General k-Anonymization is Hard

Adam Meyerson [1] Ryan Williams[2]

March 2003

CMU-CS-03-113

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The strategy of *k-anonymization*, proposed by Sweeney [7] is a technique for protecting the privacy of individuals while allowing large-scale data mining. An optimum anonymization is one which obtains the required privacy protection (insuring that every individual is a member of a group of k identical individuals) while minimizing the amount of data hidden from the data miner. We prove that finding such an optimum anonymization is NP-Hard.

[1]Aladdin Project, Carnegie-Mellon University. Research supported by NSF grant CCR-0122581. Email: `adam@cs.cmu.edu`

[2]Computer Science Department, Carnegie Mellon University. Email: `ryanw@cs.cmu.edu`

Keywords: privacy, anonymization, NP-Hardness, complexity, data mining

1 Introduction

Data privacy and data mining, two long-studied and celebrated areas, are quite often at odds with each other. A data miner wishes to have access to large amounts of personal data, in the hopes that he/she may spot unusual trends or correlations. For example, in bioterrorism surveillance, one would like to mine recent medical records of local hospitals, attempting to find patterns in unusual symptoms, locations of patients with these symptoms, and temporal trends (e.g. increases in such patients). [7]. However the direct release and study of such data clearly violates the privacy of individuals. Ideally, we'd like the best of both worlds: to infer overall trends without inferring vital information about particulars. Recently, myriad approaches have been suggested and implemented (cf. [1, 2]), with few complexity-theoretic results specifically geared towards this cause. One example is Kleinberg, Papadimitriou, and Raghavan [3], who show that the problem of *auditing* Boolean attributes is NP-hard in general.

Here we focus on the strategy of k-*anonymization*, first proposed by Sweeney [7]. Let $k > 1$ be a constant 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 is now indistinguishable from $k - 1$ other records that appear, we say that the new modified table is k-anonymized. For 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 our database has been augmented to permit the proper values for attributes, then one possible 2-anonymization of the table would be:

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

Performing k-anonymization provides a way to protect the privacy of individuals while maintaining data integrity. We consider the complexity of rendering relations of private records k-*anonymous*, while still maximizing the amount of information released to the public.

We will prove that two variations on the decision version of this problem are NP-hard when degree of the relation, m, is independent of the number of tuples, n. When $m \leq \log(n)$ (the situation usually observed in practice), an exact polynomial time algorithm has been proposed by Sweeney [4].

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 an alphabet of possible values for the attributes. In general, Σ could vary for each attribute; for simplicity, we consider it to be universal. So the databases under consideration are formally represented as a set $V \subseteq \Sigma^m$. Throughout the paper, we use the notation $n = |V|$, and define $v[j]$ to be the jth coordinate of $v \in V$.

We now define the kind of anonymity considered in this paper. Let $t : V \rightarrow [\Sigma \bigcup \{*\}]^m$. If for all $v \in V$ and $j = 1, \dots, m$ it is the case that $t(v)[j] \in \{v[j], *\}$, we call t an *suppressor*. Intuitively speaking, a suppressor defines an anonymization: every vector $v \in V$ has a corresponding *anonymized* vector $t(v) = v'$ in an anonymized set $V' \subseteq [\Sigma \bigcup \{*\}]^m$. The coordinates of this corresponding vector are always identical to the coordinates of v_i, except that some coordinates may be replaced by the new "anonymous" character *.

The resulting $t(V)$ from a suppressor t is defined to be *k-anonymous* if and only if for all $v_i \in V$ there exist at least k distinct coordinates $i_1, i_2, ...i_k$ such that $t(v_{i_1}) = t(v_{i_2}) = ... = t(v_{i_k}) = t(v_i)$ for each of these coordinates. In other words, any anonymized vector is a member of a set of k identical anonymized vectors. We will call such a set of vectors a *k-group*.

3 Anonymizing entire attributes is hard

Consider a relation with personal attributes such as first name, d.o.b., and diagnosis. We'd like to make public as many of these attributes as possible for the purpose of queries, but still provide k-anonymity in the database by preventing key identifying attributes from being accessed. We will show that achieving this is NP-hard to do. The following is a formalization of the anonymizing attribute problem. We will say that *attribute i is suppressed by t* if for all $v \in V$, $v[j] = *$.

k-ANONYMITY ON ATTRIBUTES: Given V and an non-negative integer L, does there exist a suppressor t such that $t(V)$ is k-anonymous, and at most L attributes are suppressed by t?

Theorem 1 *For $k > 2$, k-ANONYMITY ON ATTRIBUTES is NP-hard, for any Σ such that $|\Sigma| \geq 2$.*

The proof is a reduction from k-dimensional perfect matching, a famous NP-hard problem. We include its definition here for completeness. Recall that a *k-hypergraph* is a hypergraph where each edge contains exactly k vertices.

k-DIMENSIONAL MATCHING:*Given a k-hypergraph with n vertices and m hyperedges, does there exist a subset of $\frac{n}{k}$ hyperedges such that each vertex is contained in exactly one hyperedge?*

We will just give a proof sketch, as it quite similar to the proof in the following section, except we do not need a large alphabet. Let H be an k-hypergraph with vertices $v_1, \dots, v_n$ 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, such that 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 $v_i \in e_j$, 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_l such that $V_l[j] = b_1$, it follows that if attribute j is not suppressed in a k-anonymization, then one k-group consists exactly of these k vectors with $V_l[j] = b_1$. Two attributes i and j are not suppressed in a k-anonymization if and only if $e_i \cap e_j = \varnothing$ (this is shown carefully in the next section). This implies at least $m - n/k$ attributes must be suppressed in any k-anonymization, otherwise two edges share an 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 reamin– 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.

4 Anonymizing entries is hard

We shift to a slightly less restrictive version of the above problem, where it is possible to suppress individual entries, without having to suppress entire attributes. This problem has been explored extensively by Sweeney et al. [5, 6, 7].

In this version of k-anonymity, the quality of our solution will be measured by the total number of coordinates in all vectors of V' which contain the character *. This measure will be between 0 (best) and nm (worst). Our goal is to minimize the number of * coordinates. Unfortunately, this is difficult:

Theorem 2 *For $k \geq 3$, k-ANONYMITY ON ENTRIES is NP-hard, when $|\Sigma| \geq n$.*

The proof is again by reduction from k-dimensional perfect matching. Suppose we could solve the 3-anonymity problem in polynomial time in the size of the inputs. We will describe a polynomial-time algorithm for 3-dimensional perfect matching. The argument for arbitrary $k \geq 3$ holds via a straightforward generalization.

We define the alphabet $\Sigma_0 = \{0, 1, ..., n\}$ (which may be thought of as choosing n distinct symbols from the original Σ and relabeling them). For each node i in the hypergraph, we will define an m-dimensional vector v_i (here m is once again the number of edges in the hypergraph). Coordinate j of vector i has value $v_i[j] = i$ if node i is not on edge j, and value $v_i[j] = 0$ if node i is on edge j. We have now constructed a set V drawn from the alphabet Σ_0^m, with $|V| = n$.

Suppose we optimally solve 3-anonymity on these vectors V. We have produced a set of vectors V'. We claim that the number of *'s used to produce V' is at most $n(m-1)$ if there exists a perfect 3-dimensional matching in the original graph, and otherwise larger.

First, suppose there exists a perfect 3-dimensional matching. Let $M(i)$ represent the unique edge from this matching which contains node i. We define $v'_i[M(i)] = 0$ and $v'_i[j] = *$ for each $j \neq M(i)$. Since i is on edge $M(i)$, $v_i[M(i)] = 0$ must have held. Now consider any anonymized vector v'_i. There are three nodes on edge $M(i)$ and each of these nodes has identical anomyized vectors. Thus there exists a set of 3 vectors which are identical to v'_i (including v'_i itself). This shows that our solution is feasible. Since every anonymized vector has exactly one non-* coordinate, the value of our solution is exactly $n(m-1)$. The optimum 3-anonymized solution will have at most this many * terms (actually the above construction is optimum).

Now suppose that no perfect 3-dimensional matching exists. Consider any 3-anonymous solution to the vector problem. Can there exist a vector with two non-* coordinates in its anonymized form? Consider some such vector v'_i. There must exist two other identical vectors. Since the non-* coordinates have the same values as in the original v_i vectors, we must have three v_i vectors which are identical in two coordinates. Any two v_i vectors can match only in coordinates which are 0. The coordinate j is zero only if node i is on edge j. It follows that we have three nodes which are on each of two edges. But this means two edges are identical, which could not occur in the original graph. Thus we have a contradiction and every vector has at most one non-* coordinate in its anonymized form.

If we are to obtain a solution with at most $n(m-1)$ terms that are *, we must have every vector with exactly one non-* coordinate. If this is the case, we will construct a perfect matching. For each i we consider the non-* coordinate in v_i'. This coordinate must have value zero (otherwise there can be no identical vectors). If this is coordinate j, we will add edge j to our matching. Clearly we produce a set of edges such that each node is on at least one edge. How many edges are in our set? Since there are 3 identical vectors for every solution vector, at least three vectors produce any given edge in our set. It follows that there are at most $\frac{n}{3}$ edges. Since we need $\frac{n}{3}$ edges at minimum to cover every node, there must be exactly $\frac{n}{3}$ edges, which comprise a perfect matching. This is a contradiction, so it follows that there exists a 3-dimensional perfect matching if and only if the optimum 3-anonymous solution has at most $n(m-1)$ *'s. This completes the proof of reduction. □

5 Acknowledgements

Thanks to Manuel Blum, Latanya Sweeney, and Maverick Woo for discussions on the problem.

References

[1] R. Agrawal and S. Ramakrishnan. *Privacy-preserving data mining.* Proc. of ACM International Conference on Management of Data, pp. 439–450, 2000. R. Agrawal and R. Srikant. *Privacy Preserving Data Mining.*

[2] D. Agrawal and C. C. Aggarwal. *On the design and quantification of privacy preserving data mining algorithms.* Proc. of ACM Symposium on Principles of Database Systems, 2001. `http://citeseer.nj.nec.com/agrawal01design.html`

[3] J. Kleinberg, C. Papadimitriou, P. Raghavan. *Auditing Boolean Attributes.* Proc. of ACM Symposium on Principles of Database Systems, 86-91, 2000. `http://citeseer.nj.nec.com/445207.html.`

[4] L. Sweeney. *Optimal anonymity using k-similar, a new clustering algorithm.* Under review for publication, 2003.

[5] L. Sweeney. *k-anonymity: a model for protecting privacy.* International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 557-570, 2002.

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

[7] L. Sweeney. *Guaranteeing anonymity when sharing medical data, the datafly system.* Proceedings of Journal of the American Medical Informatics Association. Hanley and Belfus, Inc., 1997.