# Efficient Anonymizations with Enhanced Utility

Jacob Goldberger
School of Engineering
Bar-Ilan University, Ramat-Gan, Israel
goldbej@eng.tau.ac.il

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

***Abstract*—The $k$-anonymization method is a commonly used privacy-preserving technique. Previous studies used various measures of utility that aim at enhancing the correlation between the original public data and the generalized public data. We, bearing in mind that a primary goal in releasing the anonymized database for data mining is to deduce methods of predicting the private data from the public data, propose a new information-theoretic measure that aims at enhancing the correlation between the generalized public data and the private data. Such a measure significantly enhances the utility of the released anonymized database for data mining. We then proceed to describe a new and highly efficient algorithm that is designed to achieve $k$-anonymity with high utility. That algorithm is based on a modified version of sequential clustering which is the method of choice in clustering, and it is independent of the underlying measure of utility.**

***Keywords*-data privacy, mutual information, k-anonymity**

## I. Introduction

Our society experiences in recent years unprecedented growth in the amount of data that is collected on individuals, organizations, companies and other entities. Of particular interest are data containing structured information on individuals. Data holders are then faced with the intricate task of releasing information that does not compromise privacy. The goal is to allow releasing of data in order to detect interesting trends or correlations, while still protecting the privacy of individuals. Privacy-preserving data mining [2] has been proposed as a paradigm of exercising data mining while protecting the privacy of individuals. Many approaches were suggested, implemented and theoretically studied for playing this delicate game that requires finding the right path between data hiding and data disclosure. One of these approaches, proposed by Samarati and Sweeney [14], is $k$-anonymization. The technique of $k$-anonymization suggests to modify the values of the public attributes of the data by means of generalization so that if the database is projected on the subset of the public attributes, each record of the table becomes indistinguishable from at least $k-1$ other 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. The model of $k$-anonymity has been shown to be insufficient to protect against all types of linking attack, e.g. [6], [10], [16]. However, $k$-anonymity remains an important model since it is used in practice and its study is relevant to other aggregation-based methods as well.

The main challenge is to achieve $k$-anonymity with minimal loss of information or, alternatively speaking, with maximal utility. The definition of the target function, namely, the measure of utility, is critical in this discussion. Several measures of utility were suggested in the literature. The problem of finding the $k$-anonymization with maximal utility (or minimal information loss) was shown to be NP-hard [1], [8], [11]. Hence, the possible approaches are either heuristical algorithms [4], [7], [12] or approximation algorithms with a guaranteed approximation factor [1], [8], [11]. Usually, the former type of algorithms outperforms the latter type.

Our contribution in this study is twofold: First, we propose a new information-theoretic measure of utility that takes into account the private attributes and aims at enhancing the correlation between the generalized public data and the private data. This is in contrast to most all measures of utility that were used in previous studies that rely only on the public data and quantify the correlation between the original public data and the generalized public data. A primary goal of data mining is to find frequent patterns or rules to predict the private data from the public data. Hence, we deem our measure as one that best serves the purpose of obtaining anonymized tables with maximal utility for such applications of data mining. Then, we proceed to describe a sequential clustering algorithm that obtains high-utility anonymizations. Our algorithm, which is independent of the underlying utility measure, is based on a modified version of sequential clustering that is the method of choice in clustering. It offers similar results (in terms of utility) as the current heuristical algorithm of choice, but it is significantly faster.

The paper is organized as follows. In Section II we provide the basic notations and terminology. In Section III we discuss the mutual information utility measure [8]. That discussion sets the ground for the introduction of our *private* mutual information utility measure in Section IV. Then, we proceed to describe in Section V our proposed sequential clustering algorithm. In Section VI we describe the experimental results.

## II. Notations and Terminology

Consider a database that holds information on individuals in some population. Each individual is described by a collection of $r$ public attributes, $A_1, \ldots, A_r$ (e.g. gender, age, address), and a private attribute, $A_{r+1}$ (that could represent, for example, a medical diagnosis for that individual, his credit limit etc.).[1] Each of the attributes consists of several possible values: $A_j = \{a_{j,\ell} : 1 \leq \ell \leq m_j\}$, $1 \leq j \leq r+1$. 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; letting $n$ denote the number of individuals, it takes the form

$$D = \{R_1, \ldots, R_n\}, \quad \text{where } R_i \in A_1 \times \cdots \times A_r. \quad (1)$$

The corresponding private database holds the private information,

$$\tilde{D} = \{S_1, \ldots, S_n\}, \quad S_i \in A_{r+1}. \quad (2)$$

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

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 public database with subsets of values, so that each entry $R_i(j) \in A_j$ is replaced by a subset $\overline{R}_i(j) \subseteq A_j$ that includes that element.

*Definition 2.1:* Let $A_j$, $1 \leq j \leq r$, be finite sets and let $\overline{A}_j$ 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 $(b_1, \ldots, b_r) \in A_1 \times \cdots \times A_r$, it holds that $(b_1, \ldots, b_r) \in g(b_1, \ldots, b_r)$.

If, for example, $\overline{A}_j$ consists of all singleton subsets plus the entire set, i.e. $\overline{A}_j = A_j \cup \{A_j\}$, this is the case of generalization by suppression (each entry either remains unchanged or is totally suppressed, e.g. [11]). A more refined scheme of generalization [1] is that in which there is a hierarchy of clusterings of $A_j$, the finest one consisting of all singleton subsets, and the coarsest one consisting of just the entire set.

There are two main models of generalization. In *global recoding*, each collection of subsets $\overline{A}_j$ is a clustering of the set $A_j$ and then every entry in the $j$th column of the database is mapped to the unique subset in $\overline{A}_j$ that contains it. As a consequence, every single value $a \in A_j$ is always generalized in the same manner. In *local recoding*, the collection of subsets $\overline{A}_j$ is a cover of the set $A_j$ but it is not a clustering. In that case, each entry in the table's $j$th column is generalized independently to one of the subsets in $\overline{A}_j$ that includes it. In such a model, if the age 34 appears in the table in several records, it may be left unchanged in some, and be generalized to $30-39$ or totally suppressed in other records. Clearly, local recoding is more flexible and might enable $k$-anonymity with smaller losses of information.

[1] We assume one private attribute for the sake of simplicity; the extension to any number of private attributes is straightforward.

## III. The Mutual Information Utility Measure

Many measures of loss of information were suggested in the study of $k$-anonymity, e.g., the Loss Metric [9], [12], the Ambiguity Metric [12], the Discernibility Metric [3], or the Classification Metric [9]. None of those measures was information-theoretic, even though they aim to measure information. An information-theoretic measure of information-loss was recently introduced in [8]. We describe it here in a manner that is based on the notion of mutual information. Our description is somewhat different from the one in [8]. Specifically, we describe it as a *utility* measure (denoted $U(g(D))$ rather than a measure of *information-loss*; as such, the goal is to maximize it (while measures of information-loss are sought to be minimized). The discussion here provides the technical background and motivation for the new utility measure that we introduce in the next section.

Let $D = \{R_1, \ldots, R_n\}$ be a database and let $A_1, \ldots, A_r$ be its public attributes. For each $1 \leq j \leq r$, denote by $X_j$ the random variable that corresponds to the attribute $A_j$. By looking at the table's $j$th column – $\{R_1(j), \ldots, R_n(j)\}$ – as the sample space for the variable $X_j$, we get the probability distribution:

$$\Pr(X_j = a) = \frac{|\{1 \leq i \leq n : R_i(j) = a\}|}{n}, \quad a \in A_j. \quad (3)$$

The entropy of $X_j$ is then defined as follows [5],

$$H(X_j) = -\sum_{a \in A_j} \Pr(X_j = a) \log \Pr(X_j = a). \quad (4)$$

First, we derive the mutual information utility measure in the case of *global* recoding. In such settings, each column in $g(D)$ includes subsets that constitute a clustering of the corresponding attribute. Letting $A_j$, $1 \leq j \leq r$, be one of the public attributes, the corresponding column in the generalized table includes values from $\hat{A}_j = \{C_1, \ldots, C_{t_j}\}$ where $\hat{A}_j$ is just a clustering of $A_j$ in the sense that $C_1, \ldots, C_{t_j}$ are disjoint subsets of $A_j$ whose union equal $A_j$. (For example, if $A_j$ is the age, $\hat{A}_j$ may consist of ranges of ages of the form $10-19$, $20-29$, $30-39$ etc.)

While the $j$th column in $D$ defines a random variable $X_j$ on $A_j$, the $j$th column in $g(D)$ defines a random variable $\hat{X}_j$ on $\hat{A}_j$, where for each $C_\ell \in \hat{A}_j$:

$$\Pr(\hat{X}_j = C_\ell) = \sum_{a \in C_\ell} \Pr(X_j = a).$$

The conditional entropy of $X_j$ given $\hat{X}_j$ is:

$$H(X_j | \hat{X}_j) = -\sum_{a \in A_j} \Pr(X_j = a) \log \Pr(X_j = a | \hat{X}_j = g(a)),$$

where $g(a)$ is the (unique) generalization of $a \in A_j$.

The mutual information between two random variables is a measure of the information that is disclosed on one of those variables by providing the value of the other one. The mutual information between $X_j$ and $\hat{X}_j$ is:

$$I(X_j; \hat{X}_j) = H(X_j) - H(X_j|\hat{X}_j) =$$

$$\sum_{a \in A_j} \Pr(X_j = a) \log \frac{\Pr(X_j = a|\hat{X}_j = g(a))}{\Pr(X_j = a)} .$$

Using (3) we get that

$$I(X_j; \hat{X}_j) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\Pr(X_j = R_i(j)|X_j \in \overline{R}_i(j))}{\Pr(X_j = R_i(j))} . \quad (5)$$

The mutual information between the tuples $\langle X_1, ..., X_r \rangle$ and $\langle \hat{X}_1, ..., \hat{X}_r \rangle$ is a natural way to measure the information that the anonymized table reveals on the original table. However, the relative sparsity of the multidimensional data makes the empirical estimation unreliable. Hence, we use instead an approximation based on the assumption that the attribute random variables are independent (an assumption that implicitly underlies all previously used measures). This yields the *mutual information utility measure* $U(g(D)) := I(D; g(D))$ where $I(D; g(D))$ is the following mutual information,

$$I(D; g(D)) := \frac{1}{r} \sum_{j=1}^{r} I(X_j; \hat{X}_j) .$$

Hence, the goal is to find a clustering of each of the attributes that will render the database $k$-anonymized while keeping the mutual information, $I(D; g(D))$, maximal.

Having defined the mutual information utility measure in the case of global recoding, we proceed to define it in the case of local recoding. Assuming that $\overline{A}_j$ is the collection of subsets of $A_j$ that may be used as generalized values, the generalized table $g(D)$ takes the form $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ where $\overline{R}_i(j) \in \overline{A}_j$. Although we cannot formalize this local generalization as a joint distribution of the two random-variables (the original one and the generalized one) we can still apply the local interpretation of the mutual information between the $j$th column in the original table and the corresponding column in the anonymized table, (5). Therefore, the preserved information per attribute can still be written as

$$I(X_j; \overline{R}(j)) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\Pr(X_j = R_i(j)|X_j \in \overline{R}_i(j))}{\Pr(X_j = R_i(j))} \quad (6)$$

where $\overline{R}(j)$ stands for the $j$th column in $g(D)$. Finally, the mutual information (MI) utility measure is $U(g(D)) := I(D; g(D))$ where

$$I(D; g(D)) := \frac{1}{r} \sum_{j=1}^{r} I(X_j; \overline{R}(j)) \quad (7)$$

$$= \frac{1}{nr} \sum_{i=1}^{n} \sum_{j=1}^{r} \log \frac{\Pr(X_j = R_i(j)|X_j \in \overline{R}_i(j))}{\Pr(X_j = R_i(j))} .$$

The corresponding mutual information measure of information-loss is

$$\Pi_{MI}(D, g(D)) = \quad (8)$$

$$-\frac{1}{nr} \sum_{i=1}^{n} \sum_{j=1}^{r} \log \Pr(X_j = R_i(j)|X_j \in \overline{R}_i(j)) .$$

Clearly, $\Pi_{MI}(D, g(D))$ is minimized when the utility measure $U(g(D)) = I(D; g(D))$ in (7) is maximized.

## IV. The Private Mutual Information Utility Measure

All previously used measures of information-loss are based entirely on the public information and ignore the private information. (The only exception is the Classification Metric, which is a very basic measure that was suggested in [9].) However, one should keep in mind that one of the main goals in publishing the database is to learn the relation between the public data and private data and to deduce methods of predicting the private data from the public data. Therefore, the information-loss caused by the generalization process should be measured in the context of this prediction task. Specifically, generalization of public attributes that are weakly correlated with the private data should be less penalized than generalization of other public attributes that are strongly correlated with the private data.

We proceed to present here a new utility measure that quantifies the mutual information between the generalized public data and the private data. So, instead of looking at $I(D; g(D))$ (namely, how much information do the generalized public data reveal on the original public data), we look at $U(g(D)) := I(\tilde{D}; g(D))$ – the amount of information that the generalized public data reveal on the private data. (Recall that $\tilde{D} = \{S_1, \ldots, S_n\}$, where $S_i \in A_{r+1}$, is the $i$th private record, see Equation (2).) As before, let $X_j$ denote the random variable that corresponds to the $j$th public attribute and let $\overline{R}(j)$ stand for the $j$th column in $g(D)$, $1 \leq j \leq r$. In addition, we introduce the random variable $Y$ that corresponds to the private attribute. (The probability distribution of $Y$ on the set of possible values for the attribute $A_{r+1}$ is derived from the private database $\tilde{D}$ in similarity to the way that we defined the probability distribution of $X_j$ according to the $j$th column in $D$.) In a way similar to definition (6) in the previous section, we define the mutual information between $Y$ and the anonymized version of the $j$th public attribute $X_j$ as follows:

$$I(Y; \overline{R}(j)) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\Pr(Y = S_i|X_j \in \overline{R}_i(j))}{\Pr(Y = S_i)} . \quad (9)$$

The mutual information $I(Y; \overline{R}(1), ..., \overline{R}(r))$ can be utilized to measure the information that the anonymized table reveals

on the private data. However, as discussed in the previous section, the relative sparsity of the multidimensional data makes the empirical estimation unreliable. Hence, we approximate that expression with the following one that can be easily computed:

$$I(\tilde{D}; g(D)) := \frac{1}{r} \sum_{j=1}^{r} I(Y; \overline{R}(j)) \,. \tag{10}$$

The goal is then to maximize $U(g(D)) := I(\tilde{D}; g(D))$ that is defined through (9)+(10), i.e.,

$$I(\tilde{D}; g(D)) = \frac{1}{nr} \sum_{i=1}^{n} \sum_{j=1}^{r} \log \frac{\Pr(Y = S_i | X_j \in \overline{R}_i(j))}{\Pr(Y = S_i)} \,. \tag{11}$$

We refer to this utility measure as the *private mutual information utility measure* (PMI). The corresponding information-loss measure is

$$\Pi_{PMI}(D, g(D)) = \tag{12}$$

$$-\frac{1}{nr} \sum_{i=1}^{n} \sum_{j=1}^{r} \log \Pr(Y = S_i | X_j \in \overline{R}_i(j)) \,.$$

It expresses the amount of mutual information that is lost by replacing $D$ with $g(D)$. Clearly, $\Pi_{PMI}(D, g(D))$ is minimized when $I(\tilde{D}; g(D))$ is maximized.

The PMI utility measure is defined in (10) as an average of the mutual information between the private attribute and each of the generalized public attributes. As such an averaging might hide a strong correlation of one of the generalized public attributes with the private attribute, another possible definition of that measure is

$$I(\tilde{D}; g(D)) := \max_{1 \le j \le r} I(Y; \overline{R}(j)) \,. \tag{13}$$

A possible compromise between the $\ell_1$-norm in (10) and the $\ell_\infty$-norm in (13) is the $\ell_2$-norm version

$$I(\tilde{D}; g(D)) := \frac{1}{r} \left( \sum_{j=1}^{r} I(Y; \overline{R}(j))^2 \right)^{1/2} \,. \tag{14}$$

In this study we focus on the first $\ell_1$-version, (10). The comparison between the effectiveness of the different versions is left for future experiments and study.

We now turn to discuss the monotonicity of the PMI utility measure, $U(g(D)) = I(\tilde{D}; g(D))$. A natural property that one might expect from any utility measure is monotonicity. In other words, we expect that coarser generalizations will be characterized by smaller values of the utility measure. We focus here on the case of global recoding. Assume that $g'(D)$ is a generalization that is coarser than $g(D)$. Since both $g(D)$ and $g'(D)$ are based on global recoding, they define for the $j$th attribute two random variables, $\hat{X}_j$ and $\hat{X}_j'$ respectively (see Section III). The random variables $Y, X_j, \hat{X}_j$ and $\hat{X}_j'$ form a Markov chain:

$$Y \longleftarrow X_j \longrightarrow \hat{X}_j \longrightarrow \hat{X}_j' \,.$$

Hence, the Data Processing Lemma [5] implies that $I(Y; \hat{X}_j) \ge I(Y; \hat{X}_j')$. Summing up those inequalities for all $1 \le j \le r$ and dividing by $r$ we conclude that $I(\tilde{D}; g(D)) \ge I(\tilde{D}; g'(D))$.

In the more general case of generalization based on local recoding, the PMI utility measure (11) is not always monotone. For example, let $D||\tilde{D}$ be the following table with a single public attribute and a single private attribute:

| $D$ | a | a | a | a | b | b | b | b | c |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{D}$ | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

(15)

Consider the following 3-anonymization of $D$:

| $g(D)$ | a | a | a | * | * | b | b | b | * |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{D}$ | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

(16)

In this case, by (12), $\Pi_{PMI}(D, g(D)) \approx -0.126 < 0$. Therefore, $I(\tilde{D}; D) < I(\tilde{D}; g(D))$, even though $g(D)$ is a generalization of $D$, whence monotonicity is violated. Nonetheless, the PMI utility measure is still meaningful for local recoding and serves well the purposes of data-mining, as we proceed to explain. The generalization $g(D)$ in (16) is the 3-anonymization that maximizes $I(\tilde{D}; g(D))$, since it selects to obfuscate the outlier records. The PMI utility measure favors the generalization $g(D)$ over the original table $D$ in (15), since the latter has outlier records that blur the two prominent association rules "a implies 0" and "b implies 1", while the former eliminates those outliers and accentuates those two rules.

The above example exemplifies the advantage that our newly proposed utility measure has to offer with respect to the previous measures. Measures that rely only on the public attributes cannot distinguish between the various 3-anonymizations of $D$ and cannot identify $g(D)$ as the best one. Hence, using the PMI measure may yield anonymized tables with greater utility for data mining.

## V. Algorithms for $k$-Anonymity

The problem of finding a $k$-anonymization of a given table with minimal information loss is NP-hard. Several polynomial-time approximation algorithms were devised for this problem. The first one [11] has an approximation guarantee of $O(k \log n)$ and runtime of $O(rn^2 + n^3)$ (for the case of suppressions only). The algorithm in [1] runs in time $O(kn^2)$ and approximates the optimal solution to within $O(k)$ (for the case of generalization by hierarchical clustering). As $k$ may be relatively large in practice, those approximation factors might be unsatisfactory. A significant improvement was proposed in [8], with an $O(\log k)$-approximation algorithm that applies to any generalization

and any measure. Alas, its run time, $O(n^{2k})$, renders it impractical. A more efficient $O(\log k)$-approximation algorithm was proposed in [13], but it is restricted only to generalizations by suppression.

Due to the poor performance and limitations of the provable approximation algorithms, heuristical algorithms are invoked. The current popular approach is agglomerative algorithms that are based on a bottom-up merging procedure (Section V-B). We describe here (Section V-C) an alternative approach, based on sequential clustering, that we propose in this context. In order to discuss those two approaches, we begin by providing the basic definitions of cluster closure and generalization cost.

### A. Generalization cost

Any $k$-anonymization induces a clustering of the records in $D$ to clusters of size at least $k$. Conversely, every clustering of $D$ into clusters of size at least $k$ induces a $k$-anonymization, $g(D)$, in the following manner. Assume that $\{R_{i_1}, \ldots, R_{i_m}\}$ is one of the clusters. Then the records $\overline{R}_{i_1}, \ldots, \overline{R}_{i_m}$ in $g(D)$ will be all equal, and their $j$th entry will be the minimal set in $\overline{A}_j$ that includes the values $R_{i_1}(j), \ldots, R_{i_m}(j)$. We aim at finding such a clustering that induces an optimal $k$-anonymization under a given measure of loss of information.

Let $\mathcal{C} = \{C_1, \ldots, C_t\}$ be a clustering of the records in $D$, where all clusters are of size at least $k$. Such a clustering induces a $k$-anonymization $g(D)$ of $D$. Letting $\Pi$ be some measure of information-loss, we proceed to define an anonymization cost, $\mathsf{gc}$, for each of the clusters, $C_i$, $1 \leq i \leq t$. The generalization cost $\mathsf{gc}$ will be defined so that the information-loss of the anonymization $g(D)$ will be given by

$$\Pi(D, g(D)) = \frac{1}{n} \sum_{j=1}^{t} \mathsf{gc}(C_j) \cdot |C_j| \,. \tag{17}$$

In other words, we wish to define $\mathsf{gc}$ so that the average over all $n$ records in $D$ of the $\mathsf{gc}$ value of that record's cluster will be the information-loss of the anonymization $g(D)$ that is induced by that clustering.

We proceed to define $\mathsf{gc}$. Let $C$ be one of the clusters in $\mathcal{C}$. Without loss of generality, we assume that $C = \{R_1, \ldots, R_m\}$. The *closure* of $C$ is the minimal generalized record $\overline{R}$ that generalizes every single record in $C$. Namely, for all $1 \leq j \leq r$, $\overline{R}(j)$ is the minimal set in the collection $\overline{A}_j$ that includes all of the values $R_1(j), \ldots, R_m(j)$. In the anonymized table $g(D)$ that corresponds to the clustering $\mathcal{C}$, all records in $C$ will be replaced by the closure of $C$. Then the corresponding generalization cost of $C$, $\mathsf{gc}(C)$, is the average information loss that is caused by replacing each of the records in $C$ by the generalized record $\overline{R} = (\overline{R}(1), \ldots, \overline{R}(r))$.

For the MI measure, (8), and the PMI measure, (12), the generalization cost may differ from one record to another in the same cluster. In the former, it depends on the original public attributes in the record,

$$\mathsf{gc}_{MI}(C) = -\frac{1}{mr} \sum_{i=1}^{m} \sum_{j=1}^{r} \log \Pr(X_j = R_i(j) | X_j \in \overline{R}_i(j)) \,,$$

while in the latter it depends on the private attribute in the record,

$$\mathsf{gc}_{PMI}(C) = -\frac{1}{mr} \sum_{i=1}^{m} \sum_{j=1}^{r} \log \frac{\Pr(Y = S_i | X_j \in \overline{R}_i(j))}{\Pr(Y = S_i | X_j = R_i(j))} \,.$$

### B. Agglomerative algorithms

Agglomerative algorithms were proposed in [4], [7], [12]. The basic idea in such algorithms is to start with singleton clusters and then keeping unifying the two closest clusters until all clusters become larger than $k$. A key ingredient in all agglomerative algorithms is the definition of distance between clusters. It is natural to define the distance so that it best fits the cost function of the $k$-anonymization. We used in our experiments one of the distance functions that were proposed in [7],

$$\text{dist}(A, B) = |A \cup B| \cdot \mathsf{gc}(A \cup B) - |A| \cdot \mathsf{gc}(A) - |B| \cdot \mathsf{gc}(B) \,,$$

which, in view of (17), expresses the difference in the overall generalization cost if we unify the clusters $A$ and $B$.

### C. Sequential Clustering Algorithm

The most fundamental non-agglomerative clustering technique is $K$-means. As the number of clusters is unknown, but is bounded from above by $\lfloor n/k \rfloor$, we may set $K$ to a number in the vicinity of that upper bound, select $K$ random centers, and then use any of the utility measures that were defined in the previous sections as the underlying metric. Alternatively, we can apply a greedy sequential algorithm that can be viewed as a sequential version of the $K$-means algorithm. The sequential greedy algorithm is known to perform well in terms of both clustering quality and computational complexity [15].

The sequential clustering algorithm starts with a random partition of the records into clusters. Then, it goes over the $n$ records in a cyclic manner and for each record checks whether it may be moved from its current cluster to another one while increasing the utility of the induced anonymization. This loop may be iterated until either we reach a local optimum (i.e., a stage in which no single-record transition offers an improvement) or the local improvements of the utility become sufficiently small. As there is no guarantee that such a procedure finds the global optimum, it may be repeated several times with different random partitions as the starting point, in order to find the best local optimum among those repeated searches.

Usually, the number of clusters is given as an input to the algorithm. In agglomerative (bottom-up) algorithms, the merging process is executed until the desired number of

clusters is obtained. Sequential clustering algorithms are initialized with a random clustering having the specified number of clusters. However, in our $k$-anonymization clustering problem, the constraint is on the size of the clusters rather than on their number. To cope with this constraint, the scheduling of the sequential algorithm should be modified.

A possible solution is the following. The initial number of clusters in the random clustering is set to $\lfloor n/k_0 \rfloor$ and the initial clusters are chosen so that all of them are of size $k_0$ or $k_0+1$, where $k_0 = \alpha k$ is an integer and $\alpha$ is some parameter that needs to be determined. Then, during the sequential algorithm, we allow the size of the clusters to vary in the range $[2, \omega k]$, for some predetermined fixed parameter $\omega$.

When a cluster becomes a singleton, we remove it and place that record in one of the other clusters where it fits best. If a cluster becomes too large (i.e., its size becomes larger than the upper bound $\omega k$), we split it into two equally-sized clusters in a random manner. The main loop of the algorithm is repeated until we reach a stage where an entire loop over all records of the database found no record that could be moved to another cluster in order to improve the utility. At this point, some of the clusters are large, in the sense that their size is at least $k$, while others are small. If there exist small clusters, we apply the agglomerative algorithm on those clusters in order to merge them into larger clusters of size $k$ or more. Finally, if we are left at the end with a single cluster that is small, we merge it with the closest large cluster. This approach is summarized in Algorithm 1.

Note that the agglomerative algorithm is in fact a special case of this sequential algorithm that corresponds to the selection $k_0 = 1$.

*1) Measuring the information loss:* Let $C_j$ and $C_\ell$ be two clusters in $\mathcal{C}$, and assume that $R_i$ is a record in $C_j$. A basic check that the algorithm performs is whether we may gain utility by moving $R_i$ from $C_j$ to $C_\ell$. In view of (17), the difference in the overall information loss as a result of such an action is

$$\begin{aligned}\Delta_{i:j\to\ell} = \frac{1}{n} \cdot \{ & [\mathsf{gc}(C_j \setminus \{R_i\}) \cdot (|C_j| - 1) \\ & + \mathsf{gc}(C_\ell \cup \{R_i\}) \cdot (|C_\ell| + 1)] - \\ & [\mathsf{gc}(C_j) \cdot |C_j| + \mathsf{gc}(C_\ell) \cdot |C_\ell|] \} \ .\end{aligned} \tag{18}$$

If that difference is negative, then we gain from such a transition. Therefore, in each step of the algorithm we reexamine the current location of each of the records $R_i$ in $D$ and then look for an alternative location (cluster) that provides the best improvement in terms of information loss.

It is preferable to have in the final clustering clusters of size close to $k$, since larger clusters imply lesser utility. One way of controlling the cluster sizes is by selecting properly the size of the initial clusters, $k_0 = \alpha k$, and by selecting the upper limit of cluster size, $\omega k$. Our tests indicated that it is preferable to set $\alpha$ to a value smaller than 1 (namely, initially all clusters are smaller than $k$), and to set $\omega$ to a value smaller than 2. In all of our tests we used $\alpha = 0.5$ and $\omega = 1.5$.

**Algorithm 1** Sequential clustering algorithm for $k$-anonymization

**input** Table $D = \{R_1, \ldots, R_n\}$, integer $k$.
**output** A clustering of $D$ into clusters of size at least $k$.

1: Choose a random partition of the data records into $t := \lfloor n/k_0 \rfloor$ clusters of sizes either $k_0$ or $k_0 + 1$. Denote the clusters by $C_1, \ldots, C_t$.
2: **for** $i = 1, \ldots, n$ **do**
3: Let $C_j$ be the cluster to which record $R_i$ currently belongs.
4: For each of the other clusters, $C_\ell$, $\ell \neq j$, compute the difference in the information loss if we move $R_i$ from $C_j$ to $C_\ell$ — $\Delta_{i:j\to\ell}$.
5: Let $C_{\ell_0}$ be the cluster for which $\Delta_{i:j\to\ell}$ is minimal.
6: If $C_j$ is a singleton, move $R_i$ from $C_j$ to $C_{\ell_0}$ and remove cluster $C_j$.
7: Else, if $\Delta_{i:j\to\ell_0} < 0$, move $R_i$ from $C_j$ to $C_{\ell_0}$.
8: **end for**
9: If there exist clusters of size greater than $\omega k$, split each of those clusters randomly into two equally-sized clusters.
10: If at least one record was moved during the last loop, go to Step 2.
11: **while** the number of clusters of size smaller than $k$ is greater than 1 **do**
12: Unify the two closest small clusters.
13: **end while**
14: If there exists a small cluster, unify it with the cluster to which it is closest.
15: Output the resulting clustering.

## VI. Experiments

In Section VI-A we describe experiments that demonstrate the advantages of the sequential algorithm over the agglomerative algorithm. In Section VI-B we describe the experiments that compare our proposed PMI measure to the MI measure; those experiments used the sequential algorithm.

### A. Comparing the Sequential and Agglomerative Algorithms

We tested our algorithm versus the agglomerative algorithm on the dataset `Adult` from the UCI Machine Learning Repository.[2] That dataset was extracted from the US Census Bureau Data Extraction System. It contains demographic information of a small sample of US population with 14 public attributes such as age, education-level, marital-status,

[2] http://mlearn.ics.uci.edu/MLSummary.html

occupation, and native-country. The private information is an indication whether that individual earns more or less than 50 thousand dollars annually. The adult data contains 45,222 records after tuples with missing values are removed.

In the experiments that we report herein we restricted our attention to generalization by suppression. In addition, we implemented a non-repetitive version of the sequential algorithm (i.e., one that does not perform the sequential clustering several times, each time starting with a different random initial clustering). We ran each of the two algorithms with seven values of the anonymity parameter, $k = 10, 20, 30, 40, 50, 75, 100$, and using three measures – LM, MI, and our proposed PMI. (Namely, each of the two algorithms ran on the `Adult` dataset 21 times in total.) For each of those three measures and seven values of $k$, the two algorithms issued generalizations with equivalent anonymity (as dictated by the value of $k$) and with the same utility score. However, while the two algorithms offer comparable generalizations, they differ significantly in their runtime. The runtime of the sequential algorithm was much faster than that of the agglomerative one: Figure 1 reports the average runtime of each of the two algorithms for each value of $k$ (the average being taken over the three different runs with respect to the three utility measures). The two algorithms were implemented in C on a Pentium (R) 4 CPU 3.40 GHz, 1.49 GB of RAM.

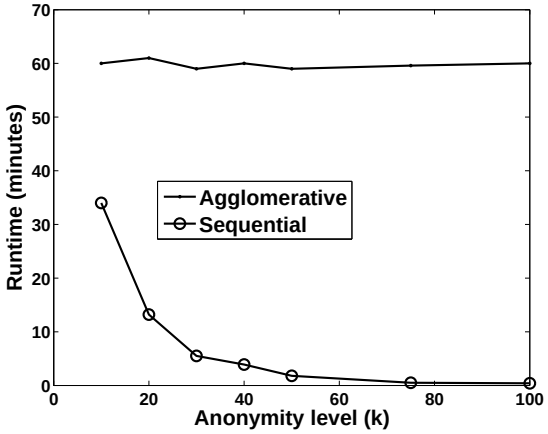


Figure 1. Runtime comparison between the agglomerative and sequential algorithms.

Note that the runtime of the agglomerative algorithm slightly increases with $k$. Indeed, regardless of $k$, the agglomerative algorithm starts with $n$ clusters and the main computational effort occurs in the early iterations where the number of clusters is still $O(n)$. The number of clusters reduces by 1 in each iteration until all clusters are of size at least $k$ whence the final number of clusters is $O(n/k)$. Therefore, the number of iterations in the agglomerative algorithm is roughly $n - n/k$.

The runtime of the sequential algorithm, on the other hand, decreases with $k$. Since the number of clusters is $O(n/k)$, then each pass over all records in the table involves $O(n^2/k)$ computations of utility gain by moving a record from one cluster to another. The faster than $O(1/k)$ decrease in the runtime stems from the fact that the number of iterations also reduces with $k$. Since, in practice, higher values of $k$ are required for greater privacy, the advantage offered by the sequential algorithm over the agglomerative one in terms of runtime is prominent.

We tested also the dependence of the runtime of the two algorithms with respect to the number of public attributes. We ran the two algorithms on a reduced `Adult` dataset that has only 7 public attributes (selected out of the existing 14 attributes) and on an extended `Adult` dataset that has 28 public attributes (the additional 14 attributes were artificial ones). As expected, the runtime of each of the two algorithms depends linearly on the number of public attributes, regardless of the value of $k$ or the underlying utility measure.

### *B. Comparing the PMI and MI Measures*

After establishing the superiority of sequential clustering over agglomerative clustering, we proceeded to test the PMI utility measure and compare it to the MI measure. To that end, we recall the definition of *diversity* as appears in [10]. Given a cluster of anonymized records, its diversity is the entropy of the distribution that it induces on the private attribute. For example, if all records in that cluster have the same private value then that cluster's diversity is zero; but if, on the other hand, all records have a unique private value then the diversity is $\log m$ where $m$ is the cluster size. On one hand, we wish to arrive at a clustering in which every cluster has a low diversity since that would indicate a strong correlation between the generalized public data and the private data. On the other hand, a too low diversity (that helps learning) might jeopardize the privacy of the individuals in that cluster. Therefore, Machanavajjhala et. al. [10] suggested to impose a minimal diversity as a privacy measure.

We ran the sequential clustering on the `Adult` database with a weighted MI measure,

$$U_{wMI} = w \cdot U_{MI} + (1 - w) \cdot U_{PMI}\,,$$

with $w = 0, .25, .5, .75, 1$, for $k = 50, 75, 100$. The average diversities of the resulting clusterings in each of those cases are shown in Table 1.

| $k\backslash w$ | 0 | 0.25 | 0.5 | 0.75 | 1 |
|---|---|---|---|---|---|
| 50 | 0.07 | 0.14 | 0.31 | 0.51 | 0.54 |
| 75 | 0.08 | 0.14 | 0.32 | 0.51 | 0.56 |
| 100 | 0.08 | 0.15 | 0.34 | 0.54 | 0.58 |

Table I
AVERAGE DIVERSITIES FOR DIFFERENT VALUES OF $k$ AND $w$.

We see that when $w = 0$ (which corresponds to the PMI measure) the correlation between the generalized public data and the private one is much stronger than in the case $w = 1$ (which corresponds to the MI measure). Hence, it is apparent that anonymizations that were obtained by using the PMI measure clearly are more valuable for mining association rules.

It should be pointed out that in all of our experiments (either with the above $U_{wMI}$ measures or with other measures such as the LM) there were clusters with zero diversity. Hence, the problem that was identified in [10] does occur, regardless of the utility measure (or the clustering algorithm). Hence, it is necessary to impose also $\ell$-diversity in the sense that each final cluster has diversity at least $\ell$. We note that sequential clustering, due to its flexibility, can be easily modified to guarantee also $\ell$-diversity, as opposed to agglomerative clustering which is more rigid. Indeed, each basic step in sequential clustering moves a record from its current cluster to another cluster if such a transition increases the utility. By starting with initial clustering that respects $\ell$-diversity and performing only transitions of records that do not violate $\ell$-diversity, we may apply sequential clustering to obtain $k$-anonymized as well as $\ell$-diversified tables. Agglomerative clustering, on the other hand, is less accommodating for $\ell$-diversification since it starts with a clustering that violates $\ell$-diversity (as all initial clusters have zero diversity) and the basic operation in that algorithm is the unification of clusters rather than transitions of single records.

## VII. Conclusions

In this study we proposed the private mutual information (PMI) utility measure that aims at maximizing the correlation between the generalized public data and the private data. We showed that this measure is much more adequate for the purposes of data mining that aims at finding association rules to predict the private data from the public data. We then described the sequential clustering algorithm. That algorithm, which is independent of the underlying utility measure, is comparable to agglomerative clustering in terms of the resulting utility, but it is significantly faster, and it may be $\ell$-diversified more easily.

Our initial experiments regarding the diversity show that the PMI measure is much more suitable when the goal is to achieve anonymizations from which association rules or methods of predicting the private data from the public data can be mined. A more thorough experimental validation of this claim will proceed as follows: We intend to obtain several $k$-anonymizations of the same table using different measures of information-loss. Then each of those tables will be used either for mining association rules or for the computation of a classifier. Our conjecture, which is supported by our initial experiments that we reported here, is that the PMI-related table will produce a set of association rules which is closer to the set of association rules that can be mined from the original table; also, the PMI-derived classifier is believed to be more accurate than a classifier that is based on anonymizations that are based on other measures of information-loss. It should be noted that the actual design of such experiments requires a substantial theoretical study and the derivation of methods that, to the best of our knowledge, are still not available.

## References


[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for $k$-anonymity. *J. of Privacy Tech.*, 2005.

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

[3] R. Bayardo and R. Agrawal. Data privacy through optimal $k$-anonymization. In *ICDE*, 2005.

[4] J.W. Byun, A. Kamra, E. Bertino, and N.Li. Efficient k-anonymization using clustering techniques. In *DASFAA*, 2007.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1991.

[6] J. Domingo-Ferrer and V. Torra. A critique of k-anonymity and some of its enhancements. In *ARES*, 2008.

[7] A. Gionis, A. Mazza, and T. Tassa. $k$-Anonymization revisited. In *ICDE*, 2008.

[8] A. Gionis and T. Tassa. $k$-Anonymization with minimal loss of information. In *ESA*, 2007.

[9] V. Iyengar. Transforming data to satisfy privacy constraints. In *SIGKDD*, 2002.

[10] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. $l$-Diversity: privacy beyond $k$-anonymity. In *ICDE*, 2006.

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

[12] M. E. Nergiz and C. Clifton. Thoughts on $k$-anonymization. In *ICDE Workshops*, 2006.

[13] H. Park and K. Shim. Approximate algorithms for $k$-anonymity. In *SIGMOD*, 2007.

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

[15] N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *ACM SIGIR*, 2002.

[16] R. Chi wing Wong, J. Li, A. Wai chee Fu, and K. Wang. ($\alpha$, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *ACM SIGKDD*, pages 754–759, 2006.