k-Anonymization with Minimal Loss of Information

Aristides Gionis * Tamir Tassa †

July 4, 2006

Abstract

The technique of k-anonymization allows the releasing of databases that contain personal information while ensuring some degree of individual privacy. Given a database D that needs to be released, one produces a so-called k-anonymized version of that database where each record is indistinguishable from at least $k-1$ additional records. The anonymization process is usually performed by suppressing or generalizing database entries. We formally study the concept of generalization, and propose two information-theoretic measures for capturing the amount of information that is lost during the anonymization process. We call these two measures, *the entropy measure* and *the non-uniform entropy measure*. The proposed measures are more general and more accurate than the measures that were proposed by Meyerson and Williams [MW04] and Aggarwal et al. [AFK+05]. We then study the problem of achieving k-anonymity with minimal loss of information. We prove that this problem is NP-hard and then study polynomial approximations for the optimal solution. Our first algorithm relies on similar ideas as the approximation algorithm that was proposed in [MW04]. It gives an approximation guarantee of $O(\ln k)$, for the entropy measure as well as for the previously studied measures. This improves the best-known $O(k)$-approximation of [AFK+05]. While the approximation algorithms of [AFK+05, MW04] relied on the so-called *graph representation* framework, which was shown in [AFK+05] to be limited to $\Omega(k)$-approximations, our algorithm relies on a novel *hypergraph representation* that enables the improvement in the approximation ratio from $O(k)$ to $O(\ln k)$. As the running time of the algorithm is $O(n^{2k})$, we also show how to adapt the algorithm of [AFK+05] in order to obtain a strongly polynomial approximation algorithm for our entropy measure with approximation guarantee of $O(k)$. We leave as an open problem to design an approximation algorithm, strongly polynomial or not, for the non-uniform entropy measure.

1 Introduction

Consider a database that holds information on individuals in some population $U = \{u_1, \ldots, u_n\}$. Each individual is described by a collection of r public attributes (also known as *quasi-identifiers*), $A_1, \ldots, A_r$, and s private attributes, $Z_1, \ldots, Z_s$. Each of the attributes consists of several possible values:

$$A_j = \{a_{j,\ell} : 1 \leq \ell \leq m_j\}, \quad 1 \leq j \leq r\,,$$

and

$$Z_j = \{z_{j,\ell} : 1 \leq \ell \leq n_j\}, \quad 1 \leq j \leq s\,.$$

*Basic Research Unit, HIIT, Department of Computer Science, University of Helsinki, Finland. gionis@cs.helsinki.fi

†Division of Computer Science, The Open University, Ra'anana, Israel. tamirta@openu.ac.il

For example, if A_j is gender then $A_j = \{M, F\}$, while if it is the age of the individual, it is a bounded nonnegative natural number. The public database holds all publicly available information on the individuals in U; it takes the form,

$$D = \{R_1, \ldots, R_n\}\,, \quad \text{where } R_i \in A_1 \times \cdots \times A_r\,, \quad 1 \leq i \leq n\,. \tag{1}$$

The corresponding private database holds the private information,

$$D' = \{S_1, \ldots, S_n\}\,, \quad \text{where } S_i \in Z_1 \times \cdots \times Z_s\,, \quad 1 \leq i \leq n\,. \tag{2}$$

The complete database is the concatenation of those two databases, $D\|D' = \{R_1\|S_1, \ldots, R_n\|S_n\}$. We refer hereinafter to the tuples R_i and S_i, $1 \leq i \leq n$, as (public or private) records. The j-th component of the record R_i (namely, the (i,j)-th entry in the database D) will be denoted hereinafter by $R_i(j)$.

Such databases may be of interest to the general public even though they hold information on individuals. For example, if the database holds data on patients that are hospitalized in some hospital, its content may be required for medical research. In such cases, the researchers would like to get as much data as possible in order to find interesting patterns by means of statistical analysis and data mining. However, the hospital is committed to respect the privacy of its patients and, consequently, it cannot release the database as is since an adversary may be able to link the public data in some of the records in the database to some individuals and then learn confidential private information about those individuals. Hence, it is desired to reveal information in order to allow data mining, while respecting the privacy of the individuals that are represented in the database. (In other words, we would like to allow learning information about the *public* but not about the *individuals* of which that public consists.)

Many approaches were suggested for playing this delicate game that requires finding the right path between data hiding and data disclosure. Such approaches include query auditing [DN03, KMN05, KPR03], output perturbation [BDMN05, DN03, DN04], secure multi-party computation [AMP04, FNP04, GMW87, LP02, Yao86], and data sanitization [AA01, AS00, AST05, CDM$^+$05, EGS03]. One of the recent approaches, proposed by Samarati and Sweeney [Sam01, SS98, Swe02] is k-anonymization. The main idea in this approach is to suppress or generalize some of the public data in the database so that each of the public records becomes indistinguishable from at least $k-1$ additional records. Consequently, the private data may be linked to sets of individuals of size no less than k, whence the privacy of the individuals is protected to some extent.

For example, assume that there are $r = 3$ public attributes — name, age, and address — and one ($s = 1$) private attribute — disease. In order to achieve k-anonymity for some $k > 1$, one might suppress the name attribute, replace the age with a range of ages, and replace the exact address with just the zip code. Then, instead of releasing the database $D\|D'$, one releases the anonymized database $g(D)\|D'$, where $g(D)$ is the public database that was obtained from D after applying the above described suppressions and generalizations. It is clear that by such actions of replacing public database entries with more general subsets of values that are consistent with the original values of those entries, one may always arrive at a k-anonymized database $g(D)\|D'$, for any given $k \leq n$.

The problem that we study here is the problem of k-anonymization with minimal loss of information: Given a public database D, and acceptable generalization rules for each of its attributes, find its "nearest" k-anonymization; namely, find a k-anonymization of D that conceals a minimum

amount of information. Meyerson and Williams [MW04] introduced this problem and studied it under the assumption that database entries may be either left intact or totally suppressed. In that setting, the goal is to achieve k-anonymity while minimizing the number of suppressed entries. They showed that the problem is NP-hard and devised two approximation algorithms for that problem: One that runs in time $O(n^{2k})$ and achieves an approximation ratio of $O(k \ln k)$; and another that has a strongly polynomial running time and guarantees an approximation ratio of $O(k \ln n)$. Aggarwal et al. [AFK$^+$05] extended the setting of suppressions-only by allowing more general rules for generalizing database entries towards achieving k-anonymity. They proposed a way of penalizing each such action of generalizing a database entry and showed that the problem of achieving k-anonymity in that setting with minimal penalty is NP-hard. They then devised an approximation algorithm for that problem that guarantees an approximation ratio of $O(k)$.

In this study we extend the framework of k-anonymization to include any type of generalization operators and define two measures of loss of information that are both more general and more accurate than the measure that was used in [AFK$^+$05] (the measure that was used in [MW04] is a special case of the one that was used in [AFK$^+$05]). We call these two measures, *the entropy measure* and *the non-uniform entropy measure*. We show that the problem of k-anonymization with minimal loss of data (measured by either of our two measures) is NP-hard and then proceed to describe an approximation algorithm for the entropy measure. The approximation guarantee of our algorithm is $O(\ln k)$ — a significant improvement over the previous best result of $O(k)$. We note that Meyerson and Williams [MW04] hypothesized that k-anonymization cannot be approximated, in polynomial time, with an approximation factor that is $o(\ln k)$. What enabled this significant improvement was our novel approach to this approximation problem. The approximation algorithms in both [MW04] and [AFK$^+$05] were based on the so-called *graph representation*. In [AFK$^+$05] it was shown that using the graph representation it is impossible to achieve an approximation ratio that is better than $\Theta(k)$. We were able to offer the significantly better $O(\ln k)$ approximation ratio by breaking out of the graph representation framework and using a *hypergraph* approach instead.

The paper is organized as follows: In Section 2 we give a precise definition of what is generalization, and we describe and illustrate several natural types of generalization. In Section 3 we describe the measures of loss of information that were used in [AFK$^+$05, MW04] and their shortcomings (Section 3.1); we then propose two measures of loss of information that are more general and more accurate than the previously used ones: the entropy measure (Section 3.2) and the non-uniform entropy measure (Section 3.3). In Section 4 we define the problem of k-anonymization with minimal loss of information and we prove that it is NP-hard with respect to both measures of loss of information. In Section 5 we present an algorithm that approximates optimal k-anonymity with approximation ratio of $O(\ln k)$, for the entropy measure. The running time of that algorithm is $O(n^{2k})$. We then proceed to describe how to adapt the approximation algorithm of [AFK$^+$05] to achieve an $O(k)$-approximation ratio with respect to the entropy measure, in time that is polynomial in both n and k. Finally, Section 6 includes a summary of our study and discussion of some open problems.

2 Generalization

The basic technique for obtaining k-anonymization is by means of *generalization*. By generalization we refer to the act of replacing the values that appear in the database with subsets of values, so that entry $R_i(j)$, $1 \leq i \leq n$, $1 \leq j \leq r$, which is an element of A_j, is replaced by a subset of A_j

that includes that element.

Definition 2.1 *Let A_j, $1 \le j \le r$, be finite sets and let $\overline{A}_j \subseteq \mathcal{P}(A_j)$ be a collection of subsets of A_j. A mapping $g : A_1 \times \cdots \times A_r \to \overline{A}_1 \times \cdots \times \overline{A}_r$ is called a generalization if for every $(b_1, \ldots, b_r) \in A_1 \times \cdots \times A_r$ and $(B_1, \ldots, B_r) = g(b_1, \ldots, b_r)$, it holds that $b_j \in B_j$, $1 \le j \le r$.*

We illustrate the concept of generalization by several examples of natural generalization operators.

The trivial generalization. Assume that for all $1 \le j \le r$ the collection of subsets $\overline{A}_j$ includes all singleton subsets $\{a_{j,\ell}\}$, $1 \le \ell \le m_j$. Then the generalization

$$g(b_1, \ldots, b_r) = (\{b_1\}, \ldots, \{b_r\})$$

is the *trivial generalization* that leaves all entries unchanged. It is always natural to assume that the collection of subsets of each of the attributes includes all singleton subsets because whenever possible we prefer to leave the database entries unchanged. We formulate this assumption as follows:

$$A_j \subseteq \overline{A}_j, \qquad 1 \le j \le r\,. \tag{3}$$

Note the freedom of notation that we take here and that we adopt hereinafter: The set on the right hand side of (3) is a set of *sets*, while the set on the left hand side is a set of *elements*. However, we always identify the element $a_{j,\ell}$ with the set $\{a_{j,\ell}\}$. Hence, the notation A_j on the left hand side of (3) means $A_j = \big\{\{a_{j,1}\}, \ldots, \{a_{j,m_j}\}\big\}$ (as opposed to the original meaning $A_j = \big\{a_{j,1}, \ldots, a_{j,m_j}\big\}$).

Generalization by suppression. Assume that $\overline{A}_j = A_j \cup \{A_j\}$ for all $1 \le j \le r$ and that g either leaves entries unchanged (no generalization) or replaces them by the entire set of attribute values (total generalization),

$$g(b_1, \ldots, b_r) = (\overline{b}_1, \ldots, \overline{b}_r), \quad \text{where} \quad \overline{b}_j \in \{b_j, A_j\}, \quad 1 \le j \le r\,.$$

In that case we refer to g as *generalization by suppression*. Letting * denote an element outside $\bigcup_{1 \le j \le r} A_j$, it is more convenient to think of g as follows,

$$g(b_1, \ldots, b_r) = (\overline{b}_1, \ldots, \overline{b}_r), \quad \text{where} \quad \overline{b}_j \in \{b_j, *\}\,.$$

Generalization by hierarchical clustering trees. In [AFK+05], Aggarwal et al. considered a setting in which for every attribute A_j there is a corresponding balanced tree, $\mathcal{T}(A_j)$, that describes a hierarchical clustering of A_j. Each node of $\mathcal{T}(A_j)$ represents a subset of A_j, the root of the tree is the entire set A_j, the descendants of each node represent a partition of the subset that corresponds to the ancestor node, and the leaves correspond to the singleton subsets. Given such a balanced tree, they considered generalization operators that may replace an entry $R_i(j)$ with any of the ancestors of $R_i(j)$ in $\mathcal{T}(A_j)$. Generalization by suppression is a special case of generalization by clustering trees where all trees are of height 2.

Unrestricted generalization. The case where $\overline{A}_j = \mathcal{P}(A_j)$ is the case of *unrestricted generalization*. Here, each entry $R_i(j)$ may be replaced by any of the subsets of A_j that includes it. Generalizations where $\overline{A}_j \subsetneq \mathcal{P}(A_j)$ will be referred to hereinafter as *restricted generalizations*.

Some of our results require that the collection of subsets $\overline{A}_j$, $1 \le j \le r$, satisfy the following natural property.

Definition 2.2 *Given an attribute $A = \{a_1, \ldots, a_m\}$, a corresponding collection of subsets $\overline{A}$ is called proper if it includes all singleton subsets $\{a_i\}$, $1 \leq i \leq m$, it also includes the entire set A, and for all $B_1, B_2 \in \overline{A}$ that have nonempty intersection, either $B_1 \subseteq B_2$ or $B_2 \subseteq B_1$.*

Lemma 2.3 *Let A be an attribute and $\overline{A}$ be a corresponding collection of subsets. Then $\overline{A}$ is proper if and only if it is consistent with the (possibly unbalanced) hierarchical clustering tree framework.*

Proof. Assume first that $\overline{A}$ is consistent with the (possibly unbalanced) hierarchical clustering tree framework. Then $A \subset \overline{A}$, since the leaves of the tree represent all singleton subsets, and $A \in \overline{A}$ since the root of the tree represents the entire set. In addition, any two intersecting subsets in that tree must appear on the same path from the root to one of the leaves, whence one of them is a subset of the other. Such a collection of subsets is therefore proper.

Assume next that $\overline{A}$ is proper. Construct a directed graph $G = (V, E)$, where $V = \overline{A}$ and for any two distinct sets $B, B' \in \overline{A}$, the graph has the directed edge (B, B') if and only if $B \subset B'$ and there exists no subset $B'' \in \overline{A} \setminus \{B, B'\}$ such that $B \subset B'' \subset B'$. We proceed to show that the obtained directed graph $G = (V, E)$ is a hierarchical clustering tree. Clearly, as $\overline{A}$ includes all singleton subsets, the set of nodes in the graph G having zero in-degree is exactly the set of singleton subsets. As $A \in \overline{A}$, the graph G has exactly one node with a zero out-degree (the root) and that is the node that corresponds to the entire set. It is also clear that every node in G is connected to the root. Hence, it remains only to show that every two nodes $B, B' \in \overline{A}$, can be connected through at most one directed path. Assume, towards contradiction, that there are two directed paths that connect B to B'. All the subsets that appear on either of those two paths include B, so they have non-empty intersection. Hence, as $\overline{A}$ is proper, the relation of set inclusion is a total order on the collection of those subsets. Therefore, those subsets must reside linearly on a single directed path in G. This completes the proof. □

Note that the framework of proper collections of subsets extends the hierarchical clustering tree framework, as it allows unbalanced trees.

Example 2.4

Consider the age attribute, A, and let us assume that $A = \{1, \ldots, 120\}$. In unrestricted generalization we may replace an entry that has the value, say, 27 by any subset of age values that includes 27, say, $\{18, 27, 41, 55\}$. In generalization by suppression we may either leave that entry unchanged or replace it with an undefined entry '$*$' that stands for the set of all possible ages. Assume next that we arrange the age values in a 3-level balanced tree where the root stands for $A = \{1, \ldots, 120\}$, it has 12 descendants that stand for the subsets $\{10(i-1)+1, \ldots, 10i\}$, $1 \leq i \leq 12$, and each of those nodes has 10 descendants that are all singleton leaves. Then in that model we may leave the entry 27 unchanged, or replace it by the range of ages $\{21, \ldots, 30\}$, or totally generalize it by replacing it with the symbol '$*$'. Finally, we may consider other models of restricted generalization in this case: for example, a generalization by intervals allows only subsets of the form $\{i : s \leq i \leq t\}$. Such generalization by intervals, like the unrestricted generalization, is non-proper. □

So far we spoke of generalizations of records. We now turn to speak of generalizations of an entire database.

Definition 2.5 *Let $D = \{R_1, \ldots, R_n\}$ be a database having public attributes $A_1, \ldots, A_r$, let $\overline{A}_1, \ldots, \overline{A}_r$ be corresponding collections of subsets, and let $g_i : A_1 \times \cdots \times A_r \to \overline{A}_1 \times \cdots \times \overline{A}_r$ be corresponding generalization operators. Denoting $\overline{R}_i := g_i(R_i)$, $1 \leq i \leq n$, we refer to the database $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ as a generalization of D.*

Recall that D is a multiset, meaning that it may have repeated records. If all records of D are distinct then we may apply to all records in the database the same mapping $g : A_1 \times \cdots \times A_r \to \overline{A}_1 \times \cdots \times \overline{A}_r$. However, if D includes repeated records, say $R_i = R_j$, for $1 \leq i < j \leq n$, the above definition allows generalizations where $\overline{R}_i \neq \overline{R}_j$.

We conclude this section with the following definitions:

Definition 2.6 *Define a relation $\sqsubseteq$ on $\overline{A}_1 \times \cdots \times \overline{A}_r$ as follows: If $R, R' \in \overline{A}_1 \times \cdots \times \overline{A}_r$ then $R \sqsubseteq R'$ if and only if $R(j) \subseteq R'(j)$ for all $1 \leq j \leq r$.*

It is easy to see that $\sqsubseteq$ defines a partial order on $\overline{A}_1 \times \cdots \times \overline{A}_r$. We may use this partial order to define a partial order on the set of all generalizations of a given database.

Definition 2.7 *Let D be a database and let $g(D)$ and $g'(D)$ be two generalization of D. Then $g(D) \sqsubseteq g'(D)$ if $g(D)_i \sqsubseteq g'(D)_i$ for all $1 \leq i \leq n$.*

3 Measures of loss of information

3.1 Previously used measures

In previous studies of k-anonymity, the quality of a k-anonymization of a given database was measured by the amount of information that was lost due to generalization. Meyerson and Williams [MW04] concentrated on the case of generalization by suppression. Their measure of loss of information was the number of generalized entries (namely, *s) in the k-anonymized database. Aggarwal et al. [AFK+05], who considered generalizations by hierarchical clustering trees, offered the following measure (which we dub *the tree measure*): Assume that the values of an attribute A_j are arranged in a balanced tree $\mathcal{T}(A_j)$, as described above, having $\ell_j + 1$ levels: $L_{j,0}, \cdots, L_{j,\ell_j}$ (the level $L_{j,0}$ consists of the leaves while L_{j,ℓ_j} is the level of the root). Then the cost of replacing the original entry $R_i(j)$ with a subset of A_j that appears in the tree $\mathcal{T}(A_j)$ in level $L_{j,r}$ is r/ℓ_j. The overall cost of the entire k-anonymization is the sum of costs in all entries. Note that the tree measure is a generalization of the measure proposed by Meyerson and Williams (since in the case of generalization by suppression all entries are either left unchanged, thus incurring a zero cost, or replaced by the root of the corresponding tree, thus incurring a maximal cost of 1).

We find the tree measure quite arbitrary. For example, if one attribute is gender and another attribute is age, the loss of information by concealing the gender is much less than that incurred by concealing the age. Also, the levels of the trees $\mathcal{T}(A_j)$ need not be equally-spaced in terms of information loss.

3.2 The entropy measure

Following [DW99] and [WD01], we suggest to use the standard measure of information, namely entropy, in order to assess more accurately the amount of information that is lost by anonymization.

The public database $D = \{R_1, \ldots, R_n\}$ induces a probability distribution for each of the public attributes. Let X_j, $1 \leq j \leq r$, denote hereinafter the value of the attribute A_j in a randomly selected record from D. Then

$$\Pr(X_j = a) = \frac{\#\{1 \leq i \leq n : R_i(j) = a\}}{n}\,.$$

The entropy of X_j is a measure of the amount of information that is delivered by revealing the value of a random sample of X_j (or, equivalently, the amount of uncertainty regarding the value of the random sample before its value is revealed). It is defined as

$$H(X_j) = -\sum_{a \in A_j} \Pr(X_j = a) \log \Pr(X_j = a)\,,$$

where hereinafter $\log = \log_2$. Let B_j be a subset of A_j. Then the conditional entropy $H(X_j|B_j)$ is defined as

$$H(X_j|B_j) = -\sum_{b \in B_j} \Pr(X_j = b|X_j \in B_j) \log \Pr(X_j = b|X_j \in B_j)\,,$$

where

$$\Pr(X_j = b|X_j \in B_j) = \frac{\#\{1 \leq i \leq n : R_i(j) = b\}}{\#\{1 \leq i \leq n : R_i(j) \in B_j\}}\,, \qquad b \in B_j.$$

Note that if $B_j = A_j$ then $H(X_j|B_j) = H(X_j)$ while in the other extreme case where B_j consists of one element, we have zero uncertainty, $H(X_j|B_j) = 0$. This allows us to define the following cost function of a generalization operator:

Definition 3.1 *Let $D = \{R_1, \ldots, R_n\}$ be a database having public attributes $A_1, \ldots, A_r$, and let X_j be the random variable that equals the value of the j-th attribute A_j, $1 \leq j \leq r$, in a randomly selected record from D. Then if $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ is a generalization of D,*

$$\Pi_e(D, g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{r} H(X_j|\overline{R}_i(j)) \tag{4}$$

is the entropy measure of the loss of information caused by generalizing D into $g(D)$.

3.2.1 Discussion

Assume that the entry $R_i(j)$ is left unchanged under the generalization g. Then its contribution to the sum in (4) is $H(X_j|R_i(j)) = 0$, just like in the tree measure. However, if it is suppressed, then its contribution to the sum in (4) is $H(X_j)$, as opposed to the tree measure where it contributes 1, regardless of the properties of that attribute. Therefore, the entropy measure does distinguish between "simple" attributes (such as gender) and attributes that convey more information (like age or address). In addition, in intermediate cases where $R_i(j)$ is generalized to a subset of values $\overline{R}_i(j) \subset A_j$, the contribution to the measure of loss of information in (4) is the exact conditional entropy, and not the somewhat arbitrary fractional value in the definition of the tree measure.

Having said that, we would like to note that while the entropy measure is significantly more accurate than the tree measure, and more general (as it applies to all generalizations and not just to generalizations that comply with the hierarchical clustering tree framework), it is still not

entirely accurate. This measure, just like the tree measure, defines the information loss per entry and then adds up the information that was lost over all entries of the database. In other words, both measures assume that the columns (attributes) of the database are independent, and so are the rows (individuals).

However, the columns of the database need not be independent. For example, if one attribute is location and another is age, it is possible that some locations (say, around central university campuses) will be associated with populations that are younger than elsewhere. In this study we concentrate on the simpler model of independent attributes and leave it for a further research to extend the framework that we lay here to the more general model in which the dependence between attributes is also taken into account.

As for the rows of the database, they are not independent either. There exists dependence between the rows that stems from statistical or social reasons; for example, if one individual in the database is married to another, then they probably have the same location and a similar age. In addition, there exists another type of dependence between the rows that stems from a combinatorial reason, as we proceed to explain. Let D be the original public database, $g(D)$ be its k-anonymization, and D' be the corresponding private database. The publicly available database is $g(D)\|D'$. The records of the non-anonymized database D are also publicly known (through other sources), but they are not ordered. Some orderings of the records in D are consistent with the records of $g(D)$, but some are not. Therefore, an adversary, as well as a data-miner, may analyze all possible orderings of D that agree with $g(D)$ and deduce a-posteriori probabilities for the exact values of the generalized entries that differ from the a-priori probabilities that are implied by D alone.

Example 3.2

Consider the public database D that consists of only one attribute A_1 and four individuals, and its 2-anonymization $g(D)$:

$$D = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \end{bmatrix}, \qquad g(D) = \begin{bmatrix} * \\ * \\ 3 \\ 3 \end{bmatrix}.$$

The a-priori probabilities of the corresponding random variable X_1 in this example are

$$p_1 = \Pr[X_1 = 1] = \frac{1}{4}, \quad p_2 = \Pr[X_1 = 2] = \frac{1}{4}, \quad p_3 = \Pr[X_1 = 3] = \frac{1}{2}.$$

The entropy of such a random variable is $H(X_1) = 1.5$. Hence, according to our measure,

$$\Pi_{\mathrm{e}}(D, g(D)) = 1.5 + 1.5 + 0 + 0 = 3.$$

However, by comparing $g(D)$ to D we can deduce that the suppressed entries should be either 1 or 2, with equal probabilities. Hence, the actual amount of information lost in this case is just $1 + 1 + 0 + 0 = 2$. □

Example 3.2 demonstrates that the entropy measure may overestimate the actual amount of information that is lost by anonymization. It may also underestimate that amount, as exemplified next.

Example 3.3
Consider the following public database D and its 3-anonymization $g(D)$:

$$D = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}, \qquad g(D) = \begin{bmatrix} * \\ * \\ * \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}.$$

In this case the entropy of the random variable X_1 that corresponds to the attribute A_1 is $H(X_1) = \frac{2}{8}\log 8 + \frac{6}{8}\log\frac{8}{6} \approx 1.061$, so that $\Pi_{\mathrm{e}}(D, g(D)) = 3 \cdot H(X_1) \approx 3.183$. However, by comparing $g(D)$ with D one deduces that the suppressed entries are 1, 2, or 3 with probability $\frac{1}{3}$ each, whence the actual amount of information loss is $3 \cdot \log 3 \approx 4.755$. □

Having said that, it should be realized that the above analysis that was simple and straightforward in the given toy examples, can be extremely intricate for large databases with many rows, many columns, more complicated attributes and more general generalization operators. In fact, it is not clear to us whether it is possible to compute in polynomial time the a-posteriori probabilities and the corresponding entropy, due to the exponential number of orderings of D that agree with a given generalization $g(D)$. Hence, while the proposed entropy measure is not accurate from information-theoretic point of view, it seems to be an appropriate measure from computational point of view as we cannot rely on information that requires (possibly) super-polynomial time to reveal.

3.2.2 The non-monotonicity of the entropy measure

A natural property that one might expect from any measure of loss of information is monotonicity:

Definition 3.4 *Let D be a database, let $g(D)$ and $g'(D)$ be two generalizations of D and let Π be any measure of loss of information. Then Π is called monotone if $\Pi(D, g(D)) \leq \Pi(D, g'(D))$ whenever $g(D) \sqsubseteq g'(D)$.*

The tree measure is clearly monotone. The entropy measure Π_{e}, on the other hand, is not always monotone, as we proceed to exemplify.

Example 3.5
Consider a database with one ($r = 1$) attribute that may get the values $\{1, 2, 3, 4\}$ with probabilities $\{1 - 3\varepsilon, \varepsilon, \varepsilon, \varepsilon\}$ respectively, where $\varepsilon \ll 1$. The entropy of that attribute is $h(1 - 3\varepsilon, \varepsilon, \varepsilon, \varepsilon) \approx 0$, where hereinafter $h(p_1, \ldots, p_t) := -\sum_{i=1}^{t} p_i \log p_i$ denotes the entropy of a discrete t-valued random variable with probabilities $\{p_1, \ldots, p_t\}$.

Next, assume that the values of this attribute are arranged in a tree with three levels where the root is the entire set of values, the descendants in the next level are the subsets $\{1, 2\}$ and $\{3, 4\}$, and the third level consists of the four singleton subsets. Entries with the value 4 may be generalized to $\{3, 4\}$ or be suppressed. The first generalization, $4 \mapsto \{3, 4\}$, incurs a cost of 1 bit,

since given that the unknown attribute value is in the subset $\{3,4\}$, it can be either of the two values with equal probabilities. However, if we suppress such an entry, the resulting cost is the entropy $h(1-3\varepsilon,\varepsilon,\varepsilon,\varepsilon) \approx 0$. Namely, the entropy measure is not monotone in this case as it favors the total suppression of such entries over the partial generalizations to $\{3,4\}$. □

The question is which measure serves our goal better here — the monotone tree measure or the non-monotone entropy measure. From data mining point of view, monotonicity is essential. Namely, we should always prefer to generalize the entries of the database to as small sets as possible. On the other hand, from privacy point of view the entropy measure seems more appropriate, since the generalization $4 \to \{3,4\}$ in the above example reveals critical information and hence it should be penalized more than suppressing $4 \to *$. However, as explained in the introduction, we address the privacy concerns by respecting k-anonymity.

We believe that this is not a critical argument against the entropy measure since in practice such anomalies are rare. Besides, in practice one would usually use restricted generalization and then it is possible to define the collection of subsets of a given attribute, $\overline{A}$, so that the entropy measure is monotone on that collection. Assume, for example, that $\overline{A}$ is proper. Then, by Lemma 2.3, it may be represented by a hierarchical clustering tree. Then if the entropy measure is not monotone with respect to that collection (as in the example above), the following algorithm may be used to modify it into a (coarser) collection of subsets that does respect monotonicity.

1. *Look for an edge (B, B') in the tree, $B \supset B'$, where the conditional entropy of the attribute A with respect to B is smaller than its conditional entropy with respect to B'.*

2. *Unify the node B' with one of its siblings. If B' has only one sibling B'', remove those two nodes from the tree and connect the sons of both B' and B'' directly to B.*

3. *Repeat until the tree has no more edges that violate monotonicity.*

This algorithm clearly terminates with a tree that respects monotonicity, since if we keep unifying nodes in the tree in the manner described above, we will end up with the trivial tree with two levels that corresponds to generalization by suppression, and that tree obviously respects monotonicity.

3.3 The non-uniform entropy measure

Another problem with the entropy measure is that it is uniform for all records in the same cluster. Consider, for instance, the setting in Example 3.5. If we replace the two attribute values 1 and 2 with the generalized subset $\{1,2\}$ then the entropy measure for the information loss will be the same in all records that have one of those two values. However, the value 1 is much more frequent than the value 2. Hence, a more careful measure of information loss would indicate that the amount of information lost in the rare records with the value 2 is much larger than that in the more frequent records with the value 1.

To this end we define the following alternative measure, to which we refer as the *non-uniform entropy measure*.

Definition 3.6 *Let $D = \{R_1, \ldots, R_n\}$ be a database having public attributes A_j, $1 \le j \le r$, and*

Generalization	Entropy (Π_{e})	Non-uniform entropy (Π_{ne})
$1 \mapsto \{1,2\}$	$h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right) = O\left(\varepsilon \log \varepsilon^{-1}\right)$	$-\log \frac{1-3\varepsilon}{1-2\varepsilon} = O(\varepsilon)$
$2 \mapsto \{1,2\}$	$h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right) = O\left(\varepsilon \log \varepsilon^{-1}\right)$	$-\log \frac{\varepsilon}{1-2\varepsilon} = O\left(\log \varepsilon^{-1}\right)$
$3 \mapsto \{3,4\}$	$h\left(\frac{1}{2}, \frac{1}{2}\right) = 1$	$-\log \frac{1}{2} = 1$
$4 \mapsto \{3,4\}$	$h\left(\frac{1}{2}, \frac{1}{2}\right) = 1$	$-\log \frac{1}{2} = 1$

Table 1: Partial generalization.

Generalization	Entropy (Π_{e})	Non-uniform entropy (Π_{ne})
$1 \mapsto *$	$h\left(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon\right) = O\left(\varepsilon \log \varepsilon^{-1}\right)$	$-\log(1-3\varepsilon) = O(\varepsilon)$
$2 \mapsto *$	$h\left(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon\right) = O\left(\varepsilon \log \varepsilon^{-1}\right)$	$\log \varepsilon^{-1}$
$3 \mapsto *$	$h\left(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon\right) = O\left(\varepsilon \log \varepsilon^{-1}\right)$	$\log \varepsilon^{-1}$
$4 \mapsto *$	$h\left(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon\right) = O\left(\varepsilon \log \varepsilon^{-1}\right)$	$\log \varepsilon^{-1}$

Table 2: Generalization by suppression.

let $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ *be a generalization of* D*. Then*

$$\Pi_{ne}(D, g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{r} -\log \Pr(R_i(j) | \overline{R}_i(j)) \tag{5}$$

is the non-uniform entropy measure of the loss of information caused by generalizing D *into* $g(D)$*.*

Lemma 3.7 *The non-uniform entropy measure is monotone.*

Proof. Let X be a random variable that takes values in A and let $a \in A_1 \subseteq A_2 \subseteq A$. Then $\Pr(X = a | X \in A_1) \geq \Pr(X = a | X \in A_2)$. The monotonicity of Π_{ne} immediately follows. □

Let us exemplify the non-uniform entropy measure and compare it to the entropy measure on Example 3.5. In Table 1 we present the two measures of loss of information per record in case of partial generalization, while in Table 2 we present the corresponding data for the case of total generalization (namely, suppression) of the same database. By comparing the right-most columns in both tables we see that the values in Table 1 are less than or equal to those in Table 2, thus demonstrating the monotonicity of the non-uniform entropy measure as stated in Lemma 3.7. By comparing the first two rows in both tables we see that the non-uniform measure treats differently the values 1 and 2 and penalizes more the generalization of rare values.

While Tables 1 and 2 presented the costs per record, the overall cost is computed as the sum of costs over all records, see (4) and (5). Assume that we generalized all entries in the database by partial generalization (Table 1). Since the database holds $(1-3\varepsilon)n$ records with the value 1 and εn records with each of the values 2, 3 and 4, then the overall entropy measure of loss of information is

$$\Pi_{\mathrm{e}}(D, g(D)) = n \cdot \left[(1-2\varepsilon) h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right) + 2\varepsilon\right], \tag{6}$$

while

$$\Pi_{\mathrm{ne}}(D, g(D)) = n \cdot \left[(1-3\varepsilon) \log \frac{1-2\varepsilon}{1-3\varepsilon} + \varepsilon \log \frac{1-2\varepsilon}{\varepsilon} + 2\varepsilon \right]. \tag{7}$$

As can be easily seen, the two values in (6) and (7) coincide. The same coincidence occurs also in the case of generalization by suppression. This is no coincidence.

Lemma 3.8 *Let D be a database and let $g(D)$ be a generalization of D where for all $1 \leq i < i' \leq n$ and for all $1 \leq j \leq r$, either $g(D)_i(j) = g(D)_{i'}(j)$ or $g(D)_i(j) \cap g(D)_{i'}(j) = \emptyset$. Then $\Pi_e(D, g(D)) = \Pi_{ne}(D, g(D))$.*

Proof. For the sake of simplicity we assume that $r = 1$, namely, that the database D has only one attribute $A = \{a_1, \ldots, a_m\}$. The case of $r > 1$ trivially follows by adding up the contributions from all attributes.

By assumption, the entries in the generalized database $g(D)$ are disjoint subsets of A. Denote the subsets of A that appear in $g(D)$ by $B_1, \ldots, B_t$. Then each value $a_\ell \in A$, $1 \leq \ell \leq m$, that appears in D is generalized to a unique subset $B_{h(\ell)}$ where $1 \leq h(\ell) \leq t$.

Let X be the value of the attribute A in a randomly selected record in D. Define $p_\ell = \Pr(X = a_\ell)$, $q_j = \sum_{a_\ell \in B_j} p_\ell$, and $p'_\ell = p_\ell / q_{h(\ell)}$, for all $1 \leq \ell \leq m$, $1 \leq j \leq t$. This implies that

$$H(X|B_j) = \sum_{a_\ell \in B_j} p'_\ell \log \frac{1}{p'_\ell} = \sum_{a_\ell \in B_j} \frac{p_\ell}{q_j} \log \frac{q_j}{p_\ell}, \quad 1 \leq j \leq t\,.$$

Consequently, the entropy measure of information loss is

$$\Pi_{\mathrm{e}}(D, g(D)) = \sum_{j=1}^{t} q_j n \cdot H(X|B_j) = \sum_{j=1}^{t} q_j n \cdot \sum_{a_\ell \in B_j} \frac{p_\ell}{q_j} \log \frac{q_j}{p_\ell} = n \sum_{\ell=1}^{m} p_\ell \log \frac{q_{h(\ell)}}{p_\ell},$$

which is precisely the value of the non-uniform entropy measure $\Pi_{\mathrm{ne}}(D, g(D))$. □

The condition in Lemma 3.8 is violated when one of the columns in the generalized database includes intersecting entries. This is the case, for example, with generalization by suppression, when some entries in a given column were suppressed while some others were not. In such cases, the two measures Π_{e} and Π_{ne} might differ.

4 k-anonymization with minimal loss of data

We are now ready to define the concepts of k-anonymization and the corresponding problem of k-anonymization with minimal loss of information.

Definition 4.1 *A k-anonymization of a database $D = \{R_1, \ldots, R_n\}$ is a generalization $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ where for all $1 \leq i \leq n$, there exist indices $1 \leq i_1 < i_2 < \cdots < i_{k-1} \leq n$, all of which are different from i, such that $\overline{R}_i = \overline{R}_{i_1} = \cdots = \overline{R}_{i_{k-1}}$.*

k-ANONYMIZATION: Let $D = \{R_1, \ldots, R_n\}$ be a database having public attributes A_j, $1 \le j \le r$. Given collections of attribute values, $\overline{A}_j \subseteq \mathcal{P}(A_j)$, $1 \le j \le r$, and a measure of information loss Π, find a k-anonymization $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$, where $\overline{R}_i \in \overline{A}_1 \times \cdots \times \overline{A}_r$, $1 \le i \le n$, that minimizes $\Pi(D, g(D))$.

The following theorem and its proof are an adaptation of [MW04, Theorem 3.1]. We show that the problem of k-anonymization with minimal loss of information is NP-hard with respect to each of the two proposed entropy measures.

Theorem 4.2 *The problem of k-ANONYMIZATION with generalization by suppression, where the measure of loss of information is either the entropy measure (4), $\Pi = \Pi_e$, or the non-uniform entropy measure (5), $\Pi = \Pi_{ne}$, is NP-hard for $k \ge 3$, if $|A_j| \ge n + 1$ for all $1 \le j \le r$.*

Proof. The reduction is from k-DIMENSIONAL PERFECT MATCHING: given a simple hypergraph $H = (U, E)$ where $|U| = n$, $|e| = k$ for all $e \in E$, and $k|n$, is there a subset $S \subset E$ of n/k hyperedges such that $\bigcup_{e \in S} = U$?

Let $U = \{u_1, \ldots, u_n\}$ and $E = \{e_1, \ldots, e_m\}$. Define m identical sets of attributes $A_j = \{0, 1, \ldots, n\}$, $1 \le j \le m$, and a database $D = \{R_1, \ldots, R_n\}$ where

$$R_i(j) = \begin{cases} 0 & \text{if } u_i \in e_j \\ i & \text{otherwise} \end{cases}, \qquad 1 \le i \le n,\ 1 \le j \le m\,.$$

In order to prove the NP-hardness with respect to the entropy measure, we claim that there is a k-dimensional perfect matching if and only if there exists a k-anonymization $g(D)$ for which

$$\Pi_{\mathrm{e}}(D, g(D)) \le n(m-1) \cdot H_{k,n} \tag{8}$$

where $H_{k,n}$ is the entropy of each of the attributes. Since each of the attributes attains the value 0 in probability $\frac{k}{n}$ and $n - k$ additional values from $\{1, \ldots, n\}$, each of which in probability $\frac{1}{n}$, we get that

$$H_{k,n} = \frac{k}{n} \log \frac{n}{k} + \frac{n-k}{n} \log n\,.$$

Assume first that there exists a k-dimensional perfect matching, $S \subset E$, and define the generalization $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ where

$$\overline{R}_i(j) = \begin{cases} 0 & \text{if } u_i \in e_j \text{ and } e_j \in S \\ * & \text{otherwise} \end{cases}\,. \tag{9}$$

If $e \in S$ and $u_i, u_{i'} \in e$, then $\overline{R}_i = \overline{R}_{i'}$. Hence, since $|e| = k$ for all $e \in E$, the above generalization is indeed a k-anonymization. As each suppressed entry incurs a loss of information of $H_{k,n}$ and there are $n(m-1)$ such entries, we see that $g(D)$ satisfies (8).

In order to prove the converse, we first observe that each record $\overline{R}_i$ in $g(D)$ can have at most one non-* entry. Indeed, by the definition of D, all non-* entries must be zero. Therefore, if $\overline{R}_i$ has two non-* entries, say $\overline{R}_i(j) = \overline{R}_i(j') = 0$, then by k-anonymity, there exist at least k rows that have zeros in the columns that correspond to e_j and $e_{j'}$. This implies that e_j and $e_{j'}$ coincide, in contradiction to our assumption that H is simple.

Conversely, assume there exists a k-anonymization $g(D)$ that satisfies (8). Hence, there are at most $n(m-1)$ generalized entries in that k-anonymization. Since each record $\overline{R}_i$ in $g(D)$ has at most one non-* entry, we conclude that each $\overline{R}_i$ has exactly one non-* entry. Those non-* entries must be zero, since if $\overline{R}_i = \overline{R}_{i'}$ their non-* entry must be zero. Therefore, each cluster of identical rows corresponds to a cluster of nodes that belong to the same hyperedge $e \in E$. This implies that each of the clusters of identical rows in $g(D)$ have size k. Those clusters give rise to a k-dimensional perfect matching of H.

In order to prove the NP-hardness with respect to the non-uniform entropy measure, we claim that there is a k-dimensional perfect matching if and only if there exists a k-anonymization $g(D)$ for which

$$\Pi_{\mathrm{ne}}(D, g(D)) \leq (nm - t)\log n + (t - n)\log\frac{n}{k} \tag{10}$$

where $t = \sum_{i=1}^{n} t_i$ and t_i is the degree of $u_i \in U$ in the hypergraph H (namely, t_i is the number of hyperedges $e \in E$ such that $u_i \in e$).

Assume first that there is a k-dimensional perfect matching, $S \subset E$, and define the same k-anonymization as in (9). In each record R_i, $1 \leq i \leq n$, there are two types of suppressed entries. Entries $R_i(j) = 0$ that correspond to hyperedges e_j such that $u_i \in e_j \notin S$, and entries $R_i(j) = j$ that correspond to to hyperedges e_j such that $u_i \notin e_j$. There are $t_i - 1$ entries of the first type, each of which incurs a cost of $\log\frac{n}{k}$, and $m - t_i$ entries of the second type, each of which incurs a cost of $\log n$. Hence, the k-anonymization in (9) satisfies

$$\Pi_{\mathrm{ne}}(D, g(D)) = \sum_{i=1}^{n}\left[(m - t_i)\log n + (t_i - 1)\log\frac{n}{k}\right] =$$

$$(nm - t)\log n + (t - n)\log\frac{n}{k}\,,$$

in accord with (10).

Finally, assume that there exists a k-anonymization $g(D)$ that satisfies (10). We showed earlier that each record $\overline{R}_i$ in $g(D)$ can have at most one non-* entry and that such non-generalized entries must be zero. On the other hand, each $\overline{R}_i$ must have at least one non-generalized entry, for otherwise the amount of information lost would exceed the value on the right hand side of (10). As shown earlier, such a k-anonymization defines a k-dimensional perfect matching of H. □

5 Approximating optimal k-anonymity

In this section we describe two approximation algorithms for the problem of k-anonymization with minimal loss of information. We concentrate on approximating optimal k-anonymity with respect to the entropy measure. The first algorithm, described in Sections 5.1-5.3, achieves an approximation ratio of $O(\ln k)$ — a significant improvement over the algorithm due to Aggarwal et al. [AFK$^+$05] that offers an approximation ratio of $O(k)$. In Section 5.1 we define the key notion of the *generalization cost* of a set of records and compare it to the related notion of the *diameter* of such sets that played an important role in the approximation algorithm of [MW04]. In Section 5.2 we explore the relations between k-anonymizations, clusterings and covers of a given database D. Using these relations, we describe in Section 5.3 an approximation algorithm for optimal k-anonymization that uses an approximation algorithm for the problem of finding a

minimum-weight cover. The algorithm of Section 5.3 runs in time $O(n^{2k})$. In Section 5.4 we discuss another approximation algorithm that is strongly polynomial. We show that the $O(k)$-approximation algorithm of [AFK+05] that runs in time $O(kn^2)$ may be used also for approximating optimal k-anonymity when using the entropy measure. The question of the existence of a strongly polynomial approximation algorithm with a logarithmic approximation ratio remains open. It also remains open to find an approximation algorithm, strongly polynomial or not, for the non-uniform entropy measure.

5.1 The generalization cost of subsets

Any k-anonymization of D defines a clustering (namely, a partition) of D where each cluster consists of all records that were replaced by the same generalized record. In order to lose a minimal amount of information, all records in the same cluster are replaced with the minimal generalized record that generalizes all of them. To that end we define the closure of a set of records[1].

Definition 5.1 *Let $A_1, \ldots, A_r$ be attributes with corresponding collections of subsets $\overline{A}_1, \ldots \overline{A}_r$ that are all proper. Then given $M \subseteq A_1 \times \cdots \times A_r$, its closure is defined as*

$$\overline{M} = \min_{\sqsubseteq} \left\{ C \in \overline{A}_1 \times \cdots \times \overline{A}_r : R \sqsubseteq C \text{ for all } R \in M \right\} .$$

Definition 5.2 *Let $D = \{R_1, \ldots, R_n\}$ be a database with attributes $A_1, \ldots, A_r$, having proper collections of subsets $\overline{A}_1, \ldots \overline{A}_r$. Let X_j be the value of the attribute A_j in a randomly selected record from D. Then given a subset of records, $M \subseteq D$, we define its generalization cost by the entropy measure as follows,*

$$d(M) = \sum_{j=1}^{r} H(X_j | \overline{M}_j) \,. \tag{11}$$

The generalization cost of M is therefore the amount of information that we lose for each record $R \in M$ if we replace it by the minimal generalized record $\overline{M}$.

We noted earlier that the entropy measure is not necessarily monotone in the sense of Definition 3.4. Hence, it is possible that for a given set M there exists a record $C \in \overline{A}_1 \times \cdots \times \overline{A}_r$ that dominates the closure of M, i.e., $\overline{M} \sqsubseteq C$, but $\sum_{j=1}^{r} H(X_j|\overline{M}_j) \geq \sum_{j=1}^{r} H(X_j|C_j)$. Namely, for such a set M it is better to replace all records in M with the generalized record C and not with $\overline{M}$. As noted earlier, we may always avoid this problem by narrowing down the collections $\overline{A}_j$, $1 \leq j \leq r$, until the entropy measure becomes monotone with respect to them. For the sake of simplicity, we assume monotonicity hereinafter. Namely,

$$M \subseteq M' \subseteq A_1 \times \cdots \times A_r \text{ implies that } d(M) \leq d(M') \,. \tag{12}$$

The notion of the generalization cost of a set of records is related to the notion of the *diameter* of such a set, as defined in [MW04]. The diameter of a set of records $M \subseteq A_1 \times \cdots \times A_r$ was defined as

$$\mathrm{diam}(M) = \max_{R,R' \in M} \mathrm{dist}(R, R'), \quad \text{where } \mathrm{dist}(R, R') = |\{1 \leq j \leq r : R(j) \neq R'(j)\}| \,. \tag{13}$$

[1] In our discussion, a *set* actually means a *multiset*; namely, it may include repeated elements.

In other words, if the two records R and R' were to be generalized by means of suppression, $\mathrm{dist}(R, R')$ equals the minimal number of attributes that would be suppressed in each of the two records in order to make them identical. The two notions, (11) and (13), are functions that associate a *size* to a given set of records. Our notion, though, of generalization cost, improves that of the diameter as follows:

1. The generalization cost, (11), generalizes the definition of the diameter, (13), in the sense that it applies to any type of generalization (the definition of the diameter is restricted to generalization by suppression).

2. The notion of the generalization cost uses the more accurate entropy measure (the definition of diameter only counts the number of suppressed entries).

3. Most importantly, while the size of a set of records that is defined in (13) is a *diameter* (namely, it is based on pairwise distances), the size that is defined in (11) is a *volume*. Both notions offer measures for the amount of information that is lost if the entire set of records, M, is to be anonymized in the same way. But while the diameter does this by only looking at pairs of records in M, the generalization cost does this by looking simultaneously at all records in M and computing the information loss that their closure entails. This simple difference turns out to be of paramount importance, as we show below.

Before moving on, we prove the following basic lemma that will be needed for our later analysis.

Lemma 5.3 *Assume that all collections of subsets, $\overline{A}_j$, $1 \leq j \leq r$, are proper. Then the generalization cost $d(\cdot)$ is sub-additive in the sense that for all $S, T \subseteq A_1 \times \cdots \times A_r$,*

$$S \cap T \neq \emptyset \text{ implies that } d(S \cup T) \leq d(S) + d(T)\,. \tag{14}$$

Proof. Denote $U = S \cup T$ and let

$$S_j = \{s(j) : s \in S\}, \quad T_j = \{t(j) : t \in T\}, \quad U_j = \{u(j) : u \in U\}$$

denote the set of values of the j-th attribute, $1 \leq j \leq r$, that appear in S, T, and U, respectively. Let $\overline{S}_j$, $\overline{T}_j$ and $\overline{U}_j$ be the minimal sets in $\overline{A}_j$ that include S_j, T_j and U_j, respectively. Since $S \cap T \neq \emptyset$, we conclude that $S_j \cap T_j \neq \emptyset$. Hence $\overline{S}_j \cap \overline{T}_j \neq \emptyset$. But since $\overline{A}_j$ is proper, we have that $\overline{S}_j \subseteq \overline{T}_j$ or $\overline{T}_j \subseteq \overline{S}_j$. Therefore, $\overline{U}_j = \overline{S}_j$ or $\overline{U}_j = \overline{T}_j$, whence

$$H(X_j|\overline{U}_j) \leq H(X_j|\overline{S}_j) + H(X_j|\overline{T}_j)\,. \tag{15}$$

Summing (15) for all $1 \leq j \leq r$ we arrive at (14). □

Lemma 5.3 does not necessarily hold for generalizations that are not proper. As a simple example, consider the case of one attribute ($r = 1$), where $A_1 = \{1, 2, 3\}$, and $\overline{A}_1 = \mathcal{P}(A_1)$ (note that a generalization that allows any subset of attribute values is indeed non-proper). Let $S_1 = \{1, 2\}$, $T_1 = \{2, 3\}$, and assume that $\Pr(X_1 = 1) = \frac{1}{2} - \varepsilon$, $\Pr(X_1 = 2) = 2\varepsilon$, and $\Pr(X_1 = 3) = \frac{1}{2} - \varepsilon$. Then

$$d(S \cup T) = H(X_1) \approx 1\,,$$

while

$$d(S) = H(X_1|S_1) \approx 0, \quad d(T) = H(X_1|T_1) \approx 0\,.$$

Hence, in this case $d(S \cup T) > d(S) + d(T)$.

5.2 Covers, clusterings, k-anonymizations and their generalization cost

As noted earlier, any k-anonymization of D defines a clustering of D. Without loss of generality, we may assume that all clusters are of sizes between k and $2k-1$; indeed, owing to monotonicity, any cluster of size greater than $2k$ may be split into clusters of sizes in the range $[k, 2k-1]$ without increasing the amount of information loss due to k-anonymization. Let:

1. $\mathcal{G}$ be the family of all k-anonymizations of D, where the corresponding clusters are of sizes in the range $[k, 2k-1]$.
2. Γ be the family of all covers of D by subsets of sizes in the range $[k, 2k-1]$.
3. $\Gamma^0 \subset \Gamma$ be the family of all covers in Γ that are clusterings (or partitions); namely, all covers in Γ consisting of non-intersecting subsets.

There is a natural one-to-one correspondence between $\mathcal{G}$ and Γ^0.

Given a cover $\gamma \in \Gamma$, we define its generalization cost as follows:

$$d(\gamma) = \sum_{S \in \gamma} d(S)\,, \tag{16}$$

where $d(\cdot)$ is as in Definition 5.2. This cost is closely related to the measure of loss of information by k-anonymization, as stated in the next lemma.

Lemma 5.4 *Let $g \in \mathcal{G}$ be a k-anonymization of D and let $\gamma^0 \in \Gamma^0$ be its corresponding clustering of D. Then*

$$k \cdot d(\gamma^0) \leq \Pi_e(D, g(D)) \leq (2k-1) \cdot d(\gamma^0)\,. \tag{17}$$

Proof. As we have

$$k \leq |S| \leq 2k-1, \text{ for all } S \in \gamma^0, \tag{18}$$

and

$$\Pi_{\mathrm{e}}(D, g(D)) = \sum_{S \in \gamma^0} |S| \cdot d(S), \tag{19}$$

inequality (17) follows from (19), (18) and (16). □

Next, we claim the following:

Theorem 5.5 *Let $\hat{\gamma}$ be a cover that achieves minimal generalization cost $d(\cdot)$ in Γ. Let $g \in \mathcal{G}$ be a k-anonymization and let $\gamma^0 \in \Gamma^0$ be its corresponding clustering. Then*

$$\Pi_e(D, g(D)) \leq \frac{2d(\gamma^0)}{d(\hat{\gamma})} \cdot OPT_e(D), \tag{20}$$

where

$$OPT_e(D) := \min_{g \in \mathcal{G}} \Pi_e(D, g(D)). \tag{21}$$

Proof. Let g^* be a k-anonymization for which $OPT_{\mathrm{e}}(D) = \Pi_{\mathrm{e}}(D, g^*(D))$ and let γ^* be its corresponding clustering. On one hand, by the lower bound in (17) and the definition of $\hat{\gamma}$,

$$OPT_{\mathrm{e}}(D) = \Pi_{\mathrm{e}}(D, g^*(D)) \geq k \cdot d(\gamma^*) \geq k \cdot d(\hat{\gamma}). \tag{22}$$

On the other hand, by the upper bound in (17),

$$\Pi_{\mathrm{e}}(D, g(D)) \leq (2k-1) \cdot d(\gamma^0). \tag{23}$$

Hence, by (23) and (22),

$$\Pi_{\mathrm{e}}(D, g(D)) \leq \frac{2k-1}{k} \cdot \frac{d(\gamma^0)}{d(\hat{\gamma})} \cdot OPT_{\mathrm{e}}(D) \leq \frac{2d(\gamma^0)}{d(\hat{\gamma})} \cdot OPT_{\mathrm{e}}(D)\,.$$

□

5.3 Approximating optimal k-anonymization

Our approximation algorithm follows the algorithm of [MW04]. It has two phases, as described hereinafter.

Phase 1: Producing a cover. Let $\hat{\gamma}$ be a cover that minimizes $d(\cdot)$ in Γ. In the first phase of the algorithm we execute the greedy algorithm for approximating the WEIGHTED SET COVER problem [Joh74].

1. Set $\mathcal{C}$ to be the collection of all subsets of D with cardinality in the range $[k, 2k-1]$. Each set S is associated with a cost $d(S)$. Also set $\gamma = \emptyset$ and $E = \emptyset$.
2. While $E \neq D$ do:
   - For each $S \in \mathcal{C}$ compute the ratio $r(S) = d(S)/|S \cap (D \setminus E)|$.
   - Choose S that minimizes $r(S)$.
   - $E = E \cup S$, $\gamma = \gamma \cup \{S\}$, $\mathcal{C} = \mathcal{C} \setminus \{S\}$.
3. Output γ.

The result of that phase is a cover $\gamma \in \Gamma$ for which

$$d(\gamma) \leq (1 + \ln 2k) d(\hat{\gamma})\,, \tag{24}$$

see [Chv79].

Phase 2: Translating the cover into a k-anonymization. In the second phase we translate the cover $\gamma \in \Gamma$ to a clustering $\gamma^0 \in \Gamma^0$ and then to its corresponding k-anonymization $g \in \mathcal{G}$. The translation procedure works as follows:

1. Input: $\gamma = \{S_1, \ldots, S_t\}$, a cover of $D = \{R_1, \ldots, R_n\}$.
2. Set $\gamma^0 = \gamma$.

3. Repeat until the cover γ^0 has no intersecting subsets:
   - Let $S_j, S_\ell \in \gamma^0$ be such that $S_j \cap S_\ell \neq \emptyset$ and let R be a record in D that belongs to $S_j \cap S_\ell$.
   - If $|S_j| > k$ set $S_j = S_j \setminus \{R\}$.
   - Else, if $|S_\ell| > k$ set $S_\ell = S_\ell \setminus \{R\}$.
   - Else (namely, if $|S_j| = |S_\ell| = k$) remove S_ℓ from γ^0 and set $S_j = S_j \cup S_\ell$.
4. Output the following k-anonymization: For $i = 1, \ldots, n$, look for $S_j \in \gamma^0$ such that $R_i \in S_j$ and then set $g(D)_i = \overline{S}_j$.

Theorem 5.6 *The k-anonymization g that is produced by the above described algorithm satisfies*

$$\Pi_e(D, g(D)) \leq 2(1 + \ln 2k) \cdot OPT_e(D)\,, \tag{25}$$

where $OPT_e(D)$ is the cost of an optimal k-anonymization, (21).

Proof. First, we observe that $d(\gamma^0) \leq d(\gamma)$, as implied by our monotonicity assumption, (12), and by Lemma 5.3. Hence, by (24), $d(\gamma^0) \leq (1 + \ln 2k) d(\hat{\gamma})$. Finally, by Theorem 5.5, the k-anonymization g satisfies (25). □

The corresponding result in [MW04] is Theorem 4.1 there, according to which the approximation algorithm achieves an approximation factor of $3k \cdot (1 + \ln 2k)$. Aggarwal et al. proposed an improved approximation algorithm that achieves an $O(k)$ approximation factor [AFK$^+$05, Theorem 5]. The approximation algorithms in both [MW04] and [AFK$^+$05] were based on the so-called *graph representation*. In that approach, the records of D are viewed as nodes of a complete graph, where the weight of each edge (R_i, R_j) is the generalization cost of the set $\{R_i, R_j\}$. Both algorithms work with such a graph representation and find the approximate k-anonymization based only on the information that is encoded in that graph. Such an approach is limited since it uses only the distances between pairs of nodes. In [AFK$^+$05] it was shown that using the graph representation it is impossible to achieve an approximation ratio that is better than $\Theta(k)$.

We were able to offer the significantly better $O(\ln k)$ approximation ratio by breaking out of the graph representation framework. As explained in Section 5.1, our cost function $d(\cdot)$ is defined for sets of records, rather than pairs of records. Hence, it represents *volume* rather than a *diameter*. This upgrade from the graph representation to a hypergraph representation enabled the improvement from a linear approximation ratio to a logarithmic one.

Finally we note that the algorithm described in this section runs in time $O(n^{2k})$. Such a non-strongly polynomial running time is due to the fact that we examine all subsets of records of D with cardinalities between k and $2k - 1$.

5.4 A strongly polynomial approximating algorithm

5.4.1 Preliminaries

We describe here the algorithm due to Aggarwal et al. [AFK$^+$05] for approximating optimal k-anonymization and we show that it may be applied also to the entropy measure.

Let $D = \{R_1, \ldots, R_n\}$ be a database having public attributes A_j, $1 \leq j \leq r$, and assume that all collections of subsets, $\overline{A}_j$, $1 \leq j \leq r$, are proper. Such a database may be represented by a graph.

Definition 5.7 *The graph representation for the database $D = \{R_1, \ldots, R_n\}$ is the complete weighted graph $G = (V, E)$ where $V = D$, $E = \{e_{i,j} = \{R_i, R_j\} : 1 \leq i < j \leq n\}$, and $w(e_{i,j}) = d(\{R_i, R_j\})$, where $d(\cdot)$ is the generalization cost by the entropy measure, (11).*

Let $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_s\}$ be a spanning forest of G; namely, each $\mathcal{T}_j$ is a tree in G and every node R_i, $1 \leq i \leq n$, belongs to exactly one tree $\mathcal{T}_{j(i)} \in \mathcal{F}$. If all tress in that forest are of size at least k then that forest induces a k-anonymization of D, denoted $g_{\mathcal{F}}$. The charge of each node with respect to $g_{\mathcal{F}}$ is defined as $c(R_i, g_{\mathcal{F}}) = d(\mathcal{T}_{j(i)})$, where $d(\cdot)$ is the generalization cost by the entropy measure. The generalization cost of $g_{\mathcal{F}}$ is then

$$\Pi_{\mathrm{e}}(D, g_{\mathcal{F}}(D)) = \sum_{i=1}^{n} c(R_i, g_{\mathcal{F}})\,. \tag{26}$$

An important observation in designing the algorithm is the following.

Lemma 5.8 *Let $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_s\}$ be a spanning forest of G, and let $g_{\mathcal{F}}$ be its corresponding anonymization. Then the charge of each node with respect to that anonymization is bounded by the sum of weights of all edges in the tree to which that node belongs:*

$$c(R_i, g_{\mathcal{F}}) \leq w(\mathcal{T}_{j(i)}) := \sum_{e \in \mathcal{T}_{j(i)}} w(e)\,. \tag{27}$$

Proof. We need to prove that for any given tree, $\mathcal{T}$, we have $d(\mathcal{T}) \leq w(\mathcal{T})$, where $d(\mathcal{T})$ is the generalization cost of $\mathcal{T}$ by the entropy measure and $w(\mathcal{T})$ is the sum of weights all edges in $\mathcal{T}$. We prove the claim by induction on the size of $\mathcal{T}$. If $|\mathcal{T}| \leq 2$ the claim is obviously true. Assume next that we proved the claim for all trees of size less than $|\mathcal{T}|$ and we proceed to prove it for $\mathcal{T}$. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two subtrees of $\mathcal{T}$ where $|\mathcal{T}_1 \cap \mathcal{T}_2| = 1$ and $\max\{|\mathcal{T}_1|, |\mathcal{T}_2|\} < |\mathcal{T}|$. Then by the sub-additivity of the generalization cost with respect to the entropy measure, Lemma 5.3,

$$d(\mathcal{T}) = d(\mathcal{T}_1 \cup \mathcal{T}_2) \leq d(\mathcal{T}_1) + d(\mathcal{T}_2)\,.$$

As the induction hypothesis applies to both $\mathcal{T}_1$ and $\mathcal{T}_2$ we infer that

$$d(\mathcal{T}_1) + d(\mathcal{T}_2) \leq w(\mathcal{T}_1) + w(\mathcal{T}_2) = w(\mathcal{T})\,,$$

thus proving the claim. □

We are now able to state the main result.

Theorem 5.9 *Let $OPT = OPT_e(D)$ be the cost of an optimal k-anonymization of D with respect to the entropy measure, and let L be an integer such that $L \geq k$. Let $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_s\}$ be a spanning forest of G whose total weight is at most OPT and in which each of the trees is of size in the range $[k, L]$. Then the corresponding k-anonymization, $g_{\mathcal{F}}$, is an L-approximation for the optimal k-anonymization, i.e.,*

$$\Pi_e(D, g_{\mathcal{F}}(D)) \leq L \cdot OPT\,.$$

Proof. Invoking (26), (27), and the fact that each node belongs to exactly one tree in the forest, we conclude that

$$\Pi_{\mathrm{e}}(D, g_{\mathcal{F}}(D)) = \sum_{i=1}^{n} c(R_i, g_{\mathcal{F}}) \leq \sum_{i=1}^{n} w(\mathcal{T}_{j(i)}) = \sum_{j=1}^{s} |\mathcal{T}_j| \cdot w(\mathcal{T}_j)\,.$$

Hence, since all trees are of size L at the most, we get that

$$\Pi_{\mathrm{e}}(D, g_{\mathcal{F}}(D)) \leq L \cdot \sum_{j=1}^{s} w(\mathcal{T}_j) \leq L \cdot OPT\,.$$

□

5.4.2 The approximation algorithm

The algorithm has two stages:

1. Stage 1: Create a spanning forest $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_s\}$ whose total weight is at most OPT and in which all trees are of size at least k.
2. Stage 2: Compute a decomposition of this forest such that each component has size in the range $[k, L]$ for $L = \max\{2k-1, 3k-5\}$.

The first stage constructs a directed forest where the out-degree of each node is at most one, and (R_i, R_j) is an edge in that forest only if R_j is one of the $k-1$ nearest neighbors of R_i.

Algorithm Forest

1. Set $\mathcal{F} = (V, E)$ where $V = D$ and $E = \emptyset$. We continue to add directed edges to the forest $\mathcal{F}$ while respecting two rules: the added edges contain no cycles and the out-degree of each node in the forest is at most one.
2. Repeat until all components (trees) of $\mathcal{F}$ have size at least k:
   - Pick a component $\mathcal{T}$ of $\mathcal{F}$ of size $|\mathcal{T}| < k$.
   - Let $R \in \mathcal{T}$ be a node without any outgoing edges.
   - As $|\mathcal{T} \setminus \{R\}| \leq k-2$, there exists a node R' outside $\mathcal{T}$ that is one of the $k-1$ nearest neighbors of R; find such a node and add to the forest the directed edge (R, R').

Lemma 5.10 *The forest produced by the algorithm* Forest *has minimum tree size k and has weight at most OPT, provided that the collections of subsets $\overline{A}_j$, $1 \leq j \leq r$, are such that the entropy measure is monotone with respect to them, (12).*

Proof. As the algorithm repeats adding edges to forest components of size less than k until the forest has no more such components, the first claim is obvious. As for the second claim, it follows from the assumed monotonicity of the entropy measure. Let g^* be an optimal k-anonymization, $\gamma^* =$

$\{S_1, \ldots, S_t\}$ be its corresponding clustering, and $c(R_i, g^*)$ be the charge of R_i in that anonymization. Then $c(R_i, g^*) = d(S_{j(i)})$ where $S_{j(i)} \in \gamma^*$ is the cluster to which R_i belongs. Let $\{R'_1, \ldots, R'_{k-1}\}$ be the $k-1$ nearest neighbors of R_i in the graph G and let $\{R''_1, \ldots, R''_{k-1}\}$ be the $k-1$ nearest neighbors of R_i in $S_{j(i)}$. Then, by monotonicity,

$$c(R_i, g^*) = d(S_{j(i)}) \geq d(\{R_i, R''_1, \ldots, R''_{k-1}\}) \geq \max_{1 \leq \ell \leq k-1} d(\{R_i, R''_\ell\}) \geq \max_{1 \leq \ell \leq k-1} d(\{R_i, R'_\ell\}) \,.$$

This implies that the charge of a node R_i in an optimal k-anonymization is greater than the weight of the edge that out-goes from R_i in the forest $\mathcal{F}$, if such an edge exists. Summing up over all edges we get that OPT is greater than or equal to the weight of the forest. □

The second stage operates on the forest that is output by the first stage and breaks every component of size greater than $L = \max\{2k-1, 3k-5\}$ to two components of size at least k. This is accomplished by applying algorithm DECOMPOSE-COMPONENT, that is described in [AFK+05, Section 4.2], to each such component, until no more components of size greater than L are left. We omit further details on that algorithm since it is a pure graph algorithm that does not depend on the underlying measure of loss of information. Both algorithms, FOREST and DECOMPOSE-COMPONENT, run in time $O(kn^2)$ so the overall running time is strongly polynomial.

6 Conclusions

In this paper we studied the problem of k-anonymization, and we proposed two information-theoretic measures that capture the amount of information that is lost during the anonymization process. Our measures are more general and more accurate than previous measures that were studied in the literature. We proved that the problem of finding the optimal k-anonymization of a database is NP-hard with respect to the proposed entropy measures.

We then continued to study the approximability of that problem, with respect to the entropy measure. First, we adapted the algorithm of Meyerson and Williams [MW04] and obtained an $O(\ln k)$-approximation guarantee. The same guarantee holds also for the previously proposed measures, thus, our result improves upon the best-known $O(k)$-approximation ratio obtained by Aggarwal et al. [AFK+05]. While the approximation algorithms of [AFK+05, MW04] relied on the so-called *graph representation* framework, which was shown in [AFK+05] to be limited to $\Omega(k)$-approximations, our algorithm relies on a novel *hypergraph representation* that enables the improvement in the approximation ratio from $O(k)$ to $O(\ln k)$. As the running time of our suggested algorithm is $O(n^{2k})$, we also showed how to adapt the algorithm of [AFK+05] in order to obtain a strongly polynomial approximation algorithm for our entropy measure with an $O(k)$-approximation guarantee.

Two main open problems remain. The first is to find a strongly polynomial approximation algorithm for the entropy measure, with performance guarantee better than $O(k)$. The second open problem is to design an approximation algorithm, strongly polynomial or not, for the non-uniform entropy measure. We expect that, in practice, our hypergraph-based algorithm works well for the non-uniform entropy measure too, since a good cover tends to contain disjoint sets, and, consequently, may be easily converted to a clustering. However, the main difficulty of proving an approximation guarantee in the case of the non-uniform entropy measure is that the *sub-additivity* property does not hold for the non-uniform entropy measure, whence it is not clear how to convert a cover to a clustering without increasing the cost of the solution.

References

[AA01] D. Agrawal and C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of 20th ACM–SIGMOD Symposium on Principles of Database Systems (PODS)*, 2001.

[AFK+05] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. In *Proceedings of 10th International Conference on Database Theory (ICDT)*, 2005.

[AMP04] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth-ranked element. In *Advances in Cryptology: Proceedings of Eurocrypt*, 2004.

[AS00] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of 19th ACM–SIGMOD International Conference on Management of Data*, 2000.

[AST05] R. Agrawal, R. Srikant, and D. Thomas. Privacy preserving OLAP. In *Proceedings of 24th ACM–SIGMOD International Conference on Management of Data*, 2005.

[BDMN05] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *Proceedings of 24th ACM–SIGMOD Symposium on Principles of Database Systems (PODS)*, 2005.

[CDM+05] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Proceedings of 2nd Theory of Cryptography Conference*, 2005.

[Chv79] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[DN03] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of 22nd ACM–SIGMOD Symposium on Principles of Database Systems (PODS)*, 2003.

[DN04] C. Dwork and K. Nissim. Privacy-preserving data mining on vertically partitioned databases. In *Advances in Cryptology: Proceedings of Crypto*, 2004.

[DW99] A. G. DeWaal and L. C. R. J. Willenborg. Information loss through global recoding and local suppression. *Netherlands Official Statistics, Special issue on SDC*, 14:17–20, 1999.

[EGS03] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of 22nd ACM–SIGMOD Symposium on Principles of Database Systems (PODS)*, 2003.

[FNP04] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology: Proceedings of Eurocrypt*, 2004.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of 19th Annual ACM Symposium on Theory of Computing*, 1987.

[Joh74] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[KMN05] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. In *Proceedings of 24th ACM–SIGMOD Symposium on Principles of Database Systems (PODS)*, 2005.

[KPR03] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. *Journal of Computer and System Sciences*, 6:244–253, 2003.

[LP02] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3):177–206, 2002.

[MW04] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of 23rd ACM–SIGMOD Symposium on Principles of Database Systems (PODS)*, 2004.

[Sam01] P. Samarati. Protecting respondent's privacy in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13:1010–1027, 2001.

[SS98] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of 17th ACM–SIGMOD Symposium on Principles of Database Systems*, 1998.

[Swe02] L. Sweeney. k-Anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[WD01] L. Willenborg and T. DeWaal. *Elements of Statistical Disclosure Control*. Springer-Verlag, New York, 2001.

[Yao86] A. Yao. How to generate and exchange secrets. In *Proceedings of 27th IEEE Symposium on Foundations of Computer Science*, 1986.