k-Anonymization with Minimal Loss of Information

Aristides Gionis [*] Tamir Tassa [†]

June 5, 2007

Abstract

The technique of k-anonymization allows the releasing of databases that contain personal information while ensuring some degree of individual privacy. Anonymization is usually performed by 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. The proposed measures are more general and more accurate than those that were proposed by Meyerson and Williams [22] and Aggarwal et al. [1]. We study the problem of achieving k-anonymity with minimal loss of information. We prove that it is NP-hard and study polynomial approximations for the optimal solution. Our first algorithm gives an approximation guarantee of $O(\ln k)$ for our measures as well as for the previously studied measures. This improves the best-known $O(k)$-approximation of [1]. While the previous approximation algorithms relied on the *graph representation* framework, 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 [1] in order to obtain an $O(k)$-approximation algorithm that is polynomial in both n and k.

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:

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

and

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

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,

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

The corresponding private database holds the private information,

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

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, $1 \le i \le n$, 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

[*] Yahoo! Research, Barcelona, Spain, gionis@yahoo-inc.com

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

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.)

Back in 1977, Dalenius [9] articulated a desideratum for database security, saying that any information that may be extracted from a statistical database about an individual could also be learnt without an access to that database. That notion of security is similar to the notion of semantic security for cryptosystems, as defined by Goldwasser and Micali [17]. Alas, while semantic security for cryptosystems may be achieved, Dalenius' idealized goal may not be achieved [12]. Hence, a more realistic goal of privacy is to limit the risk to one's privacy as a result of one's participation in a statistical database.

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 [11, 19, 20], output perturbation [6, 11, 13], secure multi-party computation [2, 15, 16, 21, 27], and data sanitization [3, 4, 5, 7, 14]. One of the recent approaches, proposed by Samarati and Sweeney [23, 24, 25] 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 [22] 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 fully polynomial running time (namely, it depends polynomially on both n and k) and guarantees an approximation ratio of $O(k \ln n)$. Aggarwal et al. [1] 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 [1] (the measure that was used in [22] is a special case of the one that was used in [1]). We call these measures *the entropy measure* and *the monotone entropy measure*. We show that the problem of k-anonymization with minimal loss of data (measured by either of those measures) is NP-hard. We then proceed to describe an approximation algorithm with an approximation guarantee of $O(\ln k)$—a significant improvement over the previous best result of $O(k)$. The algorithm applies to both of our measures, as well as the measures that were used in [22] and [1]. We note that Meyerson and Williams [22] 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 [22] and [1] were based on the so-called *graph representation*. In [1] 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 define and discuss our two measures of loss of information. 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 and monotone entropy measures. The running time of that algorithm is $O(n^{2k})$. We then proceed to describe how to adapt the approximation algorithm of [1] to achieve an $O(k)$-approximation ratio with respect to our measures, in time that is polynomial in both n and k.

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, $1 \leq j \leq r$, 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 $b_j \in B_j$, $1 \leq j \leq r$.*

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

Generalization by suppression. Assume that $\overline{A}_j = A_j \cup \{A_j\}$ for all $1 \leq j \leq r$ and that g either leaves entries unchanged or replaces them by the entire set of attribute values, i.e., $g(b_1, \ldots, b_r) = (\bar{b}_1, \ldots, \bar{b}_r)$, where $\bar{b}_j \in \{b_j, *\}$, and * denotes an element outside $\bigcup_{1 \leq j \leq r} A_j$. In that case we refer to g as *generalization by suppression*.

Generalization by hierarchical clustering trees. Aggarwal et al. [1] 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 $\overline{A}_j$, $1 \leq j \leq r$, 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 (i) it includes all singleton subsets $\{a_i\}$, $1 \leq i \leq m$, (ii) it includes the entire set A, and (iii) it is a laminar collection in the sense that $B_1 \cap B_2 \in \{\emptyset, B_1, B_2\}$ for all $B_1, B_2 \in \overline{A}$.*

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.*

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

Definition 2.4. *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 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.*

We conclude this section with the following definitions:

Definition 2.5. *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 2.6. *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

The quality of a k-anonymization of a given database is typically measured by the amount of information that is lost due to generalization. Meyerson and Williams [22] 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. [1] considered generalizations by hierarchical clustering trees and proposed to penalize by r/ℓ_j each generalization of an entry $R_i(j)$ to a subset residing at the r-th level of the hierarchical clustering tree $\mathcal{T}(A_j)$, the height of which is ℓ_j. The tree measure is a generalization of the measure proposed by Meyerson and Williams.

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 [10] and [26], 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,*

$$\Pi_e(D, g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{r} H(X_j|\overline{R}_i(j)) \tag{3}$$

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

3.2.1 The non-monotonicity of the entropy measure

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

Definition 3.2. *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 $\Pi_{\rm e}$, on the other hand, is not always monotone, as we proceed to exemplify. 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 $h(p_1, \ldots, p_t) := -\sum_{i=1}^{t} p_i \log p_i$. 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 $\{3, 4\}$.

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. From privacy point of view, on the other hand, this is not always true since the generalization $4 \rightarrow \{3, 4\}$ in the above example reveals critical information and hence it should be penalized more than suppressing $4 \rightarrow *$. However, as explained in the introduction, we address the privacy concerns by respecting k-anonymity.

We believe that non-monotonicity is not a critical argument against the entropy measure for two reasons. The first reason is that such anomalies are rare. Namely, given a random variable X that takes values in a finite set A, and given two subsets $B_1 \subset B_2 \subseteq A$, usually $H(X|B_1) \leq H(X|B_2)$. To verify our claim, we conducted the following test: we sampled integers to represent the sizes of the two subsets, $n_1 = |B_1|$ and $n_2 = |B_2|$ (we always took $2n_1 \leq n_2 \leq 10n_1$). We then sampled uniformly at random the vector of probabilities for $X|B_2$. Finally, we computed $H(X|B_1)$ and $H(X|B_2)$. The desired inequality $H(X|B_1) \leq H(X|B_2)$ was violated only in a fraction of less than 10^{-5} of the total number of tests that we ran.

The second reason why the non-monotonicity of the entropy measure is not grave, is that it may always be rectified. More specifically, given any collection of subsets of a given attribute, $\overline{A}$, it is always possible to find a partial collection, $\hat{A} \subseteq \overline{A}$, so that the entropy measure is monotone on $\hat{A}$. 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 monotone entropy measure

Here we introduce the *monotone entropy measure*, a simple variant of the entropy measure that respects monotonicity.

Definition 3.3. *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,

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

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

Comparing (4) to (3), we see that each of the conditional entropies is multiplied by the corresponding probability. The monotone entropy measure coincides with the entropy measure when considering generalization by suppressions only. However, when the collections of subsets $\overline{A}_j$ include also intermediate subsets, the entropy that is associated with such a subset is multiplied by the probability of the subset. Since this multiplier increases as the subset includes more elements, the monotone entropy measure penalizes generalizations more than the entropy measure does.

Lemma 3.4. *The monotone entropy measure is monotone.*

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 $1 \le i \le n$, there exist indices $1 \le i_1 < i_2 < \cdots < i_{k-1} \le n$, 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 is an adaptation of [22, Theorem 3.1].

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

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 assume here that all collections of subsets are proper. The first algorithm, described in Sections 5.1-5.3, achieves an approximation ratio of $O(\ln k)$—a significant improvement with respect to the best known $O(k)$-approximation algorithm [1]. As that algorithm runs in time $O(n^{2k})$, we show in Section 5.4 that the $O(k)$-approximation algorithm of [1] that runs in time $O(kn^2)$ may be used also for approximating optimal k-anonymity when using the entropy and monotone entropy measures. The question of the existence of a fully polynomial approximation algorithm with a logarithmic approximation ratio remains open.

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]

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

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, $M \subseteq D$, its generalization cost by the entropy measure is,*

$$d(M) = d_{e}(M) = \sum_{j=1}^{r} H(X_j | \overline{M}_j) , \tag{5}$$

while its generalization cost by the monotone entropy measure is,

$$d(M) = d_{me}(M) = \sum_{j=1}^{r} \Pr(\overline{M}_j) \cdot H(X_j | \overline{M}_j) . \tag{6}$$

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. However, as we argued before, this problem rarely occurs, and we may always avoid it by narrowing down the collections $\overline{A}_j$, $1 \le j \le r$, until the entropy measure becomes monotone with respect to them. For the sake of simplicity, we assume monotonicity hereinafter. Namely,

$$M \subseteq M' \subseteq A_1 \times \cdots \times A_r \text{ implies that } d(M) \le d(M') . \tag{7}$$

If we use the generalization cost by the monotone entropy measure, $d(M) = d_{\mathrm{me}}(M)$, then (7) always holds.

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 [22]. The diameter of a set of records $M \subseteq A_1 \times \cdots \times A_r$ was defined as

$$\text{diam}(M) = \max_{R, R' \in M} \text{dist}(R, R'), \quad \text{where } \text{dist}(R, R') = |\{1 \le j \le r : R(j) \ne R'(j)\}| . \tag{8}$$

In other words, if the two records R and R' were to be generalized by means of suppression, 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.

Our notions of generalization cost, (5) and (6), and the notion of the diameter, (8), are functions that associate a *size* to a given set of records. Our notions, though, of generalization cost, improve that of the diameter as follows:

1. The generalization costs, (5) and (6), generalize the definition of the diameter, (8), in the sense that they apply to any type of generalization (the definition of the diameter is restricted to generalization by suppression).

2. The notions of the generalization cost use the more accurate entropy and monotone entropy measures (the definition of the diameter only counts the number of suppressed entries).

3. Most importantly, while the size of a set of records that is defined in (8) is a *diameter* (namely, it is based on pairwise distances), the size that is defined in (5) and (6) is a *volume*. All three 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 only by looking at pairs of records in M, the generalization costs do 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 very important, 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, $\overline{A}_j$, $1 \le j \le r$, are proper. Then the generalization costs $d(\cdot)$, (5) and (6), are sub-additive in the sense that for all $S, T \subseteq A_1 \times \cdots \times A_r$,*

$$S \cap T \ne \emptyset \text{ implies that } d(S \cup T) \le d(S) + d(T) . \tag{9}$$

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.

Hereinafter, Π denotes either the entropy measure of loss of information, $\Pi = \Pi_{\mathrm{e}}$, or the monotone entropy measure of loss of information, $\Pi = \Pi_{\mathrm{me}}$. The corresponding generalization cost is then denoted by $d(\cdot)$ (namely, $d(\cdot) = d_{\mathrm{e}}$ if $\Pi = \Pi_{\mathrm{e}}$ and $d(\cdot) = d_{\mathrm{me}}$ if $\Pi = \Pi_{\mathrm{me}}$).

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

$$d(\gamma) = \sum_{S \in \gamma} d(S)\,. \tag{10}$$

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. Let $d(\cdot)$ be the generalization cost by the measure Π. Then*

$$k \cdot d(\gamma^0) \leq \Pi(D, g(D)) \leq (2k-1) \cdot d(\gamma^0)\,. \tag{11}$$

Proof. As we have

$$k \leq |S| \leq 2k-1, \text{ for all } S \in \gamma^0, \tag{12}$$

and

$$\Pi(D, g(D)) = \sum_{S \in \gamma^0} |S| \cdot d(S), \tag{13}$$

inequality (11) follows from (13), (12) and (10). □

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*

$$\Pi(D, g(D)) \leq \frac{2d(\gamma^0)}{d(\hat{\gamma})} \cdot OPT(D), \tag{14}$$

where

$$OPT(D) := \min_{g \in \mathcal{G}} \Pi(D, g(D)). \tag{15}$$

Proof. Let g^* be a k-anonymization for which $OPT(D) = \Pi(D, g^*(D))$ and let γ^* be its corresponding clustering. On one hand, by the lower bound in (11) and the definition of $\hat{\gamma}$,

$$OPT(D) = \Pi(D, g^*(D)) \geq k \cdot d(\gamma^*) \geq k \cdot d(\hat{\gamma}). \tag{16}$$

On the other hand, by the upper bound in (11),

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

Hence, by (17) and (16),

$$\Pi(D, g(D)) \leq \frac{2k-1}{k} \cdot \frac{d(\gamma^0)}{d(\hat{\gamma})} \cdot OPT(D) \leq \frac{2d(\gamma^0)}{d(\hat{\gamma})} \cdot OPT(D)\,.$$

□

5.3 Approximating optimal k-anonymization

Our approximation algorithm follows the algorithm of [22]. 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 [18].

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.

Since the greedy algorithm for the WEIGHTED SET COVER problem has logarithmic approximation guarantee (see, e.g., [8]), the result of that phase is a cover $\gamma \in \Gamma$ for which $d(\gamma) \leq (1 + \ln 2k) d(\hat{\gamma})$.

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*

$$\Pi(D, g(D)) \leq 2(1 + \ln 2k) \cdot OPT(D)\,, \tag{18}$$

where $OPT(D)$ is the cost of an optimal k-anonymization, (15).

Proof. First, we observe that $d(\gamma^0) \leq d(\gamma)$, as implied by our monotonicity assumption, (7), and by Lemma 5.3. Hence, by the fact that $d(\gamma) \leq (1 + \ln 2k) d(\hat{\gamma})$, we get $d(\gamma^0) \leq (1 + \ln 2k) d(\hat{\gamma})$. Finally, by Theorem 5.5, the k-anonymization g satisfies (18). □

The corresponding result in [22] 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 [1, Theorem 5]. The approximation algorithms in both [22] and [1] 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 [1] 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.

It should be noted that our improved approximation algorithm works also with the tree measure, if we modify the definition of the generalization cost, Definition 5.2, to be consistent with that measure. Such a modified generalization cost is clearly monotone, (7), and sub-additive, (9), whence all of our claims hold also for that cost. The algorithm described in this section runs in time $O(n^{2k})$. The exponential dependence of the running time on k is due to the fact that we examine all subsets of records of D with cardinalities between k and $2k - 1$.

5.4 A fully polynomial approximating algorithm

Here we describe briefly (due to space limitations) the algorithm of Aggarwal et al. [1], and concentrate on the necessary modifications that are required in order to make it work for our entropy measure.

The algorithm starts by considering the graph representation $G = (V, E)$ of the database D. This is a complete weighted graph, where $V = D = \{R_1, \ldots, R_n\}$, and the edge $e_{i,j} = \{R_i, R_j\} \in E$ has weight $w(e_{i,j}) = d(\{R_i, R_j\})$, where $d(\cdot)$ is the generalization cost by the entropy measure, (5). Let $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_s\}$ be a spanning forest of G. If all trees in that forest are of size at least k then that forest induces a k-anonymization of D, denoted $g_{\mathcal{F}}$ (namely, all records in the tree $\mathcal{T}_\ell$ are replaced by the closure of that tree, $\overline{\mathcal{T}_\ell}$). 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 measure Π (that could be either the entropy measure, Π_{e}, or the monotone entropy measure, Π_{me}). The generalization cost of $g_{\mathcal{F}}$ is then

$$\Pi(D, g_{\mathcal{F}}(D)) = \sum_{i=1}^{n} c(R_i, g_{\mathcal{F}}) \,. \tag{19}$$

An important observation in designing the algorithm is the following.

Lemma 5.7. *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:*

$$c(R_i, g_{\mathcal{F}}) \le w(\mathcal{T}_{j(i)}) := \sum_{e \in \mathcal{T}_{j(i)}} w(e) \,. \tag{20}$$

Proof. We need to prove that for any given tree, $\mathcal{T}$, we have $d(\mathcal{T}) \le w(\mathcal{T})$, 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 $|\mathcal{T}| \le 2$ 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 and monotone entropy measures, Lemma 5.3, $d(\mathcal{T}) = d(\mathcal{T}_1 \cup \mathcal{T}_2) \le d(\mathcal{T}_1) + d(\mathcal{T}_2)$. As the induction hypothesis applies to both $\mathcal{T}_1$ and $\mathcal{T}_2$ we infer that $d(\mathcal{T}) \le w(\mathcal{T}_1) + w(\mathcal{T}_2) = w(\mathcal{T})$, thus proving the claim. □

Theorem 5.8. *Let $OPT = OPT(D)$ be the cost of an optimal k-anonymization of D with respect to the measure of loss of information, Π, and let L be an integer such that $L \ge k$. 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.,*

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

Proof. Invoking (19), (20), and the fact that each node belongs to exactly one tree in the forest, we conclude that

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

Hence, since all trees are of size L at the most, we get that $\Pi(D, g_{\mathcal{F}}(D)) \leq L \cdot \sum_{j=1}^{s} w(\mathcal{T}_j) \leq L \cdot OPT$. □

The algorithm then proceeds in two stages:

STAGE 1: Create a spanning forest $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_s\}$ whose total weight is at most OPT (the cost of an optimal k-anonymization) and in which all trees are of size at least k.

STAGE 2: Compute a decomposition of this forest such that each component has size in the range $[k, L]$ for $L = \max\{2k-1, 3k-5\}$.

Both stages are described in detail in [1]. In view of Theorem 5.8, this algorithm achieves an approximation ratio of $O(k)$. Its analysis, to a large extent, is independent of the underlying measure of loss of information that determines the weight of the edges. Furthermore, it is a fully polynomial algorithm whose running time is $O(kn^2)$.

Acknowledgements. The authors thank Jacob Goldberger who proposed the *monotone entropy measure* as a modification of the entropy measure that respects monotonicity.

References

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. In *ICDT*, 2005.

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

[3] D. Agrawal and C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *PODS*, 2001.

[4] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD*, 2000.

[5] R. Agrawal, R. Srikant, and D. Thomas. Privacy preserving OLAP. In *SIGMOD*, 2005.

[6] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *PODS*, 2005.

[7] 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.

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

[9] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[10] 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.

[11] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, 2003.

[12] C. Dwork. Differential privacy. In *ICALP*, 2006.

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

[14] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 2003.

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

[16] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC*, 1987.

[17] S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, 28:270–299, 1984.

[18] D. S. Johnson. Approximation algorithms for combinatorial problems. *JCSS*, 9:256–278, 1974.

[19] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. In *PODS*, 2005.

[20] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. *JCSS*, 6:244–253, 2003.

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

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

[23] P. Samarati. Protecting respondent's privacy in microdata release. *TKDE*, 13:1010–1027, 2001.

[24] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, 1998.

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

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

[27] A. Yao. How to generate and exchange secrets. In *FOCS*, 1986.