Beyond Multivariate Microaggregation for Large Record Anonymization

Jordi Nin

Barcelona Supercomputing Center (BSC)
Universitat Politècnica de Catalunya (BarcelonaTech)
Barcelona, Catalonia, Spain
nin@ac.upc.edu

Abstract. Microaggregation is one of the most commonly employed microdata protection methods. The basic idea of microaggregation is to anonymize data by aggregating original records into small groups of at least k elements and, therefore, preserving k-anonymity. Usually, in order to avoid information loss, when records are large, i.e., the number of attributes of the data set is large, this data set is split into smaller blocks of attributes and microaggregation is applied to each block, successively and independently. This is called *multivariate microaggregation*. By using this technique, the information loss after collapsing several values to the centroid of their group is reduced. Unfortunately, with multivariate microaggregation, the k-anonymity property is lost when at least two attributes of different blocks are known by the intruder, which might be the usual case.
In this work, we present a new microaggregation method called *one dimension microaggregation* ($Mic1D-k$). With $Mic1D-k$, the problem of k-anonymity loss is mitigated by mixing all the values in the original microdata file into a single non-attributed data set using a set of simple pre-processing steps and then, microaggregating all the mixed values together. Our experiments show that, using real data, our proposal obtains lower disclosure risk than previous approaches whereas the information loss is preserved.

Keywords: Microaggregation, k-anonymity, Privacy in Statistical Databases

1 Introduction

Managing confidential data is a common practice in any organization. In many cases, these data contain valuable statistical information required by third parties for data analysis and, thus, privacy becomes essential, making it necessary to release data sets preserving the statistics without revealing confidential information. This is a typical problem, for instance, in statistics institutes.

One of the most popular approaches to achieve such a privacy level is to apply *perturbative protection methods* on the microdata source file to be protected. This approach consists in distorting the original data file so that the resulting data, which is publicly released, does not permit the disclosure of sensitive information.

A large number of protection methods exist (see e.g. [1], [6] and [28]). Apart from protecting the privacy of the confidential information, perturbative data protection methods must preserve the statistical utility of the original data as far as possible. In this situation, the main challenge is to find the trade-off between privacy and statistical utility.

Recently, microaggregation has emerged as one of the most promising perturbative data protection methods. For example, [9] shows that microaggregation is used by many statistical agencies for data anonymization. The basic implementation of microaggregation works as follows [6, 7, 23]: given a data set with n_{att} attributes, small clusters of at least k elements (records) are built and each original record is replaced with the centroid of the cluster to which the record belongs to. A certain level of privacy is ensured because k records have an identical protected value (k-*anonymity* [22, 25, 26]).

However, when n_{att} is large, the statistical utility of the basic microaggregation technique is diminished, specially if the attributes are not highly correlated [2]. This is so because the larger the number of attributes, the larger the distance between the original records in the data set and their corresponding centroids. Therefore, a lot of information on the original data is lost when the protected microdata file is released. To solve this drawback, the following natural strategy is applied by statistical agencies: the microdata file is split into smaller blocks of attributes, and microaggregation is independently applied to each block. This way, the information loss decreases, at the cost of decreasing the achieved level of privacy since the property of k-anonymity is not ensured, as we see later on in this paper. This kind of microaggregation methods are known as *multivariate microaggregation* methods. Another important drawback of this type of methods is that finding the optimal multivariate microaggregation (*i.e.*, finding the clusters that minimize the sum of square errors) is NP-hard [20].

In this work, we propose to combine a set of preprocessing steps along with microaggregation in order to minimize the disclosure risk without losing information. We test this new method using real data showing that $Mic1D-k$ is able to outperform previous multivariate microaggregation methods diminishing the risk of disclosure without increasing the information loss. Specifically, we compare our new method with some of the most commonly used microaggregation methods, showing that $Mic1D-k$ achieves lower disclosure risk than previous algorithms when different groups of attributes are known by an intruder.

This paper is organized as follows. In Section 2 we review some basic concepts related to protection methods, focusing on microaggregation techniques. In Section 3, we present our new microaggregation method called *One dimension microaggregation*. Section 4 is devoted to compare traditional microaggregation algorithms and our new microaggregation method using real data; we present our experiments and the obtained results. Finally, Section 5 draws some conclusions and presents some future work.

2 Preliminaries on Protection Methods

In this section we present some basic concepts that will be useful for the understanding of the work presented in this paper. Namely, we first describe the scenario where a microdata protection method is applied to preserve the privacy of the owners of some statistical data. Then, we explain in detail several microaggregation techniques. Finally, we describe the most usual ways to measure the quality of microaggregation methods, according to the levels of privacy and statistical utility that they provide.

2.1 Statistical Data Protection

Let a microdata file X be a matrix with n rows (*records*) and n_{att} columns (*attributes*), where each row contains n_{att} attributes of an individual. The attributes in a microdata file can be classified into two different categories, *identifiers* or *quasi-identifiers*, depending on their capability to identify unique individuals. Identifier attributes are used to identify the individual unambiguously. The matrix containing all the values related to these attributes will be denoted in this paper by Id. A typical example of identifier is the passport number. A quasi-identifier attribute is an attribute that is not able to identify a single individual when it is used alone. However, when it is combined with other quasi-identifier attributes, they can uniquely identify an individual. Among the quasi-identifier attributes, we distinguish between *confidential* (X_c) and *non-confidential* (X_{nc}), depending on the kind of information they contain. Therefore, we define a microdata file as $X = (Id, X_{nc}, X_c)$. A first naive approach would be to eliminate the identifier attributes and release (X_{nc}, X_c) in order to avoid the linkage of confidential data (X_c) to real individuals. In this scenario, an intruder would be able to re-identify individuals by obtaining the non-confidential quasi-identifier attributes together with identifiers from other data sources and, therefore, disclosing confidential information.

In order to preserve statistical disclosure control, we use assume the solution proposed in [6] to compare several protection methods. This solution is graphically depicted in Figure 1 and it works as follows:

(i) Identifier attributes in X are either removed or encrypted.
(ii) Confidential quasi-identifier attributes X_c are not modified; in this way, the statistical utility of the confidential attributes is completely preserved.
(iii) A microdata protection method ρ is applied to non-confidential quasi-identifier attributes, in order to preserve the privacy of the individuals whose confidential data is being released, $X'_{nc} = \rho(X_{nc})$.
(iv) The released microdata file is $X' = (\rho(X_{nc}), X_c)$.

In this scenario, as shown in Figure 2, an intruder might try to re-identify individuals by obtaining the non-confidential quasi-identifier data (X_{nc}) together with identifiers (Id) from other data sources. By applying record linkage between the protected attributes (X'_{nc}) and the same attributes obtained from other

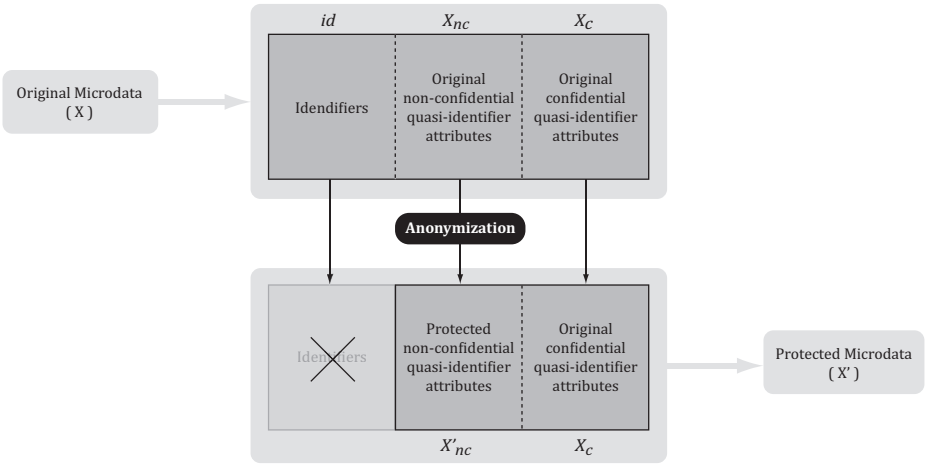

Fig. 1. Data protection and release process.

data sources (X_{nc}), the intruder might be able to re-identify a percentage of the protected individuals together with their confidential data (X_c). The quality of a protection methods depends on the percentage of information that it allows to re-identify, among other aspects.

2.2 Microaggregation

As introduced before, microaggregation ensures k-*anonymity* by building small clusters of at least k elements and replacing the original values by the centroid of the cluster to which the record belongs to.

There are other ways to achieve k-anonymity. For instance, in [3] authors present a clustering technique where the released microdata file preserves k-anonymity, as in basic microaggregation. In other solutions, such as those presented in [10], the data holder chooses different subsets of attributes ensuring k-anonymity for each of these subsets independently, similarly to multivariate microaggregation.

We have seen that, in order to solve the information loss problem of the basic microaggregation method, *multivariate microaggregation* is used at the cost of increasing the disclosure risk. Specifically, after dividing attributes into different blocks and applying the basic microaggregation technique to each block separately, the k records which fall in the same cluster for the first block of attributes, may fall in a different cluster for any of the other blocks of attributes. So, the resulting protected records will not be equal and no k-anonymity is ensured. The easiest case for microaggregation in terms of attribute blocking complexity occurs when the size of the attribute blocks is equal to one. In other words, when each attribute is protected independently. This corresponds to *Univariate Microaggregation* or *Individual Ranking Microaggregation*.

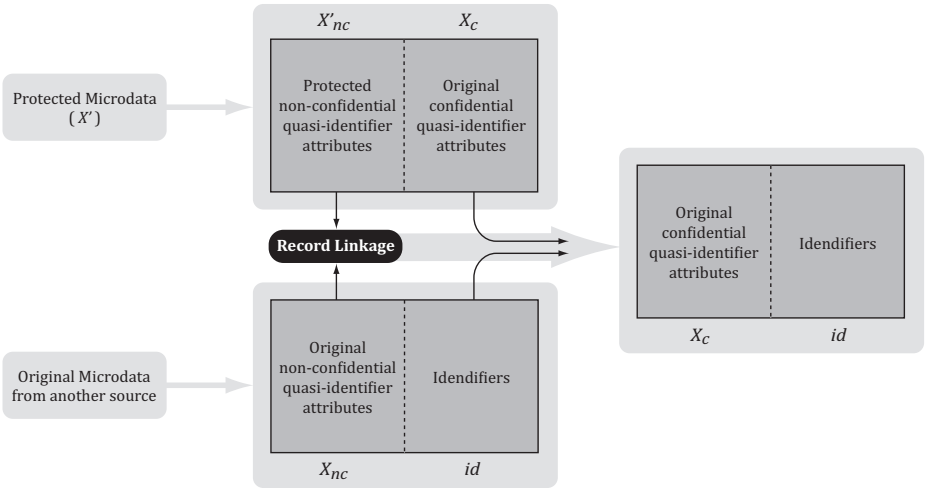

Fig. 2. Disclosure Risk Scenario.

In order to preserve information loss as low as possible, microaggregation methods try to minimize the total sum of distances between all the elements to be protected and the centroid of the cluster where an element belongs to, *i.e* minimize the total Sum of Square Errors (SSE):

$$SSE = \sum_{i=1}^{c} \sum_{x_{ij} \in C_i} (x_{ij} - \bar{x}_i)^T (x_{ij} - \bar{x}_i),$$

where c is the total number of clusters, C_i is the i-th cluster and $\bar{x}_i$ is the centroid of C_i. The restriction is $|C_i| \geq k$, for all $i = 1, \ldots, c$. In general, the larger value of k the lower the disclosure risk. Therefore, in order to parametrize microaggregation methods, k has to be as large as possible without compromising the statistical utility of the protected information.

The rational of this process is to make the protected data as similar as possible to the original one. In any case, methods should provide clusters with at least k elements. As introduced before, finding the optimal multivariate microaggregation has been proven to be an NP-Hard problem. For this reason, heuristic methods have been proposed in the literature.

For our work, we will use some of these different algorithms proposed for microaggregation in order to compare them to our new microaggregation technique. In this section, we explain a deterministic and optimal algorithm for univariate microaggregation, which is also used by two other methods for projection based multivariate microaggregation: PCP microaggregation and Zscores microaggregation. Finally, we describe one of the most used methods for heuristic microaggregation (specially for the multivariate case, although it can be applied to the

univariate case as well): the MDAV (Maximum Distance to Average Vector) algorithm.

Optimal Univariate Microaggregation Although multivariate microaggregation has been proven to be a very complex problem, several polynomial approaches for the optimal univariate microaggregation as [11] may be found in the literature. In [7], the authors present two relevant conclusions for the optimal univariate microaggregation:

1. When elements are sorted according to an attribute, for any optimal partition, elements in each cluster are contiguous (non overlapping clusters exist)
2. All the clusters of any optimal partition contain between k and $2k-1$ elements.

Based on these two results, in [11] authors define an optimal univariate microaggregation as follows. Let $A = (v_1 \dots v_n)$ be a vector of size n containing all the values for the attribute being protected. The values are sorted in ascending order so that if $i < j$ then $v_i \leq v_j$, where v_1 is the smallest element and v_n is the largest element in A. Let k be an integer such that $1 \leq k < n$ (k is directly obtained from the microaggregation configuration).

Given A and k, a graph $G_{k,n}$ is defined as follows. Firstly, we define the nodes of G as the elements v_i in A plus one additional node g_0 (this node is later needed to apply the Dijkstra algorithm). Then, for each node g_i, we add to the graph the directed edges (g_i, g_j) for all j such that $i + k \leq j < i + 2k$. The edge (g_i, g_j) means that the values $(v_i, \dots, v_j)$ might define one of the possible clusters. Then, the cost of the edge (g_i, g_j) is defined as the within-group sum of squared error for such cluster. That is, $SSE = \Sigma_{l=i}^{j}(v_l - \bar{v})^2$, where $\bar{v}$ is the average record of the cluster.

Given this graph, the optimal univariate microaggregation is defined by the shortest path algorithm between the nodes g_0 and g_n. This shortest path can be computed using the Dijkstra algorithm. Thus, the optimal clustering can be computed in linear time.

Projection Based Microaggregation The basic idea of Projection Based Microaggregation methods is to approximately reduce the multivariate microaggregation problem into the univariate case, by projecting $n_{att} > 1$ attributes (corresponding to some attributes of the records) into a single one.

The use of this projection techniques is motivated by the difficulty of sorting multivariate data that arises when one tries to extend the optimal univariate solution to the case of multivariate microaggregation.

Ideally, the employed projection should maintain the global statistical properties of the initial (non-projected) values. With this goal in mind, two projection methods seem particularly suitable: the principal component projection [12] and the sum of Z-scores [13].

Projected multivariate microaggregation is described in Algorithm 1, when applied to a microdata file X with n records and n_{attr} attributes.

Algorithm 1: Projected Microaggregation

Data: X: original microdata, k: integer
Result: X': protected microdata
begin
  Split the microdata file X into r sub-files $\{X_i\}_{1 \leq i \leq r}$, each one with v_i attributes of the n records, such that $\sum_{i=1}^{r} v_i = V$;
  foreach $(X_i \in X)$ **do**
    Apply a projection algorithm to the attributes in X_i, which results in an univariate vector z_i with n components (one for each record) ;
    Sort the components of z_i in increasing order;
    Apply to the sorted vector z_i the following variant of the univariate optimal microaggregation method explained in Section 2.2: use the algorithm defining the cost of the edges $\langle z_{i,s}, z_{i,t} \rangle$, with $s < t$, as the within-group sum of square error for the v_i-dimensional cluster in X_i which contains the original attributes of the records whose projected values are in the set $\{z_{i,s}, z_{i,s+1}, \ldots, z_{i,t}\}$;
    For each cluster resulting from the previous step, compute the v_i-dimensional centroid and replace all the records in the cluster by the centroid ;

Ideally, the employed projection should maintain the global statistical properties of the initial (non-projected) values. With this goal in mind, two projection methods seem particularly suitable: the principal component projection (PCP) [12] and the sum of Z-scores [13]. PCP is a projection technique that preserves the variance of the multivariate data set as much as possible in the projected data set in order to simplify the complexity of the data set, while preserving the statistical utility of the projected data. On the other hand, Z-score is a dimensionless quantity derived by subtracting the mean of each attribute from a single value and then dividing the difference by the standard deviation of that attribute. The resulting microaggregation algorithms obtained after the application of these two projection methods, called *PCP microaggregation* and *Zscores microaggregation*, will be used in Section 4 to test the quality of our new technique. See [17] for more details.

MDAV Microaggregation The MDAV (Maximum Distance to Average Vector) algorithm [7, 14] is an heuristic algorithm for clustering records in a microdata file X so that each cluster is constrained to contain at least k records. This algorithm can be used for univariate microaggregation and multivariate microaggregation. The MDAV algorithm is described in Algorithm 2.

MDAV generic algorithm can be instantiated for different data types, using appropriate definitions for distance and average. Normally, *the most distant record* and the *closest records* are computed using the Euclidean distance, and *the average record* is defined as the arithmetic mean of the records. The aver-

Algorithm 2: MDAV

```
Data: X: original microdata, k: integer
Result: X': protected microdata
begin
    while (|X| > k) do
        Compute the average record x̄ of all records in X;
        Consider the most distant record x_r to the average record x̄;
        Form a cluster around x_r. The cluster contains x_r together with the
        k − 1 closest records to x_r;
        Remove these records from microdata file X;
        if (|X| > k) then
            Find the most distant record x_s from record x_r;
            Form a cluster around x_s. The cluster contains x_s together with the
            k − 1 closest records to x_s;
            Remove these records from microdata file X;
        Form a cluster with the remaining records;
```

age record is used to replace the original records when building the protected microdata file.

2.3 Measures to Evaluate Risk and Utility

As we discussed before, a microdata protection method must guarantee a certain level of privacy (low disclosure risk). At the same time, since the goal is to allow third parties to perform reliable statistical computations over the released (protected) data, the protection method must ensure that the protected data is statistically close enough to the original one.

Therefore, given a microdata protection method, we have two inversely related aspects to measure: the *disclosure risk* (DR), which is the risk that an intruder obtains correct links between the protected and the original data; and the *information loss* (IL) caused by the protection method. When one of them increases, the other one decreases. The two extreme cases are the following ones: (i) if the original microdata is released, then information loss is zero, but the disclosure risk is maximum; (ii) if the original microdata is encrypted and then released, the disclosure risk is (almost) zero, but the information loss is maximum.

There are different generic measures proposed in the literature to evaluate the quality of a data protection method. One approach was presented in [5, 6], where the authors combine both information loss and disclosure risk in a *score* using the arithmetic mean. This method is refined in subsequent works [19, 24].

In order to calculate the *score*, first we need to calculate some information loss and disclosure risk meausures:

- **Information Loss (IL):** Let X and X' be matrices representing the original and the protected microdata files, respectively. Let V and R be the

covariance matrix and the correlation matrix of X, respectively; let $\overline{X}$ be the vector of variable averages for X and let S be the diagonal of V. Define V', R', $\overline{X}'$, and S' analogously from X'. The information loss is computed by averaging the mean variations of $X - X'$,$V - V'$,$S - S'$, and the mean absolute error of $R - R'$ and multiplying the resulting average by 100.

- **Disclosure Risk (DR):** For this measure in the original paper, the authors assume two different scenarios in order to evaluate DR: (i) *Distance Linkage Disclosure risk* (DLD), which is the average percentage of linked records using distance based RL [21]. Note that, this scenario is the same described in Section 2.1, where the intruder has access to an external data source and he is interested in disclosure the identity of the individual and (ii) *Interval Disclosure risk* (ID) which is the average percentage of original values falling into the intervals around their corresponding masked values, in this scenario the intruder has no access to an external data source and he is interested to disclose the original value of a protected one. The two values are computed over the number of attributes that the intruder is assumed to know, in particular, in this paper we assume the scenario described in [19] where the intruder knows all the possible combinations from one to all the attributes. The Disclosure Risk is computed as $DR = 0.5 \cdot DLD + 0.5 \cdot ID$.
- **Score:** The final score measure is computed by weighting the presented measures and it was also proposed in [6]:

$$score = 0.5\, IL + 0.5\, DR$$

Note that the better a protection method, the lower its score.

Apart from this generic measure for protection method evaluation, we can find specific IL and DR measures for microaggregation in the literature. For instance, the total Sum of Square Error SSE is usually used for information loss evaluation, since it is the fitness function used by microaggregation to minimize the loss of statistical utility of the protected microdata file. In order to compute the DR, in [18], a specific DR measure is defined. The idea is to consider the ratio between the total number n of records and the number of protected records which are different. This gives the average size of each 'global cluster' in the protected microdata file. This measure was denoted as k', this *real anonymity* measure is computed as

$$k' = \frac{n}{|\{x'|x' \in X'\}|}$$

Since our work compares our new microaggregation algorithm with other classical microaggregation methods, we use both the specific measures presented above and the general *score* measure.

3 One Dimension Microaggregation

In this section, we present a new microaggregation method called *one dimension microaggregation* (Mic1D-k, for short). This method gathers all the values of

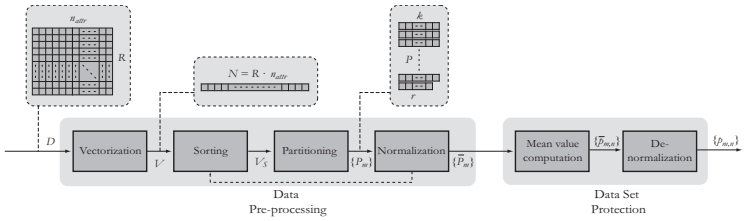

Fig. 3. Mic1D-k schema.

the microdata file into a single sorted vector, independently of the attribute they belong to. Then, it microaggregates all the mixed values together. The experiments presented here show that, by using real data, our proposal obtains lower disclosure risk than previous approaches whereas the information loss is preserved.

As shown in Figure 3, Mic1D-k is based on an important data pre-processing technique that must be applied before starting the protection process. This pre-processing phase is decomposed in several steps. Namely, vectorization, sorting, partitioning and normalization. Following, we go into further details about these steps.

Vectorization

The vectorization step gathers all the values from the microdata file in a single vector, independently on the attribute they belong to. Thereby, we ignore the attribute semantics and therefore the possible correlation between two different attributes in the microdata file. In other words, we *desemantize* the microdata file. Later, this process plays a central role in the discussion about the results achieved by Mic1D-k.

Formally speaking, let $\mathcal{D}$ be the original microdata file to be protected. We denote by R the number of records in $\mathcal{D}$. Each record consists of n_{att} numerical attributes. We assume that none of the records contains missing values. We denote by N the total number of values in $\mathcal{D}$. As a consequence, $N = R \cdot n_{att}$.

Let V be a vector of size N containing all the values in the microdata file. Mic1D-k treats values in the microdata file as if they were completely independent. In other words, the concept of record and attribute is ignored and the N values in the microdata file are placed in V.

The effect of this step on a certain microdata file is depicted in the upper half of Figure 3.

Sorting

Since the values in the vectorized microdata file belong to different source attributes, they present a pseudo-random aspect and it becomes very difficult to find the optimal partitions, *i.e.* partitions with SSE value as low as possible.

In order to simplify this search, the whole vector is sorted. This way, by the conclusions extracted from the univariate microaggregation presented in Subsection 2.2, optimal partitions are contiguous and, therefore, the partitioning process in this new vector can be done easily, as we will see later.

Formally, V is sorted increasingly. Let us call V_s the sorted vector of size N containing the sorted data and v_i the ith element of V_s, where $0 \leq i < N$.

Partitioning

Similarly to general microaggregation, in order to ensure a certain level of privacy (k-anonymity), Mic1D-k splits the vectorized microdata file in several k-partitions and it calculates the average value for each partition. By modifying the value of k, Mic1D-k allows us to adjust the trade-off between information loss (SSE) and disclosure risk. Note that if the vectorized microdata file was not sorted (previous step), k would not have this property.

Formally, V_s is divided into smaller sub-vectors or partitions. We define k where $1 < k \leq N$ as the number of values per partition. Note that if k is not a divisor of N, the last partition will contain a smaller number of values. Let P be the number of partitions containing k values. We call r the number of values in the last partition where $0 \leq r < k$. Therefore, $N = kP + r$. We will suppose that $r > 0$, so we have $P + 1$ partitions (note that $r > 0$ if and only if k does not divide N). We denote by P_m the mth partition.

Let $v_{m,n}$ be defined as the nth element of P_m:

$$\begin{cases} v_{m,n} := v_{mk+n} & n = 0 \dots k-1 \quad m = 0 \dots P-1 \\ v_{P,n} := v_{Pk+n} & n = 0 \dots r-1 \end{cases}$$

The upper half of Figure 3 shows the effect of this step on a certain microdata file.

Normalization

Since the range of the values in the different attributes could differ significantly among them, it is necessary to normalize the data to a certain predefined range of values.

There are many ways to normalize a microdata file. A possible solution would be to normalize each attribute independently before the application of the vectorization step. However, this normalization method could present problems with skewed attributes and therefore the attributes could not be merged in the sorting step. For this reason, we propose to normalize the data stored in each partition separately. Thereby, similar values are assigned to the same partition and therefore the chances to avoid the effect of skewness in the data are higher.

Formally, we denote the normalized values as $\bar{v}_{m,n}$ and the normalized partitions as $\bar{P}_m$. Let $\max_m$ and $\min_m$ be the maximum and the minimum values in the mth partition:

$$\max_m := \max_{0 \leq i < k} \{v_{m,i}\} \qquad \min_m := \min_{0 \leq i < k} \{v_{m,i}\}$$

The normalized values are then defined as:

$$\begin{cases} \bar{v}_{m,n} := \frac{v_{m,n} - \min_m}{\max_m - \min_m} & \text{if } \max_m \neq \min_m \\ \bar{v}_{m,n} := 0.5 & \text{if } \max_m = \min_m \end{cases}$$

where $0 \leq m < P$ (or $0 \leq m \leq P$ if k does not divide N,) and $0 \leq n < k$. Note that $\max_m = \min_m$ means that all the values in the partition are the same. In this case, the normalized value is centered in the normalization range.

Re-sorting and Re-normalization

One of the goals of the sorting process, apart from reducing the SSE value, is to desemantize the microdata file, *i.e.*, to merge values from different attributes in order to completely break their semantics and therefore make the re-identification process more difficult. If the range of values of a certain attribute differs significantly from the others, it is likely that it is not merged in previous steps.

Then, in order to appropriately mingle all attributes, once data has been sorted and normalized, we repeat these two steps (sorting and normalization). Since the range of values have been homogenized by normalization, attributes are conveniently mixed in the second sorting step and thus the microdata file is correctly preprocessed.

Mean Value Computation

Once data is preprocessed, for each partition $\bar{P}_m$, the mean value of its components is computed:

$$\mu_m = \sum_{n=0}^{k-1} \frac{\bar{v}_{m,n}}{k} \qquad m = 0 \dots P-1 \qquad \mu_P = \sum_{n=0}^{r-1} \frac{\bar{v}_{P,n}}{r}$$

where the latter expression is applied to the last partition if $r > 0$, *i.e.*, if k does not divide the total number of values in the microdata file.

The protected value $\bar{p}_{m,n}$ for $\bar{v}_{m,n}$ is then:

$$\begin{cases} \bar{p}_{m,n} = \mu_m & n = 0 \dots k-1 \quad m = 0 \dots P-1 \\ \bar{p}_{P,n} = \mu_P & n = 0 \dots r-1 \end{cases}$$

Finally, Mic1D-k denormalizes the data into the original range, according to the normalization and re-normalization steps in the previous block. Then, the protected values are placed in the protected microdata file in the same place occupied by the corresponding $v_{m,n}$ in the original microdata file. In this way, we are undoing the sorting and vectorization steps.

4 Experiments

We have tested Mic1D-k with real data extracted from two microdata files available in the Internet. The first one, denoted as Water-treatment, was extracted from the UCI repository [15], and it has already been used in other works dealing with disclosure risk evaluation, e.g. [16]. It contains 35 attributes corresponding to 380 entries or records. The second microdata file, called Census, was extracted using the Data Extraction System [27] of the U.S. Census Bureau, and it has been used as a reference database in many works dealing with statistical data protection, e.g. [4, 6, 17]. It contains 1080 records and 13 attributes.

As we will see later on, Mic1D-k achieves lower disclosure risk using real data than other multivariate microaggregation methods, such as MDAV, whereas it preserves a lower information loss.

4.1 Attribute Selection

To apply multivariate microaggregation to a microdata file X, we need to choose among the different microaggregation methods, the parameter k, and the number of blocks the microdata file X is split into. However, there are other parameters to be considered when the number of blocks B is larger than 1. As it was explained in [18], the way in which the attributes are grouped into blocks affects significantly the results and the quality of multivariate microaggregation.

It is a standard practice in statistical agencies to select the attributes on the basis of statistical utility. It is clear that, if the considered attributes are highly correlated, two records which are similar with respect to one attribute, will be also similar with respect to another one. Due to this, if microaggregation is applied to correlated attributes, when two values of the same attribute coming from different records are close, each pair of attributes coming from the remaining attributes in the cluster will be also close. Then, the intra-cluster distance is short and the information loss is low.

Nevertheless, as usual, statistical utility and privacy are inversely related. Therefore, the disclosure risk of microaggregation in this case is higher than in the case where correlated attributes are put into different blocks. Then as pointed out in [18], it is possible to group the attributes in a different way: blocks are formed in such a way that the first attributes of all blocks are (highly) correlated, the second attributes of all blocks are also (highly) correlated, and so on. This way, we are making blocks correlated, instead of constructing blocks with correlated attributes. The goal of this approach is to increase the resulting real anonymity k'. If two records A and B are in the same cluster for some blocks, this means that the first attribute values of these records are more or less close to each other, and the same for the second attribute of the block, etc. Then, when we consider another block, if the j-th attribute of this new block is (highly) correlated with the j-th attribute of the latter block, records A and B will probably be close to each other as well, with respect to the attributes in the second block. Therefore, with some non-negligible probability, A and B will fall in the same cluster, again. Ideally, some records will fall inside the same clusters,

for each block of attributes, and so the number of protected records which will be exactly equal will be higher, increasing in this way the real anonymity and the privacy level of the released data.

4.2 Algorithms Parameterization

We have tested Mic1D-k and compared our results with those obtained by the projected microaggregation (PCP and Zscores) and MDAV microaggregation, using the Census and Water Treatment microdata files. As we explained above, when protecting a microdata file using multivariate microaggregation, the way in which the data is split to form blocks is highly relevant with regard to the degree of privacy achieved (k' value). For this reason, we have reduced both microdata files to have 9 attributes, which we detail in Tables 1 and 2.

id	Name	Description
$a1$	PH-E	Input pH to plant
$a2$	PH-P	Input pH to primary settler
$a3$	PH-D	Input pH to secondary settler
$a4$	DQO-E	Input chemical demand of oxygen to plant
$a5$	COND-P	Input conductivity to primary settler
$a6$	COND-D	Input conductivity to secondary settler
$a7$	DBO-S	Output biological demand of oxygen
$a8$	SS-S	Output suspended solids
$a9$	SED-S	Output sediments

Table 1. Attribute description of the Water-treatment microdata file.

id	Name	Description
$a1$	AGI	Adjusted gross income
$a2$	FICA	Social security retirement payroll deduction
$a3$	INTVAL	Amount of interest income
$a4$	EMCONTRB	Employer contribution for health insurance
$a5$	TAXINC	Taxable income amount
$a6$	WSALVAL	Amount: Total wage and salary
$a7$	ERNVAL	Business or farm net earnings in 19
$a8$	PEARNVAL	Total person earnings
$a9$	POTHVAL	Total other persons income

Table 2. Attribute description of the Census microdata file

In both files, attributes $a1, a2$ and $a3$ are highly correlated as well as attributes $a4, a5$ and $a6$ and attributes $a7, a8$ and $a9$. On the contrary, attributes

in different blocks (e.g. $a1$ and $a4$) are non-correlated. For our experiments, when protecting data, we assume attributes to be split into three blocks of three attributes each. Also, we consider two situations when protecting the microdata files: blocking correlated attributes and, therefore, having non-correlated blocks (low information loss and low disclosure risk), *i.e.*, $(a1, a2, a3)$, $(a4, a5, a6)$ and $(a7, a8, a9)$; and blocking non-correlated attributes but correlated blocks, *i.e.*, $(a1, a4, a7)$, $(a2, a5, a8)$ and $(a3, a6, a9)$.

For each microdata file and attribute selection method, we apply all microaggregation methods using different configurations (*i.e.* different values of k). The selection of these values aims at covering a wide range of SSE values and, thus, studying scenarios with different *information loss* values. Namely, we protect the microdata files with parameter $k = 5$, 25, 50 for the Census microdata file, and $k = 5$, 15, 25 for the Water-treatment microdata file.

Correlated	1G	$(a1, a2, a3)$, $(a4, a5, a6)$, $(a7, a8, a9)$
	2G	$(a1, a2, a5)$, $(a1, a3, a7)$, $(a2, a3, a6)$, $(a1, a4, a5)$, $(a2, a4, a6)$ $(a5, a6, a9)$, $(a6, a7, a8)$, $(a1, a8, a9)$, $(a2, a7, a9)$
	3G	$(a1, a4, a7)$, $(a1, a5, a8)$, $(a1, a6, a9)$, $(a2, a4, a7)$, $(a2, a5, a8)$ $(a2, a6, a9)$, $(a3, a4, a7)$, $(a3, a5, a8)$, $(a3, a6, a9)$
Non-correlated	1G	$(a1, a4, a7)$, $(a2, a5, a8)$, $(a3, a6, a9)$
	2G	$(a1, a4, a5)$, $(a1, a3, a7)$, $(a4, a7, a8)$, $(a1, a2, a5)$, $(a2, a4, a8)$ $(a5, a8, a9)$, $(a3, a6, a8)$, $(a1, a6, a9)$, $(a3, a4, a9)$
	3G	$(a1, a2, a3)$, $(a1, a5, a6)$, $(a1, a8, a9)$, $(a2, a3, a4)$, $(a4, a5, a6)$ $(a4, a8, a9)$, $(a2, a3, a7)$, $(a5, a6, a7)$, $(a7, a8, a9)$

Table 3. Different groups of attributes known by the intruder.

For Mic1D-k, we use $k = 3000$, 4000, 5000 for the Census microdata file and $k = 500$, 800, 900 for the Water Treatment microdata file. Note that, since Mic1D-k *desemantizes* the microdata file, it does not make sense to consider different situations related to the correlation of the attributes and, therefore, we protect the data just once for each parametrization. In order to make a fair comparison, we have chosen the values of k in Mic1D-k to obtain similar SSE values to those obtained by MDAV after protecting the microdata files.

4.3 Algorithms Comparison

In order to compare the disclosure risk of microaggregation methods, we consider that a possible intruder knows the values of three random attributes of the original microdata file. Different tests are performed assuming that the intruder knows different sets of three attributes. Depending on these attributes, by using multivariate microaggregation, the intruder will have information coming from one or more groups. Table 3 shows all the considered possibilities.

	k	SSE	k'		
			1G	2G	3G
MDAV-k	5	64.99	5.00	1.92	1.00
	25	223.73	25.12	7.00	1.09
	50	328.31	51.43	14.66	1.41
PCP-k	5	131.05	5.06	1.91	1.00
	25	320.76	25.12	6.72	1.02
	50	441.66	51.43	13.97	1.15
Zscores-k	5	66.62	5.05	1.91	1.00
	25	159.95	25.12	6.80	1.04
	50	243.81	51.43	14.23	1.30
Mic1D-k	3000	32.27	8.37	9.87	5.77
	4000	129.06	20.10	22.09	13.89
	5000	738.12	72.83	76.08	55.02

Correlated attributes

	k	SSE	k'		
			1G	2G	3G
MDAV-k	5	58.49	5.00	1.96	1.02
	25	260.13	25.12	7.35	1.24
	50	356.47	51.43	15.86	2.05
PCP-k	5	124.99	5.03	1.91	1.00
	25	251.53	25.12	6.74	1.03
	50	382.69	51.43	14.00	1.22
Zscores-k	5	121.68	5.04	1.93	1.00
	25	242.26	25.12	6.97	1.07
	50	354.83	51.43	14.86	1.45
Mic1D-k	3000	32.27	5.63	8.51	8.04
	4000	129.06	13.53	19.45	19.19
	5000	738.12	59.77	67.77	67.25

Non-correlated attributes

Table 4. SSE and real k' of different microaggregation methods and parameterizations using the Census microdata file. Method-k corresponds to microaggregation using method Method (MDAV, PCP or Zscores) with initial anonymity value k.

	k	SSE	k'		
			1G	2G	3G
MDAV-k	5	28.18	5.09	1.94	1.00
	15	72.03	15.20	4.42	1.01
	25	114.56	25.33	7.28	1.10
PCP-k	5	28.59	5.14	1.94	1.01
	15	71.99	15.20	4.41	1.02
	25	110.91	25.33	7.24	1.04
Zscores-k	5	23.78	5.14	1.94	1.01
	15	72.23	15.20	4.43	1.03
	25	111.69	25.33	7.23	1.07
Mic1D-k	500	65.89	3.25	3.39	1.76
	800	80.95	7.87	7.55	4.67
	900	255.64	12.95	13.61	9.14

Correlated attributes

	k	SSE	k'		
			1G	2G	3G
MDAV-k	5	69.51	5.00	2.03	1.03
	15	173.96	15.20	5.28	1.39
	25	259.07	25.33	9.22	1.91
PCP-k	5	93.67	5.02	1.94	1.01
	15	170.12	15.20	4.47	1.03
	25	229.50	25.33	7.33	1.13
Zscores-k	5	73.52	5.02	1.97	1.02
	15	160.30	15.20	4.75	1.10
	25	231.81	25.33	8.21	1.46
Mic1D-k	500	65.89	2.78	2.58	2.63
	800	80.95	4.74	7.17	6.88
	900	255.64	9.07	14.52	11.71

Non-correlated attributes

Table 5. SSE and real k' of different microaggregation methods and parameterizations using the Water Treatment microdata file.Method-k corresponds to microaggregation using method Method (MDAV, PCP or Zscores) with initial anonymity value k.

Firstly, we suppose that the three known attributes belong to the same microaggregated block (*e.g.* $(a1, a2, a3)$ in the correlated scenario or $(a1, a4, a7)$ in the non-correlated). Since the size of the three microagreggation blocks is 3, there are only three options to consider. We denote this case by 1G. Since the intruder has only access to data from one group, multivariate microaggregation ensures the k-anonymity property (this is the best possible scenario for multivariate microaggregation). However, note that, usually, the intruder cannot choose the attributes obtained from external sources and it might be difficult to obtain all the attributes for the same group. Secondly, we assume that the known attributes belong to two different microaggregated groups. There are many possible combinations of three attributes under this assumption, so nine of them were chosen randomly. We refer to this case as 2G. Finally, case 3G is defined analogously to 2G, and also nine possibilities of known attributes are considered. Note that, in both scenarios 2G and 3G, k-anonymity is not ensured by multivariate microaggregation. Note also that, if the intruder had more than three attributes, it would not be possible to consider 1G. We are considering the case were the intruder only has three attributes to study a scenario were multivariate microaggregation can still preserve k-anonymity.

The second column of Tables 4 and 5 presents the SSE values for all the parameterizations and situations described before. Note that the range of SSE covered by all the methods is similar. This allows us to compare the disclosure risk of all the algorithms fairly. For all these scenarios, we compute k' and the mean of all the k' values in each situation is presented in the third, fourth and fifth columns. Note that, whereas multivariate microaggregation is affected by the fact that the chosen attributes are correlated or not, this effect is not noticeable using Mic1D-k. Specifically, when the attributes in a group are not correlated, the information loss (SSE) using multivariate microaggregation tends to be increased since we are trying to collapse the records in a single value, using three independent attributes or dimensions. For instance, as it is illustrated in the first row of Table 5 (MDAV microaggregation with k equal to 5), the SSE value increases from 28.18 to 69.51 when the blocks are made using non correlated attributes . Nevertheless, this effect can be neglected with Mic1D-k since, thanks to the data preprocessing, the whole microaggregation process is performed on a single dimension (vector of values), the semantics of attributes are ignored and the effect caused by attribute correlations is avoided. For this reason, in Tables 4 and 5 SSE values are identical for the correlated and non correlated attribute blocking.

The results described in Tables 4 and 5 also show that, Mic1D-k achieves lower disclosure risk levels (larger values of k') than those achieved by multivariate microaggregation for similar information loss (SSE), especially when the attributes chosen come from different microaggregated groups (2G and 3G), which is the most common case. For instance, if we observe the k' values of MDAV microaggregation using the most 'private' configuration (k equal to 25 and using non correlated attributes) we can see that the resulting k' values where the intruder has access to attributes coming from more than one group (G2 equal to

9.22 and G3 equal to 1.91) are lower than using Mic1D-k with similar SSE value (G2 equal to 14.52 and G3 equal to 11.71). Note also that, when the intruder has access to the three attributes coming from a single microaggregated group, multivariate microaggregation configurations present k' values which are similar or, in some cases, even larger than those obtained by Mic1D-k (comparing cases with similar SSE). This is normal since such methods preserve the k-anonymity in this case. However, in the remaining scenarios (2G and 3G), that represent most of the cases, Mic1D-k achieves larger k' values than those obtained by multivariate microaggregation when similar SSE values are compared.

4.4 Method comparison using generic measures

We have repeated the experiments presented in [6] where a large variety of protection methods were compared using the Census microdata file based on the *score*, presented in Subsection 2.3, to measure the results. We have computed the disclosure risk considering different scenarios ranging from the extreme case where the intruder knows only one attribute, to the opposite case where it knows all the attributes, as in [19]. Specifically, we have considered 512 different sets of attributes for each Mic1D-k and multivariate microaggregation parameterization. The total number of executions run in these experiments is 10752.

	k	IL	DLD	ID	Score
MDAV-k	5	31.67	34.11	69.70	41.79
	25	51.14	11.70	54.65	42.16
	50	60.19	5.68	47.75	43.45
PCP-k	5	63.18	4.35	47.63	44.59
	25	57.31	2.23	38.53	38.85
	50	57.86	1.81	34.41	37.98
Zscores-k	5	59.64	12.13	57.71	47.28
	25	79.90	6.96	52.86	54.90
	50	89.33	6.25	50.09	58.75
Mic1D-k	3000	22.17	35.95	54.16	33.61
	4000	57.31	16.91	48.53	45.02
	5000	82.44	4.94	32.22	50.51

Correlated attributes

	k	IL	DLD	ID	Score
MDAV-k	5	41.47	23.14	63.03	42.28
	25	51.47	6.56	49.07	39.64
	50	44.11	3.26	43.87	33.84
PCP-k	5	55.70	3.75	42.71	39.47
	25	78.39	2.02	34.81	48.40
	50	83.30	1.50	32.29	50.10
Zscores-k	5	51.76	4.80	51.66	40.00
	25	86.58	2.37	47.19	55.68
	50	90.81	2.43	43.80	56.96
Mic1D-k	3000	22.17	35.95	54.16	33.61
	4000	57.31	16.91	48.53	45.02
	5000	82.44	4.94	32.22	50.51

Non-correlated attributes

Table 6. Score k' of different microaggregation methods and parameterizations using the Census microdata file. Method-k corresponds to microaggregation using method Method (MDAV, PCP or Zscores) with initial anonymity value k.

The first main conclusion extracted from the results presented in Table 6 is that the quality obtained by Mic1D-k is orthogonal to the degree of correlation between the attributes in a cluster. On the contrary, this correlation has a significant effect on the remaining techniques. For example, the best score for

MDAV using correlated attributes is 41.79 while it is 33.84 using non-correlated attributes. In this experiments we test two extreme cases where all the attributes are correlated or none of them. In a real scenario, we would not be able to choose the attributes to be protected, and, as a consequence, we do not have any control on the correlation between them making the application of these multivariate microaggregation techniques less suitable than our proposal.

Also, the configuration process is simplified for Mic1D-k. While the other multivariate microaggregation methods must choose the best attribute blocking selection for clustering, which may be as difficult as the anonymization problem itself, we avoid this problem by replacing the attribute selection phase by a significantly less complex pre-processing phase.

Finally, Mic1D-k obtains the lowest score when k is equal to 3000 (33.61). MDAV algorithm also obtains low scores when non-correlated attributes are grouped together (k equal to 50, 33.84). However, as we have said before, using MDAV one has to decide which attributes are to be grouped together and this is not a straightforward decision.

5 Conclusions and future work

In this paper, we have presented a new type of microaggregation called *One Dimension microaggregation*. This microaggregation method significantly diminish the problem of attribute selection in multivariate microaggregation achieving in general a higher level of privacy than that obtained by three of the most well-known microaggregation algorithms. This is specially true as, from the attributes known by the intruder, the number of these coming from different microaggregation groups of multivariate microaggregation increases.

As future work, we plan to develop and implement a method for vector partitioning which considers the SSE value when the partitions are done so that we can reduce the SSE value of our method and, therefore, the information loss.

All in all, in this paper we show that microaggregation is a very useful method for the anonymization of complex records containing a large number of attributes, when it is combined with the data preprocessing proposed in our work.

Acknowledgments

This work is partially supported by the Ministry of Science and Technology of Spain under contract TIN2012-34557 and by the BSC-CNS Severo Ochoa program (SEV-2011-00067)

References

1. Adam, N. R., Wortmann, J. C.,(1989), Security-control for statistical databases: a comparative study, ACM Computing Surveys, Volume: 21, 515-556.

2. Aggarwal, C. (2005) On k-anonymity and the curse of dimensionality, Proceedings of the 31st International Conference on Very Large Databases, pages 901-909.
3. Aggarwal, G., Feder T., Kenthapadi, K., Khuller, S., Panigrahy, R., Thomas, D., Zhu, A., (2006) Achieving anonymity via clustering, Proceedings of the 25th ACM Symposium on Principles of Databases Systems, pages 153-162.
4. CASC: Computational Aspects of Statistical Confidentiality, European Project IST-2000-25069, http://neon.vb.cbs.nl/casc.
5. Domingo-Ferrer, J., Torra, V., (2001), Disclosure control methods and information loss for microdata, Pages 91-110 of [8].
6. Domingo-Ferrer, J., Torra, V. (2001) A quantitative comparison of disclosure control methods for microdata, Pages 111-133 of [8].
7. Domingo-Ferrer, J., Mateo-Sanz, J. M. (2002) Practical data-oriented microaggregation for statistical disclosure control, IEEE Trans. on Knowledge and Data Engineering 14:1 189-201.
8. Doyle, P., Lane, J., Theeuwes, J., Zayatz, L., eds. (2001) Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies, Elsevier Science.
9. Felsö, F., Theeuwes, J., Wagner, G., (2001) Disclosure Limitation in Use: Results of a Survey, Pages 17-42 of [8].
10. Fung, B., Wang, K., Yu, P., (2005) Top-down specialization for information and privacy preservation, Proceedings of the 21st IEEE International Conference on Data Engineering, pages 205-216.
11. Hansen, S., Mukherjee, S. (2003) A Polynomial Algorithm for Optimal Univariate Microaggregation. Trans. on Kwnoledge and Data Engineering, 15:4 1043-1044.
12. Jolliffe, I.T., (2002) Principal Component Analysis, Springer Series in Statistics, Springer, ISBN 978-0-387-95442-4.
13. Larsen, R.J. and Marx, M. L., (2005) An Introduction to Mathematical Statistics and Its Applications, Third Edition, ISBN-10: 0131867938, Prentice Hall.
14. Mateo-Sanz, J.M., Domingo-Ferrer, J., A method for data-oriented multivariate microaggregation, Statistical Data Protection for Official Publications of the European Communities, pages 89-99.
15. Murphy, P., M., Aha, D. W., (1994), UCI Repository machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.
16. Nin, J., Torra, V., (2005), Empirical analysis of database privacy using twofold integrals, Lecture Notes in Artificial Intelligence, volume 3801 pages: 1 - 8.
17. J. Nin, J. Herranz and V. Torra, On the Disclosure Risk of Multivariate Microaggregation. *Data and Knowledge Engineering (DKE)*, Elsevier, in press.
18. Nin, J., Herranz, J., Torra V., (2008), How to group attributes in multivariate microaggregation, Int. J. on Uncertainty, Fuzziness and Knowledge-Based Systems, 16(1):121-138.
19. Nin, J, Herranz, J., Torra, V. (2008) Towards a More Realistic Disclosure Risk Assessment. In Privacy in Statistical Databases (PSD), Lecture Notes in Computer Science 5262, 152-165.
20. Oganian, A., Domingo-Ferrer, J. (2000) On the Complexity of Optimal Microaggregation for Statistical Disclosure Control, Statistical J. United Nations Economic Commission for Europe, 18, 4, 345-354.
21. Pagliuca, D., Seri, G., (1999), Some results of individual ranking method on the system of enterprise accounts annual survey, Esprit SDC Project, Delivrable MI-3/D2.

22. Samarati, P., Sweeney, L. (1998) Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, SRI Intl. Tech. Rep.
23. Sande, G. (2002) Exact and approximate methods for data directed microaggregation in one or more dimensions, Int. J. of Unc., Fuzz. and Knowledge Based Systems 10:5 459-476.
24. Sebé, F., Domingo-Ferrer J., Mateo-Sanz, J. M., Torra V., (2002) Post-Masking Optimization of the Tradeoff between Information Loss and Disclosure Risk in Masked Microdata Sets Inference Control in Statistical Databases, Lecture Notes in Computer Science volume:2316, 187-196.
25. Sweeney, L. (2002) Achieving k-anonymity privacy protection using generalization and suppression, Int. J. of Unc., Fuzz. and Knowledge Based Systems 10:5 571-588.
26. Sweeney, L. (2002) k-anonymity: a model for protecting privacy, Int. J. of Unc., Fuzz. and Knowledge Based Systems 10:5 557-570.
27. U.S. Census Bureau, Data Extraction System (1990) http://www.census.gov/.
28. Willenborg, L., Waal, T., (2001), Elements of Statistical Diclosure Control, Lecture Notes in Statistics, Springer.