k-Anonymization with Minimal Loss of Information

Aristides Gionis * Tamir Tassa †

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 three 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 [MW04] and Aggarwal et al. [AFK$^+$05]. 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 two of our measures as well as for the previously studied measures. This improves the best-known $O(k)$-approximation of [AFK$^+$05]. 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 [AFK$^+$05] 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 hospitalized patients in the oncological department in some hospital. Each record in that database describes a patient by several "general" attributes such as age, gender, address, profession, marital status etc., as well as "specific" attributes such as the type of cancer that was diagnosed in that patient or the patient's response to some new medication.

Such databases are of interest to the general public, even though they hold information on individuals, since they can be used for medical research 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. The problem lies with those attributes (columns) to which we referred above as "general". Assume that Alice knows that her neighbor Bob was hospitalized in the oncological department that provided the data for the research. She may use her knowledge of Bob's "general" attributes (which she may know either since she knows Bob in person or because she traces that information in publicly available databases such as the voter's list) in order to uniquely trace Bob's record in the database and then infer his "specific" medical information. In other words, the table's attributes that can be found in other publicly available databases — known as the *public attributes* or *quasi-identifiers* — may be used by an adversary in order to learn the value of the specific attributes — known as the *private attributes* — for some of the individuals in the database. Hence, it is desired to reveal information in order

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

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

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 [Dal77] 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 [GM84]. Alas, while semantic security for cryptosystems may be achieved, Dalenius' idealized goal may not be achieved [Dwo06]. 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 [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].

1.1 k-Anonymization

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 records becomes indistinguishable from at least $k-1$ additional records, when projected on the subset of public attributes. 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 three public attributes — name, age, and address — and one 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. 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 for any given $k \leq n$ (where n is the number of records in the database).

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 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. [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 $O(k)$.

1.2 Our contribution

In this study we extend the framework of k-anonymization to include any type of generalization operators and define three measures of loss of information that are 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 measures, *the entropy measure*, *the monotone entropy measure*, and *the non-uniform entropy measure*. We discuss those measures and show that they serve the purposes of data mining better than the previous measures. We show that the problem of k-anonymization with minimal loss of data (measured by any one 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 the entropy and monotone entropy measures, as well as the measures that were used in [MW04] and [AFK$^+$05]. 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.

1.3 Organization of the paper

We begin with preliminaries and introducing our notation in Section 2. In Section 3 we give a precise definition of what is generalization, and we describe and illustrate several natural types of generalization. In Section 4 we describe the measures of loss of information that were used in [AFK$^+$05, MW04] and their shortcomings (Section 4.1); we then propose three measures of loss of information that are more general and more accurate than the previously used ones: the entropy measure (Section 4.2), the monotone entropy measure (Section 4.3) and the non-uniform entropy measure (Section 4.4). In Section 5 we define the problem of k-anonymization with minimal loss of information and we prove that it is NP-hard with respect to all three measures of loss of information. In Section 6 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 an adaptation of the approximation algorithm of [AFK$^+$05] that achieves an $O(k)$-approximation ratio with respect to the entropy and monotone entropy measures, in time that is polynomial in both n and k. Finally, Section 7 includes a summary of our study and discussion of some open problems.

2 Preliminaries

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

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

The corresponding private database holds the private information,

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

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 $R_i(j)$.

It should be noted that the sets in (1) and (2) may be multi-sets, in the sense that they may include repeated records. For example, if $r = 3$ and the three attributes are gender, age, and zip code, there may be two distinct individuals that are described by the public record $(M, 42, 91845)$ (namely, two 42-year-old males who reside in the area code 91845). The same observation holds for all sets of records that appear hereinafter — they may include repetitions.

3 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 $R_i(j)$, $1 \le i \le n$, $1 \le j \le r$, which is an element of A_j, is replaced by a subset of A_j that includes that element.

Definition 3.1 *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 3.2 *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 $\{a_i\}$, $1 \le i \le m$, (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 3.3 *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.*

The proof of Lemma 3.3 is given in the appendix. Note that the framework of proper collections of subsets extends the hierarchical clustering tree framework, as it allows unbalanced trees.

Example 3.4
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 $\{10(i-1)+1, \ldots, 10i\}$, $1 \le i \le 12$, 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 $\{i : s \le i \le t\}$. 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 3.5 *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 $\overline{R}_i := g_i(R_i)$, $1 \le i \le n$, we refer to the database $g(D) := \{\overline{R}_1, \ldots, \overline{R}_n\}$ as a generalization of D.*

Hereinafter, we let D_i and $g(D)_i$ denote the ith records in D and $g(D)$ respectively; namely, $D_i = R_i$ and $g(D)_i = \overline{R}_i$.

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 $1 \le i < j \le n$, the above definition allows generalizations where $\overline{R}_i \ne \overline{R}_j$.

We conclude this section with the following definitions:

Definition 3.6 *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 $1 \le j \le r$.*

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 3.7 *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 $1 \le i \le n$.*

4 Measures of loss of information

4.1 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 call *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.

4.2 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, $1 \le j \le r$, denote hereinafter the value of the attribute A_j in a randomly selected record from D. Then

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

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

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

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 4.1 *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, $1 \le j \le r$, 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 $g(D)$.

4.2.1 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.

To illustrate the significance of these two features of the entropy measure to data mining, we examine the following two examples.

Example 4.2

Consider the following public database D that consists of four individuals and two attributes — gender and zip code:

$$D = \begin{bmatrix} M & 41278 \\ M & 98705 \\ F & 41278 \\ F & 98705 \end{bmatrix}.$$

The following two generalized tables are 2-anonymizations of D:

$$g_1(D) = \begin{bmatrix} * & 41278 \\ * & 98705 \\ * & 41278 \\ * & 98705 \end{bmatrix}, \quad g_2(D) = \begin{bmatrix} M & * \\ M & * \\ F & * \\ F & * \end{bmatrix}.$$

The tree measure will consider the two generalizations as equally distant from D in terms of information loss. The entropy measure, on the other hand, will favor $g_1(D)$ since the entropy of the gender is smaller than that of the zip code. The generalized table $g_1(D)$ is more likely to serve better the purposes of data mining because it conceals the less informative attribute and leaves out the attribute that may be of better use for extracting interesting patterns and association rules. □

We would like to stress that while the entropy measure indeed offers a more accurate measurement of the public information that is lost due to anonymization, using it does not guarantee that the resulting table has better utility for data mining. For instance, if in the above example the private information is a disease that has correlation with the gender but it has no correlation with geographical location, we should prefer always the generalization of entries from the zipcode column over that of entries from the gender column. Almost all studies thus far, including the present one, concentrate on minimizing the public information that is lost due to anonymization, rather than minimizing the amount of *relevant* public information that is lost. Namely, a better anonymization stratgey would be one that already performs some data mining in order to reveal the dependencies between the public and private attributes and, consequently, make better decisions regarding which public attributes are less correlated with the private attribute. Such a strategy may yield anonymized tables that retain the maximal amount of relevant public information. A recent study [GT08], that was triggered and motivated by the present one, offers a measure of information that extends the ones presented herein in order to serve the above outlined goal.

Example 4.3

Consider a database in which one of the attributes is profession, and among the values that it includes there are: *kindergarten teacher*, *school teacher*, *university lecturer*; and *electrical engineer*, *mechanical engineer*, and *civil engineer*. Assume that in the first level of generalization, the first three professions are grouped together under *education* while the other three are grouped together under *engineering*. Finally, assume that the table has similar percentage of kindergarten teachers, school teachers and university lecturers, but, among the engineers, most of them are electrical engineers, while there are only few mechanical and civil engineers.

The tree measure will penalize a generalized entry with the value *education* in the same way it penalizes a generalized entry with the value *engineering*. However, the entropy measure will penalize the first generalization more than it does the second one, because the conditional entropy of *education* is roughly $\log 3$ while that of *engineering* is close to zero. This distinction serves the purposes of data mining, since any association rule that involves the generalized value *engineering* may be replaced with an association rule with the more precise value *electrical engineering* (barring a slight degradation of accuracy), while no similar refinement can be made with association rules that involve the generalized value *education*. Hence, the entropy measure gives an advantage to generalizations that allow more efficient data mining. □

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 4.4
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 4.4 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 4.5
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.

4.2.2 The non-monotonicity of the entropy measure

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

Definition 4.6 *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 $\Pi(D, g(D)) \le \Pi(D, g'(D))$ 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 show in the following example.

Example 4.7
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 the 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 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 3.3, 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.

4.3 The monotone entropy measure

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

Definition 4.8 *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, $1 \le j \le r$, in a randomly selected record from D. Then if $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ is a generalization of D,*

$$\Pi_{me}(D, g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{r} \Pr(\overline{R}_i(j)) \cdot H(X_j | \overline{R}_i(j)) \tag{5}$$

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

Comparing (5) to (4), 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.

Example 4.9
Consider an attribute X that takes values in $A = \{a_1, \ldots, a_m\}$ with a uniform distribution, i.e., $\Pr(X = a_j) = 1/m$ for all $1 \le j \le m$. Assume that we replace an exact value of that attribute with a subset of values, $B \subseteq A$, of size $b = |B|$. Then the entropy measure will penalize such a generalization by $\log b$, while the monotone entropy measure will penalize it by $\frac{b}{m} \cdot \log b$.

Lemma 4.10 *The monotone entropy measure is monotone.*

Proof. Let X be a random variable that takes values in $A = \{a_1, \ldots, a_m\}$ and let $B = \{a_1, \ldots, a_b\}$ be a nonempty subset of A. (For convenience and without loss of generality, we assume that B consists of the first b elements of A.) Assume that $\Pr(X = a_j) = p_j$, $1 \le j \le m$, and let $p_A = \Pr(A) = \sum_{j=1}^{m} p_j$ and $p_B = \Pr(B) = \sum_{j=1}^{b} p_j$. Then

$$\Pr(B) \cdot H(X|B) = p_B \cdot \sum_{j=1}^{b} \frac{p_j}{p_B} \log \frac{p_B}{p_j} = p_B \log p_B + \sum_{j=1}^{b} p_j \log \frac{1}{p_j}.$$

Similarly,

$$\Pr(A) \cdot H(X|A) = p_A \log p_A + \sum_{j=1}^{m} p_j \log \frac{1}{p_j}.$$

Since $p_B \le p_A$, the last two equalities imply that $\Pr(B) \cdot H(X|B) \le \Pr(A) \cdot H(X|A)$. This proves that Π_{me} is monotone. □

4.4 The non-uniform entropy measure

Both the entropy and the monotone entropy measures are uniform for all records in the same cluster. Consider, for instance, the setting in Example 4.7. 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*.

Generalization	Entropy ($\Pi_{\rm e}$)	Monotone entropy ($\Pi_{\rm me}$)	Non-uniform entropy ($\Pi_{\rm ne}$)
$1 \mapsto \{1,2\}$	$h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right)$	$(1-2\varepsilon)h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right)$	$-\log\frac{1-3\varepsilon}{1-2\varepsilon}$
$2 \mapsto \{1,2\}$	$h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right)$	$(1-2\varepsilon)h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right)$	$-\log\frac{\varepsilon}{1-2\varepsilon}$
$3 \mapsto \{3,4\}$	$h\left(\frac{1}{2}, \frac{1}{2}\right) = 1$	$2\varepsilon \cdot h\left(\frac{1}{2}, \frac{1}{2}\right) = 2\varepsilon$	$-\log\frac{1}{2} = 1$
$4 \mapsto \{3,4\}$	$h\left(\frac{1}{2}, \frac{1}{2}\right) = 1$	$2\varepsilon \cdot h\left(\frac{1}{2}, \frac{1}{2}\right) = 2\varepsilon$	$-\log\frac{1}{2} = 1$

Table 1: Partial generalization.

Generalization	Entropy ($\Pi_{\rm e}$)	Monotone entropy ($\Pi_{\rm me}$)	Non-uniform entropy ($\Pi_{\rm ne}$)
$1 \mapsto *$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$-\log(1-3\varepsilon)$
$2 \mapsto *$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$\log\varepsilon^{-1}$
$3 \mapsto *$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$\log\varepsilon^{-1}$
$4 \mapsto *$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$h(1-3\varepsilon, \varepsilon, \varepsilon, \varepsilon)$	$\log\varepsilon^{-1}$

Table 2: Generalization by suppression.

Definition 4.11 *Let $D = \{R_1, \ldots, R_n\}$ be a database having public attributes A_j, $1 \le j \le r$, and let $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ be a generalization of D. Then*

$$\Pi_{ne}(D, g(D)) = \sum_{i=1}^{n}\sum_{j=1}^{r} -\log \Pr(R_i(j)|\overline{R}_i(j)) \tag{6}$$

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

Lemma 4.12 *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 $\Pr(X = a|X \in A_1) \ge \Pr(X = a|X \in A_2)$. The monotonicity of $\Pi_{\rm ne}$ immediately follows. □

Let us exemplify the non-uniform entropy measure and compare it to the previous measures on Example 4.7. In Table 1 we present the three 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 columns for $\Pi_{\rm me}$ and $\Pi_{\rm ne}$ 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 monotone and non-uniform entropy measures. 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 (6). 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

$$\Pi_{\rm e}(D, g(D)) = n \cdot \left[(1-2\varepsilon)h\left(\frac{1-3\varepsilon}{1-2\varepsilon}, \frac{\varepsilon}{1-2\varepsilon}\right) + 2\varepsilon\right], \tag{7}$$

while

$$\Pi_{\rm ne}(D, g(D)) = n \cdot \left[(1-3\varepsilon)\log\frac{1-2\varepsilon}{1-3\varepsilon} + \varepsilon\log\frac{1-2\varepsilon}{\varepsilon} + 2\varepsilon \right]. \tag{8}$$

As can be easily seen, the two values in (7) and (8) coincide. The same coincidence occurs also in the case of generalization by suppression. This is no coincidence.

Lemma 4.13 *Let D be a database and let $g(D)$ be a generalization of D where for all $1 \le i < i' \le n$ and for all $1 \le j \le r$, either $g(D)_i(j) = g(D)_{i'}(j)$ or $g(D)_i(j) \cap g(D)_{i'}(j) = \emptyset$. Then $\Pi_e(D, g(D)) = \Pi_{ne}(D, g(D))$.*

The proof of Lemma 4.13 is given in the appendix. We note that the condition in Lemma 4.13 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 $\Pi_{\rm e}$ and $\Pi_{\rm ne}$ might differ.

5 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 5.1 *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 and its proof (that is postponed to the appendix) 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 three proposed entropy measures.

Theorem 5.2 *The problem of k-ANONYMIZATION with generalization by suppression, where the measure of loss of information is the entropy measure (4), $\Pi = \Pi_e$, the monotone entropy measure (5), $\Pi = \Pi_{me}$, or the non-uniform entropy measure (6), $\Pi = \Pi_{ne}$, is NP-hard for $k \ge 3$, if $|A_j| \ge \lceil (n-k)/(k-1) \rceil + 1$ for all attributes $A_1, \ldots, A_r$.*

6 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 and monotone entropy measures. We also assume here that all collections of subsets are proper. The first algorithm, described in Sections 6.1-6.3, 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 6.1 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 6.2 we explore the relations between k-anonymizations, clusterings and covers of a given database D. Using these relations, we describe in Section 6.3 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 6.3 runs in time $O(n^{2k})$. In Section 6.4 we discuss another approximation algorithm that is fully 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 and monotone entropy measures. The question of the existence of a fully polynomial approximation algorithm with a logarithmic approximation ratio remains open. It also remains open to find an approximation algorithm, fully polynomial or not, for the non-uniform entropy measure.

6.1 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 6.1 *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 6.2 *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,*

$$d(M) = d_e(M) = \sum_{j=1}^{r} H(X_j | \overline{M}_j) \,, \tag{9}$$

while its generalization cost by the monotone entropy measure is,

$$d(M) = d_{me}(M) = \sum_{j=1}^{r} \Pr(\overline{M}_j) \cdot H(X_j | \overline{M}_j) \,. \tag{10}$$

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 4.6. 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, this problem rarely occurs, and we may always avoid it 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,

$$M \subseteq M' \subseteq A_1 \times \cdots \times A_r \text{ implies that } d(M) \leq d(M') \,. \tag{11}$$

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

If we use the generalization cost by the monotone entropy measure, $d(M) = d_{\mathrm{me}}(M)$, then (11) 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 [MW04]. The diameter of a set of records $M \subseteq A_1 \times \cdots \times A_r$ was defined as

$$\mathrm{diam}(M) = \max_{R,R' \in M} \mathrm{dist}(R, R'), \quad \text{where} \;\; \mathrm{dist}(R, R') = |\{1 \le j \le r : R(j) \ne R'(j)\}| \,. \tag{12}$$

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.

Our notions of generalization cost, (9) and (10), and the notion of the diameter, (12), 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, (9) and (10), generalize the definition of the diameter, (12), 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 (12) is a *diameter* (namely, it is based on pairwise distances), the size that is defined in (9) and (10) 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 6.3 *Assume that all collections of subsets, $\overline{A}_j$, $1 \le j \le r$, are proper. Then the generalization costs $d(\cdot)$, (9) and (10), are sub-additive in the sense that for all $S, T \subseteq A_1 \times \cdots \times A_r$,*

$$S \cap T \ne \emptyset \textit{ implies that } d(S \cup T) \le d(S) + d(T) \,. \tag{13}$$

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, $1 \le j \le r$, 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 $S \cap T \ne \emptyset$, we conclude that $S_j \cap T_j \ne \emptyset$. Hence $\overline{S}_j \cap \overline{T}_j \ne \emptyset$. 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 $\overline{U}_j = \overline{T}_j$. We conclude that

$$H(X_j|\overline{U}_j) \le H(X_j|\overline{S}_j) + H(X_j|\overline{T}_j) \,, \tag{14}$$

and

$$\Pr(\overline{U}_j) \cdot H(X_j|\overline{U}_j) \le \Pr(\overline{S}_j) \cdot H(X_j|\overline{S}_j) + \Pr(\overline{T}_j) \cdot H(X_j|\overline{T}_j). \tag{15}$$

Summing (14) for all $1 \le j \le r$ we arrive at (13) for $d(\cdot) = d_{\mathrm{e}}$. Summing (15) for all $1 \le j \le r$ we arrive at (13) for $d(\cdot) = d_{\mathrm{me}}$. $\square$

Lemma 6.3 does not necessarily hold for generalizations that are not proper. As a simple example, consider the case of one attribute ($r = 1$), where $A_1 = A = \{1,2,3\}$, and $\overline{A} = \mathcal{P}(A)$ (note that a generalization that allows any subset of attribute values is indeed non-proper). Let $S = \{1,2\}$, $T = \{2,3\}$, and assume that $\Pr(X = 1) = \frac{1}{2} - \varepsilon$, $\Pr(X = 2) = 2\varepsilon$, and $\Pr(X = 3) = \frac{1}{2} - \varepsilon$. Then, letting $d(\cdot) = d_{\mathrm{e}}$, (9),

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

while

$$d(S) = H(X|S) \approx 0, \quad d(T) = H(X|T) \approx 0\,.$$

Hence, in this case $d(S \cup T) > d(S) + d(T)$. It may be easily seen that also the generalization cost by the monotone entropy measure, $d(\cdot) = d_{\mathrm{me}}$, (10), fails to satisfy sub-additivity in this case.

6.2 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:

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

This cost is closely related to the measure of loss of information by k-anonymization, as stated in the next lemma.

Lemma 6.4 *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*

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

Proof. As we have

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

and

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

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

Next, we claim the following:

Theorem 6.5 *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*

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

where

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

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 (17) and the definition of $\hat{\gamma}$,

$$OPT(D) = \Pi(D, g^*(D)) \geq k \cdot d(\gamma^*) \geq k \cdot d(\hat{\gamma}). \tag{22}$$

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

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

Hence, by (23) and (22),

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

□

6.3 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.

Since the greedy algorithm for the WEIGHTED SET COVER problem has logarithmic approximation guarantee (see, e.g., [Chv79]), the result of that phase is a cover $\gamma \in \Gamma$ for which

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

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

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

where $OPT(D)$ 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, (11), and by Lemma 6.3. Hence, by (24), $d(\gamma^0) \leq (1 + \ln 2k)d(\hat{\gamma})$. Finally, by Theorem 6.5, 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 6.1, 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 6.2, to be consistent with that measure. Such a modified generalization cost is clearly monotone, (11), and sub-additive, (13), 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 $2k-1$.

6.4 A fully polynomial approximating algorithm

6.4.1 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 6.7 *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, Definition 6.2.*

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 measure Π (that could be either the entropy measure, $\Pi_{\rm e}$, or the monotone entropy measure, $\Pi_{\rm me}$). The generalization cost of $g_{\mathcal{F}}$ is then

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

An important observation in designing the algorithm is the following.

Lemma 6.8 *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 6.3,

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

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

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

thus proving the claim. □

We are now able to state the main result.

Theorem 6.9 *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.,*

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

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

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

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

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

□

6.4.2 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 6.10 *The forest produced by the algorithm* Forest *has minimum tree size k and has weight at most OPT.*

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 monotonicity of Π. 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 fully polynomial.

7 Conclusions

In this paper we studied the problem of k-anonymization, and we proposed three 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 measures.

We then continued to study the approximability of that problem, with respect to the entropy and monotone entropy measures. 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 fully polynomial approximation algorithm for our entropy and monotone entropy measures with an $O(k)$-approximation guarantee.

Two main open problems remain. The first is to find a fully polynomial approximation algorithm with performance guarantee better than $O(k)$. The second open problem is to design an approximation algorithm, fully 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.

8 Appendix

8.1 Proof of Lemma 3.3

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. $\square$

8.2 Proof of Lemma 4.13

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 $a_\ell \in A$, $1 \le \ell \le m$, that appears in D is generalized to a unique subset $B_{h(\ell)}$ where $1 \le h(\ell) \le t$.

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 $1 \le \ell \le m$, $1 \le j \le t$. This implies that

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

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 $\Pi_{\mathrm{ne}}(D,g(D))$. □

8.3 Proof of Theorem 5.2

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 $H=(U,E)$ be an input to the k-dimensional perfect matching problem, and denote $U = \{u_1,\ldots,u_n\}, E=\{e_1,\ldots,e_m\}$. We construct the input for the k-anonymization problem as follows. For every node $u_i \in U$ we define a record R_i, and for every edge $e_j \in E$ we define an attribute A_j. The possible values for all attributes A_j, $1 \le j \le m$, are drawn from the set

$$A = \{0,1,\ldots,\lceil (n-k)/(k-1) \rceil\}.$$

The entries $R_i(j),\ 1 \le i \le n,\ 1 \le j \le m$ in the database D are defined according to the following rule: If $u_i \in e_j$ then $R_i(j)=0$; the rest of the $n-k$ entries of the j-th attribute in D attain values from the set $A \setminus \{0\}$, such that each value appears exactly $k-1$ times, except for the last value $\lceil (n-k)/(k-1) \rceil$ that may appear less than $k-1$ times.

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

$$\Pi_{\mathrm{e}}(D,g(D)) \le n(m-1) \cdot H_{k,n} \tag{28}$$

where $H_{k,n}$ is the entropy of each of the attributes. Note that by our definition of D, the information loss due to each suppressed entry is the same, namely $H_{k,n}$.

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

$$\overline{R}_i(j) = \begin{cases} 0 & \text{if } u_i \in e_j \text{ and } e_j \in S \\ * & \text{otherwise} \end{cases}. \tag{29}$$

We claim that this generalization is a k-anonymization. Indeed, consider the k nodes in an edge $e \in S$. Clearly, the above generalization dictates that their corresponding generalized records are equal. Since every node belongs to exactly one edge $e \in S$, we are looking at a k-anonymized table. As the number of suppressed entries is clearly $n(m-1)$, this k-anonymization satisfies (28).

Conversely, assume that there exists a k-anonymization $g(D)$ that satisfies (28). Such an anonymization must have at most $n(m-1)$ suppressed entries. We first note that in any k-anonymization of D, the only value that may be left non-suppressed is 0, because every other value appears in no more than $k-1$ entries in each attribute. Now, every row in $g(D)$ can have at most one zero entry, otherwise $g(D)$ would contain a group of k rows which have zeros in two or more attributes, leading to a contradiction because the hypergraph is simple. On the other hand, every row in $g(D)$ must contain at least one non-suppressed value, otherwise there would be more than $n(m-1)$ suppressed entries in the table. We conclude that every row contains exactly one zero, while all other entries are suppressed. Therefore, the clusters induced by the k-anonymization $g(D)$

may be translated to a k-dimensional perfect matching, by choosing only the attributes (namely, edges) which are represented by k zeros in $g(D)$.

Since the entropy measure coincides with the monotone entropy measure in case of generalization by suppression, the above proof of NP-hardness of the problem of k-anonymization with minimal loss of information with respect to the entropy measure applies also to the monotone entropy measure.

Finally, we turn to proving the NP-hardness with respect to the non-uniform entropy measure. To that end, we note that in each column in D there are exactly k entries with the value 0, $k-1$ entries with the value j, for all $1 \leq j < \lceil (n-k)/(k-1) \rceil$, and $c_{n,k}$ entries with the value $\lceil (n-k)/(k-1) \rceil$, where $c_{n,k} = k-1$ if $(k-1)|(n-k)$ and $c_{n,k} = (n-k) \mod (k-1)$ otherwise. Hence, by suppressing an entry with the value 0, the non-uniform entropy measure incurs a penalty of $\log(n/k)$; suppressing an entry with the value j, for all $1 \leq j < \lceil (n-k)/(k-1) \rceil$, incurs a penalty of $\log(n/(k-1))$; and suppressing an entry with the value $\lceil (n-k)/(k-1) \rceil$ incurs a penalty of $\log(n/c_{n,k})$. We infer that the sum of suppression penalties over all entries in the table D is

$$P_{n,k} := m \cdot [k \cdot \log(n/k) + (\lceil (n-k)/(k-1) \rceil - 1) \cdot \log(n/(k-1)) + c_{n,k} \cdot \log(n/c_{n,k})] \, .$$

With these notations, we claim that there exists a k-dimensional perfect matching if and only if there exists a k-anonymization $g(D)$ for which

$$\Pi_{\mathrm{ne}}(D, g(D)) \leq P_{n,k} - n \cdot \log(n/k) \, . \tag{30}$$

Assume first that there is a k-dimensional perfect matching, $S \subset E$, and define the same k-anonymization $g(D)$ as in (29). Since all entries in $g(D)$ are suppressed, except for one entry of the value zero in each of the n records, we infer that $g(D)$ satisfies (30). Conversely, assume that there exists a k-anonymization $g(D)$ that satisfies (30). 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 (30). As shown earlier, such a k-anonymization defines a k-dimensional perfect matching of H. $\square$

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

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.

[Dal77] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[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.

[Dwo06] C. Dwork. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP), Part II*, pages 1–12, 2006.

[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.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[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.

[GT08] J. Goldberger and T. Tassa. Learning-Enhancing Methods for Anonymization of Tables. 2008. Submitted for publication.

[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.