the capability to suppress or “generalize” various entries in the table. If this suppression/generalization is done in such a way that every record becomes textually indistinguishable (entry for entry) from $k - 1$ other records in the table, we say that the new modified table is *k-anonymized*. For a quick example, take the following short relation, given as a possible response to the query “Who had an X-ray at this hospital yesterday?”

first	last	age	race
Harry	Stone	34	Afr-Am
John	Reyser	36	Cauc
Beatrice	Stone	47	Afr-Am
John	Ramos	22	Hisp

Suppose our task is to 2-anonymize this data before its release. If the database has been augmented to permit the proper values for attributes, then one possible 2-anonymization of the table would be the following.

first	last	age	race
*	Stone	30-50	Afr-Am
John	R*	20-40	*
*	Stone	30-50	Afr-Am
John	R*	20-40	*

(Note the specification of “20-40”, “R*”, *etc.* as admissible generalizations must be given prior to the input. In this paper, we will only consider the special case of suppressions, *i.e.* each entry is either included in the output, or omitted entirely, with a * character taking its place.)

We will see that *k-anonymity* admits a very clean formalization; it is simple to propose, and has a concrete privacy parameter *k* within its definition. In this work, we will consider the complexity of rendering relations of private records *k-anonymous*, while *minimizing* the amount of information that is not released. That is, we want to simultaneously ensure the anonymity of individuals up to a group of size *k*, and

^{*}Research done while the author was a postdoc at Carnegie Mellon.

[†]Supported by the NSF ALADDIN Center (NSF Grant No. CCR-0122581) and an NSF Graduate Research Fellowship.

This paper is a slightly updated version of the original that appeared in PODS’04.

withhold a minimum amount of information to achieve this privacy level. We will show that this optimization problem is *NP*-hard in general. Moreover, we prove that a further restriction of the problem where attributes are suppressed instead of individual entries is also *NP*-hard. On the positive side, we will present a greedy $O(k \log k)$ -approximation algorithm for optimal k -anonymity via suppression of entries (note the hidden constant in the big- O is no more than 4; see Section 4 for more details). This approximation ratio is fairly nice, since the value for k used in practice is no more than 5 or 6 [9]. We remark that for the special case $m \in O(\log n)$ (where m is the degree of the relation and n is the number of tuples), an polynomial time exact algorithm has been recently proposed by Sweeney [8]. Hence our algorithm will probably be best applied in cases with high-dimensional records.

2. NOTATION

We will consider degree- m tuples in the database to be m -dimensional vectors v_i drawn from Σ^m , where Σ is a (finite) alphabet of possible values for attributes. (In general, Σ could vary for each attribute.) Thus the databases under consideration are formally represented as subsets $V \subseteq \Sigma^m$. Let $*$ be a fresh symbol not in Σ .

DEFINITION 2.1. *Let t be a map from V to $(\Sigma \cup \{*\})^m$. We say t is a suppressor on V if for all $v \in V$ and $j = 1, \dots, m$ it is the case that $t(v)[j] \in \{v[j], *\}$.*

Intuitively speaking, a suppressor defines some kind of anonymization: every vector $v \in V$ has a corresponding anonymized vector $t(v) = v'$ in an anonymized set $V' \subseteq (\Sigma \cup \{*\})^m$. The coordinates of v' are identical to the coordinates of v , except some coordinates may be *suppressed* by the new “anonymous” character $*$.

We can extend a suppressor t to act on a set of vectors V in a straightforward way (so that $t(V)$ makes sense). We will regard $t(V)$ as a multiset when two or more vectors in v map to the same suppressed vector, i.e. $v \neq v' \in V$ but $t(v) = t(v')$. We can now define the notion of k -anonymization precisely.

DEFINITION 2.2. *Let t be a suppressor on the set $V = \{v_1, \dots, v_n\} \subseteq \Sigma^m$. Then $t(V)$ is k -anonymous iff for all $v_i \in V$, there exist $k-1$ indices $i_1, i_2, \dots, i_{k-1} \in \{1, \dots, n\}$ (all pairwise distinct and distinct from i) such that $t(v_{i_1}) = t(v_{i_2}) = \dots = t(v_{i_{k-1}}) = t(v_i)$. Alternatively, we call t a k -anonymizer on V .*

In other words, when a suppressor on a database makes the database k -anonymous, it means that every anonymized vector is a member of a multiset of (at least) k identical anonymized vectors. Throughout, we will call such a set of vectors a k -group. Note that k -groups need not have cardinality exactly k , though they must have at least k .

3. HARDNESS OF K-ANONYMITY

The k -anonymity decision problem may be formally defined as follows.

k -ANONYMITY: *Given $V \subseteq \Sigma^m$, $l \in \mathbb{N}$, is there a suppressor t such that $t(V)$ is k -anonymous, and the total number of vector coordinates suppressed in $t(V)$ is at most l ?*

Our first result is that if there is no restriction on the alphabet size, then optimal k -anonymization is hard for all $k \geq 3$.

THEOREM 3.1. *k -ANONYMITY is *NP*-hard for $k \geq 3$, if $|\Sigma| \geq |V|$ is allowed.*

PROOF. The reduction is from k -DIMENSIONAL PERFECT MATCHING: *Given a k -hypergraph $H = (U, E)$ with $n = |U|$ and $m = |E|$, is there a subset $S \subseteq E$ of n/k hyperedges such that each vertex of U is contained in exactly one hyperedge of S ?*

Assume (without loss of generality) that H is simple, i.e. it has no repeated edges in its description. Let $U = \{u_1, \dots, u_n\}$ and $E = \{e_1, \dots, e_m\}$ denote the nodes and edges of a k -dimensional hypergraph H , and let the alphabet $\Sigma = \{0, 1, \dots, n\}$. We will construct a database V as follows. For each u_i , define an m -dimensional vector $v_i \in \Sigma^m$:

$$v_i[j] := \begin{cases} 0 & \text{if } u_i \in e_j, \\ i & \text{otherwise.} \end{cases}$$

Set $V := \{v_1, \dots, v_n\}$. Assume t suppresses the minimum number of vector coordinates and maintains k -anonymity. We claim that the total number of coordinates suppressed by t is at most $n(m-1)$ if and only if there is a k -dimensional perfect matching in H .

We prove the claim for $k = 3$; a straightforward generalization of our argument proves it for all larger k . First, suppose there is a perfect 3-dimensional matching M in H . For $i = 1, \dots, n$, let $j(i)$ be such that $e_{j(i)}$ is the unique hyperedge from M that contains node u_i . Define a suppressor t by

$$t(v_i)[j'] := \begin{cases} 0 & \text{if } j' = j(i), \\ * & \text{otherwise.} \end{cases}$$

Since u_i is on hyperedge $e_{j(i)}$, it follows by definition that $v_i[j(i)] = 0$, and all other coordinates are $*$. Therefore t is a suppressor on V .

Now consider any $t(v_i)$. There are three nodes $u_i, u_{i'}, u_{i''}$ on the hyperedge $e_{j(i)}$, and each node has identical anonymized vectors, i.e. $t(v_i) = t(v_{i'}) = t(v_{i''})$. Hence there are three vectors in $t(V)$ which are identical to $t(v_i)$ (including $t(v_i)$ itself). This shows that $t(V)$ is 3-anonymous. Since every $t(v) \in t(V)$ has exactly one non- $*$ coordinate, the value of our solution is exactly $n(m-1)$. Therefore the optimum 3-anonymized solution will have at most this many $*$'s in its vectors.

For the converse, consider a 3-anonymous suppressor t for V , and assume it suppressed at most $n(m-1)$ coordinates. We claim that every vector $t(v)$ in $t(V)$ has at most one non- $*$ coordinate. Suppose not, and consider a counterexample $t(v_i)$. Since $t(V)$ is 3-anonymous, there must exist two other identical vectors, say $t(v_{i'})$ and $t(v_{i''})$. Since the non- $*$ coordinates have the same values as in the original v_i vectors, it follows that $v_i, v_{i'}$, and $v_{i''}$ are identical in two distinct coordinates; call them j and j' . By construction, any two v_i vectors can match only in coordinates that are 0, and $v_i[j] = 0$ only if node u_i is on edge e_j . Hence $v_i, v_{i'}$, and $v_{i''}$ are all on two distinct edges, e_j and $e_{j'}$. But this implies that two edges of H are identical, contradicting our assumption that H is simple. Thus every vector v has at most one non- $*$ coordinate in its 3-anonymous form $t(v)$. Hence at least $n(m-1)$ coordinates in $t(V)$ are suppressed.

Therefore, if we obtain a $t(V)$ with at most $n(m-1)$ suppressed coordinates, it must be that every vector in $t(V)$ has exactly one non- $*$ coordinate. Given this fact, we can extract a perfect matching M for the original hypergraph H .

For each $i = 1, \dots, n$, consider the non- $*$ coordinate in $t(v_i)$. This coordinate must have value 0 (otherwise there can be no identical vectors). If this is coordinate j , we add hyperedge e_j to a matching M . Clearly we produce a set of hyperedges such that each node is on at least one hyperedge. Since there are 3 identical vectors for every vector v , it follows that there are at most $\frac{n}{3}$ edges in M . Since we need $\frac{n}{3}$ edges at minimum to cover every node, there must be exactly $\frac{n}{3}$ edges, the set of which is a perfect matching. \square

3.1 Anonymizing attributes is hard

Another version of k -anonymity, where we choose whether or not to suppress various attributes from the database, is also hard. Say that *attribute j is suppressed by t* if for all $v \in V$, $v[j] = *$. Define k -ATTRIBUTE-ANONYMITY to be the problem of k -anonymizing a V in a way that minimizes the number of attributes suppressed.

THEOREM 3.2. *For $k > 2$, k -ANONYMITY ON ATTRIBUTES is NP-hard, for any Σ such that $|\Sigma| \geq 2$.*

PROOF. We will just give a proof sketch, as it is quite similar to the proof in the previous section, except we do not need a large alphabet. Let H be a k -hypergraph with vertices u_1, \dots, u_n and edges e_1, \dots, e_m . We will build a database of n vectors $v_1, \dots, v_n \in \Sigma^m$ where each v_i represents a vertex in H , and there exists a suppressor that k -anonymizes the database (suppressing $m - n/k$ attributes) if and only if H has a perfect matching.

Since $|\Sigma| \geq 2$, there are two symbols $b_0 \neq b_1$ in Σ . We set $v_i[j] = b_1$ if $u_i \in e_j$, otherwise $v_i[j] = b_0$. Thus the suppression of an attribute is equivalent to the removal of a hyperedge in H . Observing that for every j there are exactly k vectors v_i such that $v_i[j] = b_1$, it follows that if attribute j is not suppressed in a k -anonymization, then one k -group consists exactly of these k vectors with $v_i[j] = b_1$. Two attributes i and j are not suppressed in a k -anonymization if and only if $e_i \cap e_j = \emptyset$ (similar to the previous theorem). This implies at least $m - n/k$ attributes must be suppressed in any k -anonymization, otherwise two edges share a vertex. If exactly $m - n/k$ attributes are suppressed in a k -anonymization, then n/k attributes remain, each representing a hyperedge disjoint from the others that remain, i.e. H has a perfect matching. If H has a perfect matching, then by suppressing those $m - n/k$ attributes not in this matching, each remaining attribute j has k vectors (with $*$'s) that share exactly the same components (they have a b_1 exactly in component j , and b_0 or $*$ elsewhere). This is a k -anonymization. \square

4. APPROXIMATING K-ANONYMITY

We have seen that producing an optimal anonymization of a database is difficult to achieve in general. We will now describe an approximation algorithm for k -anonymizing, which runs in polynomial time and never suppresses more than $O(k \log k)$ times the minimum number of entries that must be suppressed in order to achieve k -anonymity. Considering that it generally suffices in practice [9] for k to be a small constant (around 5 or 6), this is a quite positive result. The algorithm is a greedy strategy whose proof is somewhat involved, requiring several steps. We begin with a definition. Over the following sections, let Σ , V , and t be given.

DEFINITION 4.1. *Let $S \subseteq \Sigma^m$, and $u, v \in \Sigma^m$. The distance between u and v is $d(u, v) := |\{j : u[j] \neq v[j]\}|$. The*

diameter of S is

$$d(S) := \max_{u, v \in S} d(u, v).$$

Intuitively, the diameter of S is the maximum number of coordinates in which two vectors of S differ. It is useful to note that this function is a metric.

Example. Let $V = \{1010, 1110, 0110\}$ and $t(b_1 b_2 b_3 b_4) = **b_3 b_4$ (each $b_i \in \{0, 1\}$). Then there is one resulting 3-group $g = \{**10, **10, **10\}$, and the diameter of g is 2 (1010 and 0110 differ in two coordinates).

4.1 Minimum diameters and k -anonymity

In what follows, we will demonstrate a close relationship (with respect to approximation) between an optimization problem on set diameters and k -anonymity.

Let $k_1, k_2 \in \mathbb{N}$ with $k_1 \leq k_2$. Define a (k_1, k_2) -cover of V to be a collection $\{S_1, \dots, S_l\}$ of subsets of V such that $k_1 \leq |S_i| \leq k_2$ for $i = 1, \dots, l$, and for every $v \in V$ there is an S_i such that $v \in S_i$. Define a (k_1, k_2) -partition of V to be a (k_1, k_2) -cover where all the S_i are disjoint.

Observe that any k -anonymizer t on V naturally defines a $(k, |V| - k)$ partition $\Pi(t, V)$, given by the different sets of vectors made identical under t . Moreover, we may assume without loss of generality that a k -anonymizer defines a $(k, 2k - 1)$ partition: if some set S_i has cardinality $2k$ or greater, then by splitting S_i arbitrarily into two disjoint sets S'_i and S''_i (both of cardinality at least k), this new partition requires no more $*$'s to k -anonymize it than the former one.

Let $OPT(V)$ be the value of an optimal solution (the minimum number of $*$'s that must be inserted in the vectors of V to achieve a k -anonymization, over all k -anonymizers t). For $S \subseteq V$, let $ANON(S)$ be the total number of coordinates of vectors in S that must be replaced with a $*$ in order for all vectors in S to be identical. Observe $OPT(V) = \min_{\Pi} (\sum_{S \in \Pi} ANON(S))$, where the minimum is taken over all partitions Π of V into sets whose cardinalities are at least k . Finally, let Π^* be an optimal partition for k -anonymity, i.e. $\sum_{S \in \Pi^*} ANON(S) = OPT(V)$.

LEMMA 4.1. *For all $V \subseteq \Sigma^m$,*

$$k \cdot \min_{\Pi} d(\Pi) \leq k \cdot d(\Pi^*) \leq OPT(V) \leq 3k^2 \cdot d(\Pi^*),$$

where the minimum is taken over all partitions Π of V into sets whose cardinalities are in the range $[k, 2k - 1]$.

PROOF. For any $S \subseteq V$, we have that $|S| \cdot d(S) \leq ANON(S)$, since by definition of diameter, at least $d(S)$ coordinates in each vector must be suppressed in order for all vectors of S to become identical. Summing over all $S \in \Pi^*$, we have $\sum_{S \in \Pi^*} |S| d(S) \leq \sum_{S \in \Pi^*} ANON(S)$, implying

$$k \cdot \min_{\Pi} d(\Pi) \leq k \cdot d(\Pi^*) \leq OPT(V).$$

On the other hand, for some $S \in \Pi^*$, every pair $\{u, v\} \subseteq S$ has distance at most $d(S)$ from each other. That is, we only need to suppress $d(S)$ coordinates in u and v to make them identical. Suppressing at most $d(S)$ coordinates (in every vector of S) over all pairs of vectors in S , it follows that $ANON(S) \leq \binom{|S|}{2} d(S)$. Hence by summing over Π^* , and using the fact that $|S| \leq 2k - 1$ for all $S \in \Pi^*$,

$$OPT(V) \leq \binom{2k-1}{2} d(\Pi^*) < 3k^2 \cdot d(\Pi^*).$$

□

A natural optimization problem arises from the statement of the lemma: *finding a $(k, 2k - 1)$ -partition Π of V such that $d(\Pi)$ is minimized.* Let us name this the k -minimum diameter sum problem. The close relationship between k -anonymity and this problem can be articulated with a simple corollary.

COROLLARY 4.1. *Let $\alpha \geq 1$, and Π be a $(k, 2k - 1)$ -partition with diameter sum at most α times the optimal k -minimum diameter sum. Then the algorithm that anonymizes each $S \in \Pi$ by (over all pairs $\{u, v\} \subseteq S$ and all j such that $u[j] \neq v[j]$) assigning $w[j] := *$ to every $w \in S$ is a $3\alpha k$ -approximation algorithm to optimal k -anonymity.*

PROOF. Suppose $\hat{\Pi}$ achieves the optimal k -minimum diameter sum. From the proof of the previous lemma, the total number of stars inserted into the vectors of V by the described algorithm on Π is

$$\begin{aligned} \alpha \sum_{S \in \hat{\Pi}} \binom{|S|}{2} d(S) &\leq \alpha \binom{2k-1}{2} d(\hat{\Pi}) \leq \alpha \binom{2k-1}{2} d(\Pi^*) \\ &\leq \alpha \frac{\binom{2k-1}{2}}{k} OPT(V) < 3\alpha k OPT(V). \end{aligned}$$

□

Therefore, a $3k(1 + \ln 2k)$ approximation to optimal k -anonymity will follow if there is a $(1 + \ln 2k)$ -approximation to the k -minimum diameter sum problem.¹ We will now discuss the construction of such an approximation.

4.2 Approximating k -minimum diameter sum

At a high level, the approximation algorithm for k -minimum diameter sum runs in two phases:

Phase 1: Produce a $(k, 2k - 1)$ -cover whose diameter sum is at most $(1 + \ln 2k)$ times the optimal k -minimum diameter sum (for partitions).

Phase 2: Convert the cover into a $(k, 2k - 1)$ -partition, with no increase in the diameter sum.

4.2.1 Producing a cover

Let \mathcal{C} denote the collection of all subsets of V with cardinality in the range $[k, 2k - 1]$. We will execute the well-known greedy algorithm for approximating the usual set cover problem [6] on the collection \mathcal{C} .

Cover(V, \mathcal{C}):

Build \mathcal{C} as defined above.

Initially, $\Pi := \emptyset$, $D = \emptyset$.

While ($D \neq V$),

 For each $S \in \mathcal{C}$, measure the ratio

$$r(S) = \frac{d(S)}{|S \cap (V - D)|}.$$

 Choose an S such that $r(S)$ is minimum.

$D := D \cup S$.

$\Pi := \Pi \cup \{S\}$.

End while.

Return Π .

¹Note $\ln x$ denotes the natural logarithm of x .

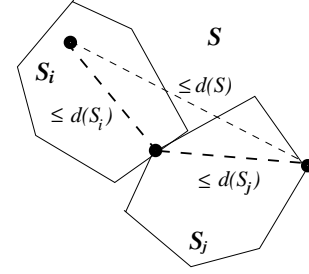


Figure 1: S is the union of S_i and S_j with non-empty intersection. The diameter of S is bounded from above by $d(S_i) + d(S_j)$, by the triangle inequality on diameters.

In each iteration, D contains the vectors in V that have been covered so far by Π . Also, any chosen S always contains an element of $V - D$, so the above takes at most $O(|V|)$ iterations. Therefore the algorithm returns a $(k, 2k - 1)$ -cover, *i.e.* the S are all of cardinality between k and $2k - 1$, and for all $v \in V$, there is an $S \in \Pi$ that contains v .

Invoking the analysis of the greedy algorithm for set cover on subsets of cardinality at most $2k$, the collection Π is a $(1 + \ln 2k)$ -approximation to the k -minimum diameter sum problem, when Π is not restricted to be a $(k, 2k - 1)$ -partition but merely a $(k, 2k - 1)$ -cover of V . Clearly, the minimum over this larger space is at most the minimum over $(k, 2k - 1)$ -partitions.

4.2.2 Converting the cover into a partition

Of course, Π is not necessarily a $(k, 2k - 1)$ -partition of V ; some sets of Π may have non-empty intersection with each other. If so, apply the following reduction:

Reduce(Π):

1. Look for $S_i, S_j \in \Pi$ and $v \in V$ such that $v \in S_i \cap S_j$.

2. If found, do one of the following:

- If either $|S_i|$ or $|S_j|$ is greater than k , remove v from the larger set. (Removing an element from a set can only decrease its diameter.)
- Otherwise, $|S_i| = |S_j| = k$. Replace S_i and S_j in Π with $S_i \cup S_j$. (Note $|S_i \cup S_j| \leq 2k - 1$ since v is in both, and $d(S_i \cup S_j) \leq d(S_i) + d(S_j)$ so again $d(\Pi)$ can only decrease; cf. Figure 1.)

3. Return the new Π .

It is easy to see that repeating **Reduce** until it no longer applies to Π will eventually produce a $(k, 2k - 1)$ -partition with diameter sum at most the original diameter sum (and could be smaller). Furthermore, at most $|V|^2$ repetitions are required, since each application of the reduction removes some v from some set in Π (by either removing v , or removing some S_i that contains it). This completes the approximation algorithm.

4.2.3 Runtime analysis

The number of steps over the two phases is roughly $O\left(\binom{|V|}{2k-1} |V|\right) = O(|V|^{2k})$. In the first phase, there are $O(\binom{|V|}{2k-1})$ sets in the collection \mathcal{C} , and we must choose at most $|V|$ sets from \mathcal{C} , with each set choice requiring $O(\binom{|V|}{2k-1})$ time to determine. In the second phase, the runtime is simply $O(|V|^3)$, so the overall time is dominated by Phase 1.

4.2.4 Summary

The below summarizes the algorithm.

Approximation for optimal k -anonymity:

Let $V \subseteq \Sigma^m$.

1. Set $\Pi := \mathbf{Cover}(V, \mathcal{C})$. (Section 4.2.1).
 2. Repeat $\Pi := \mathbf{Reduce}(\Pi)$ until $\Pi = \mathbf{Reduce}(\Pi)$ (Section 4.2.2).
 3. For each $S \in \Pi$, insert the minimum number of $*$'s in vectors of S such that all vectors of S are identical.
- Return Π .

Steps 1 and 2 produce an $(1 + \ln 2k)$ -approximation to k -minimum diameter sum. Lemma 4.1 and Corollary 4.1 show that Step 3 results in a $3k(1 + \ln 2k)$ -approximation to optimal k -anonymity. Let us record this fact before proceeding further.

THEOREM 4.1. k -ANONYMITY has a $3k(1 + \ln 2k)$ -approximation that runs in $O(|V|^{2k})$ time.

4.3 A strongly polynomial approximation

Due to Phase 1, the above algorithm suffers from a runtime that is exponential in k . We will outline a modification to Phase 1 which makes the algorithm “strongly polynomial” (i.e. polynomial in n and k), but the approximation ratio becomes $O(k \log m)$. We will restrict the greedy algorithm to find a set cover over a much smaller family of subsets. Define, for all $c \in V$ and $i \in \{1, \dots, m\}$, the set

$$S_{c,i} := \{v \in V : d(c, v) \leq i\}.$$

That is, $S_{c,i}$ is defined by a fixed center c and radius i , so there are $m|V|$ possible sets. (Alternatively, we could define for all $c, c' \in V$, $S_{c,c'} := \{v \in V : d(c, v) \leq d(c, c')\}$. This would yield a total of $|V|^2$ sets instead; we naturally advise to substitute whichever collection is smaller in what follows.) The following is straightforward, and follows from the triangle inequality on d .

LEMMA 4.2. $d(S_{c,i}) \leq 2i$.

Let \mathcal{D} denote the collection of these $S_{c,i}$ with cardinality at least k .

Say that $S \subseteq V$ has a *center* $c \in S$ if for all $v \in S$ we have $d(c, v) \leq \lceil d(S)/2 \rceil$. Consider the optimal value D^* for the k -minimum diameter sum problem on a set of vectors V . If we further insist that each group in a feasible solution has a center, we claim that the value of an optimal solution satisfying this restriction is at most $2D^*$. To this end, we prove a lemma for any $V \subseteq \Sigma^m$.

LEMMA 4.3.

$$\min_{\Pi'} d(\Pi') \leq 2 \min_{\Pi} d(\Pi),$$

where Π ranges over all $(k, 2k-1)$ -covers of V , and Π' ranges over covers of V by sets from \mathcal{D} .

PROOF. Let $\hat{\Pi}$ be a $(k, 2k-1)$ -cover such that $d(\hat{\Pi}) = \min_{\Pi} d(\Pi)$. For each set $T \in \hat{\Pi}$, let $c_T \in T$ be arbitrarily chosen. Define $\Omega := \{S_{c_T, d(T)} : T \in \hat{\Pi}\}$.

Observe that $T \subseteq S_{c_T, d(T)}$; it follows that $|S_{c_T, d(T)}| \geq k$, $\Omega \subseteq \mathcal{D}$, and that Ω is a cover of V .

Finally, by the previous lemma we have

$$d(\Omega) = \sum_{T \in \hat{\Pi}} d(S_{c_T, d(T)}) \leq \sum_{T \in \hat{\Pi}} 2d(T) = 2d(\hat{\Pi}). \quad \square$$

Now, the new Phase 1 will execute **Cover** on the (significantly smaller) collection \mathcal{D} of $S_{c,i}$ over all $c \in V$ and $i \in \{1, \dots, m\}$. This modification runs in $O(m \cdot |V|^2)$ time (or, using the alternative formulation, $O(|V|^3)$ time), but since the cardinality of the sets in \mathcal{D} can be up to m , the approximation ratio for diameter sum over this collection via the greedy algorithm is $(1 + \ln m)$. This implies a $2(1 + \ln m)$ -approximation to k -minimum diameter sum when every set in the partition has a center, and we arrive at a much more favorable result.

THEOREM 4.2. k -ANONYMITY has a $6k(1 + \ln m)$ -approximation that runs in $O(m|V|^2 + |V|^3)$ time.

We are confident that this time bound can be significantly improved using appropriate data structures and a more sophisticated analysis, but this is beyond the scope of our work. Essentially the most computationally intensive part is computing the measures $r(S)$ for each set S and taking the minimum, which might be well-approximated by some other method.

5. CONCLUSIONS

We have formally defined the problem of k -anonymizing a database via suppressing tuple components, and determined the computational complexity of k -anonymization when one wishes to withhold a minimum number of entries yet achieve a privacy level k . In general, the problem is NP -hard. However, our proof for the general case uses an alphabet Σ of large size, so it is possible that the problem is still tractable for small (constant-sized) alphabets. On the other hand, we also analyzed the complexity of the problem variant where instead of choosing individual entries to suppress, we choose entire attributes to suppress, and found it to be NP -hard even for Boolean attributes—this may suggest there is not much hope in finding tractable subcases of the general problem.

Finally, we presented two efficient greedy approximation algorithms for the problem. Can an approximation algorithm be found whose performance ratio is independent of k ? We suspect that $\Omega(\log k)$ might be a lower bound on the possible approximability of the problem (within polynomial time), given that such a lower bound exists for k -set cover.

6. ACKNOWLEDGEMENTS

Thanks to Manuel Blum, Latanya Sweeney, and Maverick Woo for discussions on the problem. We would also like to thank the anonymous reviewers for their very helpful feedback.

7. REFERENCES

- [1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proc. of the 28th International Conference on Very Large Databases*, 143–154, 2002.
- [2] R. Agrawal and S. Ramakrishnan. Privacy Preserving Data Mining. In *Proc. of ACM International Conference on Management of Data*, 439–450, 2000.
- [3] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of ACM Symposium on Principles of Database Systems*, 2001.

- [4] I. Dinur and K. Nissim. Revealing Information while Preserving Privacy. In *Proc. of ACM Symposium on Principles of Database Systems*, 202–210, 2003.
- [5] A. Evfimievski, J. E. Gehrke, and R. Srikant. Limiting Privacy Breaches in Privacy Preserving Data Mining. In *Proc. of ACM Symposium on Principles of Database Systems*, 211–222, 2003.
- [6] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* 9:256–278, 1974.
- [7] J. Kleinberg, C. Papadimitriou, P. Raghavan. Auditing Boolean Attributes. In *Proc. of ACM Symposium on Principles of Database Systems*, 86–91, 2000.
- [8] L. Sweeney. Optimal anonymity using k-similar, a new clustering algorithm. Under review, 2003.
- [9] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10(5), 557–570, 2002.
- [10] P. Samarati and L. Sweeney. Generalizing Data to Provide Anonymity when Disclosing Information (Abstract). In *Proc. of ACM Symposium on Principles of Database Systems*, 188, 1998.