Novel Iterative Min-Max Clustering to Minimize Information Loss in Statistical Disclosure Control

Abdun Naser Mahmood[1], Md Enamul Kabir[2], and Abdul K Mustafa[3]

[1]School of Engineering and Information Technology,
University of New South Wales Australian Defence Force Academy
Canberra, ACT 2600, Australia, [2]School of Human Movement Studies, University of Queensland, St Lucia, QLD 4072, Australia and [3]Humber College, School of Applied Technology, North Campus, Toronto, Canada
Abdun.Mahmood@unsw.edu.au,e.kabir@uq.edu.auabdul.mustafa@humber.ca

Abstract. In recent years, there has been an alarming increase of online identity theft and attacks using personally identifiable information. The goal of privacy preservation is to de-associate individuals from sensitive or microdata information. Microaggregation techniques seeks to protect microdata in such a way that can be published and mined without providing any private information that can be linked to specific individuals. Microaggregation works by partitioning the microdata into groups of at least k records and then replacing the records in each group with the centroid of the group. An optimal microaggregation method must minimize the information loss resulting from this replacement process. The challenge is how to minimize the information loss during the microaggregation process. This paper presents a new microaggregation technique for Statistical Disclosure Control (SDC). It consists of two stages. In the first stage, the algorithm sorts all the records in the data set in a particular way to ensure that during microaggregation very dissimilar observations are never entered into the same cluster. In the second stage an optimal microaggregation method is used to create k-anonymous clusters while minimizing the information loss. It works by taking the sorted data and simultaneously creating two distant clusters using the two extreme sorted values as seeds for the clusters. The performance of the proposed technique is compared against the most recent microaggregation methods. Experimental results using benchmark datasets show that the proposed algorithm has the lowest information loss compared with a basket of techniques in the literature.

Key words: Privacy; Microaggregation; Microdata protection; k-anonymity; Disclosure control;

1 Introduction

In recent years, the phenomenal advance of technological developments in information technology enable government agencies and corporations to accumulate

an enormous amount of personal data for analytical purposes. These agencies and organizations often need to release individual records (microdata) for research and other public benefit purposes. This propagation has to be in accordance with laws and regulations to avoid the propagation of confidential information. In other words, microdata should be published in such a way that preserve the privacy of the individuals. Microdata protection in statistical databases has recently become a major societal concern and has been intensively studied in recent years. Microaggregation for Statistical Disclosure Control (SDC) is a family of methods to protect microdata from individual identification. SDC seeks to protect microdata in such a way that can be published and mined without providing any private information that can be linked to specific individuals. SDC is often applied to statistical databases before they are released for public use.

To protect personal data from individual identification, SDC is often applied before the data are released for analysis [2, 25]. The purpose of microdata SDC is to alter the original microdata in such a way that the statistical analysis from the original data and the modified data are similar and the disclosure risk of identification is low. As SDC requires suppressing or altering the original data, the quality of data and the analysis results can be damaged. Hence, SDC methods must find a balance between data utility and personal confidentiality.

Various methods for Microaggregation has been proposed in the literature for protecting microdata [3, 4, 7, 8, 11, 12, 20, 22]. The basic idea of microaggregation is to partition a dataset into mutually exclusive groups of at least k records prior to publication, and then publish the centroid over each group instead of individual records. The resulting anonymized dataset satisfies k-anonymity [18], requiring each record in a dataset to be identical to at least $(k-1)$ other records in the same dataset. As releasing microdata about individuals poses privacy threat due to the privacy-related attributes, called quasi-identifiers, both k-anonymity and microaggregation only consider the quasi-identifiers. Microaggregation is traditionally restricted to numeric attributes in order to calculate the centroid of records, but also has been extended to handle categorical and ordinal attributes [4, 8, 19]. In this paper we propose a microaggregated method that is also applicable to numeric attributes.

The effectiveness of a microaggregation method is measured by calculating its information loss. A lower information loss implies that the anonymized dataset is less distorted from the original dataset, and thus provides better data quality for analysis. k- anonymity [17, 18, 21] provides sufficient protection of personal confidentiality of microdata, while ensuring the quality of the anonymized dataset, an effective microaggregation method should incur as little information loss as possible. In order to be useful in practice, the dataset should keep as much informative as possible. Hence, it is necessary to seriously consider the tradeoff between privacy and information loss. To minimize the information loss due to microaggregation, all records are partitioned into several groups such that each group contains at least k similar records, and then the records in each group are replaced by their corresponding mean such that the values of each variable are the same. Such similar groups are known as clusters. In the context of data

mining, clustering is a useful technique that partitions records into groups such that records within a group are similar to each other, while records in different groups are most distinct from one another. Thus, microaggregation can be seen as a clustering problem with constraints on the size of the clusters.

Many microaggregation methods derive from traditional clustering algorithms. For example, Domingo-Ferrer and Mateo-Sanz [3] proposed univariate and multivariate k-Ward algorithms that extend the agglomerative hierarchical clustering method of Ward et al. [23]. Domingo-Ferrer and Torra [6, 7] proposed a microaggregation method based on the fuzzy c-means algorithm [1], and Laszlo and Mukherjee [13] extended the standard minimum spanning tree partitioning algorithm for microaggregation [26]. All of these microaggregation methods build all clusters gradually but simultaneously. There are some other methods for microaggregation that have been proposed in the literature that build one/two cluster(s) at a time. Notable examples include Maximum Distance [15], Diameter-based Fixed-Size microaggregation and centroid-based Fixed-size microaggregation [13], Maximum Distance to Average Vector (MDAV) [8], MHM [9] and the Two Fixed Reference Points method [27]. Most recently, Lin *et al.* [28] proposed a density-based microaggregation method that forms clusters by the descending order of their densities, and then fine-tunes these clusters in reverse order.

The reminder of this paper is organized as follows. We introduce the problem of microaggregation in Section 2. Section 3 introduces the basic concept of microaggregation. Section 4 reviews previous microaggregation methods. We present a brief description of our proposed microaggregation method in Section 5. Section 6 shows experimental results of the proposed method. Finally, concluding remarks are included in Section 7.

2 Problem Statement

The algorithms for microaggregation works by partitioning the microdata into groups, where within groups the records are homogeneous but between groups the records are heterogeneous so that information loss is low. The similar groups are also called clusters. The level of privacy required is controlled by a security parameter k, the minimum number of records in a cluster. In essence, the parameter k specifies the maximum acceptable disclosure risk. Once a value for k has been selected by the data protector, the only job left is to maximize data utility. Maximizing utility can be achieved by microaggregating optimally, i.e. with minimum within-groups variability loss. So the main challenge in microaggregation is how to minimize the information loss during the clustering process. Although plenty of work has been done, to maximize the data utility by forming the clusters, this is not yet sufficient in terms of information loss. So more research needs to be done to form the clusters such that the information loss is as low as possible. This paper analyses the problem with a new multi-dimensional sorting algorithm such that the information loss is minimal.

Observing this challenge, this work presents a new clustering-based method for microaggregation, where a new multi-dimensional sorting algorithm is used in the first stage. In the second stage two distant clusters are made simultaneously in a systematic way. According to the second stage, sort all records in ascending order by using a sorting algorithm in the first stage explained in Section 5) so that the first record and the last record are most distant to each other. Form a cluster with the first record and its $(k-1)$ nearest records and another cluster with the last record and its $(k-1)$ nearest records. Sort the remaining records ($(n-2k)$, if dataset contains n records) by using the same sorting algorithm and continue to build pair clusters at the same time by using the first and the last record as seeds until some specified records remain. Finally form one/two cluster(s) depending on the remaining records. Thus all clusters produced in this way contain k records except the last cluster that may contain at the most $(2k-1)$ records. Performance of the proposed method is compared against the most recent widely used microaggregation methods. The experimental results show that the proposed microaggregation method outperforms the recent methods in the literature.

3 Background

Microdata protection through microaggregation has been intensively studied in recent years. Many techniques and methods have been proposed to deal with this problem. In this section we describe some fundamental concepts of microaggregation.

When we microaggregate data we should keep in mind two goals: data utility and preserving privacy of individuals. For preserving the data utility we should introduce as little noise as possible into the data and preserving privacy data should be sufficiently modified in such a way that it is difficult for an adversary to reidentify the corresponding individuals. Figure 1 shows an example of microaggregated data where the individuals in each cluster are replaced by the corresponding cluster mean. The figure shows that after aggregating the chosen elements, it is impossible to distinguish them, so that the probability of linking any respondent is inversely proportional to the number of aggregated elements.

Consider a microdata set T with p numeric attributes and n records, where each record is represented as a vector in a p-dimensional space. For a given positive integer $k \leq n$, a microaggregation method partitions T into g clusters, where each cluster contains at least k records (to satisfy k-anonymity), and then replaces the records in each cluster with the centroid of the cluster. Let n_i denote the number of records in the ith cluster, and $x_{ij}, 1 \leq j \leq n_i$, denote the jth record in the ith cluster. Then, $n_i \geq k$ for $i = 1$ to g, and $\sum_{i=1}^{g} n_i = n$. The centroid of the ith cluster, denoted by $\bar{x}_i$ is calculated as the average vector of all the records in the ith cluster.

In the same way, the centroid of T, denoted by $\bar{x}$, is the average vector of all the records in T. Information loss is used to quantify the amount of information of a dataset that is lost after applying a microaggregation method. In this paper

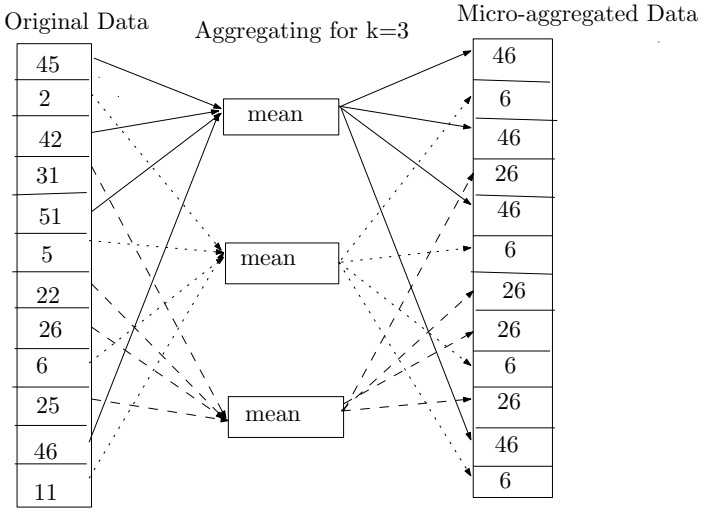

Fig. 1. Example of Microaggregation using mean

we use the most common definition of information loss by Domingo-Ferrer and Mateo-Sanz [3] as follows:

$$IL = \frac{SSE}{SST} \tag{1}$$

where SSE is the within-cluster squared error, calculated by summing the Euclidean distance of each record x_{ij} to the average value $\bar{x_i}$ as follows:

$$SSE = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \bar{x_i})^{'} (x_{ij} - \bar{x_i}) \tag{2}$$

and SST is the sum of squared error within the entire dataset T, calculated by summing the Euclidean distance of each record x_{ij} to the average value $\bar{x}$ as follows:

$$SST = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \bar{x})^{'} (x_{ij} - \bar{x}) \tag{3}$$

For a given dataset T, SST is fixed regardless of how T is partitioned. On the other hand, SSE varies of a dataset depending on the partition of the dataset. In essence, SSE measures the similarity of the records in a cluster. The lower the SSE, the higher the within-cluster homogeneity and the higher the SSE, the lower the within cluster homogeneity. If all the records in a cluster are the same, then the SSE is zero indicating no information is lost. On the other hand, if all the records in a cluster are more diverse, SSE is large indicating more information is lost. In this paper, we used SSE as a measure of similarity indicating a record will be included in a particular cluster if it causes least SSE among all other records in the dataset. Therefore, the microaggregation problem can be

enumerated as a constraint optimization problem as follows:

Definition 1 (Microaggregation problem) Given a dataset T of n elements and a positive integer k, find a partitioning $C = \{C_1, C_2, ..., C_c\}$ of T such that

1. $C_i \cap C_j = \Phi$, for all $i \neq j = 1, 2, ..., p$,
2. $\cup_{i=1}^{p} C_i = T$,
3. SSE is minimized,
4. for all $C_i \in T$, $\mid C_i \mid \geq k$ for any $C_i \in C$.

The microaggregation problem stated above can be solved in polynomial time for a univariate dataset [12] but has been shown to be NP hard for multivariate dataset [14]. It is a natural expectation that SSE is low if the number of clusters is large. Thus the number of records in each cluster should be kept close to k. Domingo-Ferrer and Mateo-Sanz [3] showed that no cluster should contain more than $(2k - 1)$ records since such clusters can always be partitioned to further reduce information loss.

4 Previous Microaggregation Methods

Previous microaggregation methods have been roughly divided into two categories, namely fixed-size and data-oriented microaggregation [3, 9]. For fixed-size microaggregation, the partition is done by dividing a dataset into clusters that have size k, except perhaps one cluster which has a size between k and $(2k-1)$, depending on the total number of records n and the anonymity parameter k. For the data-oriented microaggregation, the partition is done by allowing all clusters with sizes between k and $(2k-1)$. Intuitively, fixed-size methods reduce the search space, and thus are more computationally efficient than data-oriented methods [28]. However, data-oriented methods can adapt to different values of k and various data distributions and thus may achieve lower information loss than fixed-size methods.

Domingo-Ferrer and Mateo-Sanz [3] proposed a multivariate fixed-size microaggregation method, later called the Maximum Distance (MD) method [15]. The MD method repeatedly locates the two records that are most distant to each other, and forms two clusters with their respective $(k-1)$ nearest records until fewer than $2k$ records remain. If at least k records remain, it then forms a new cluster with all remaining records. Finally when there are fewer than k records not assigned to any cluster yet, this algorithm then individually assigns these records to their closest clusters. This method has a time complexity of $O(n^3)$ and works well for most datasets. Laszlo and Mukherjee [13] modified the last step of the MD method such that each remaining record is added to its own nearest cluster and proposed Diameter-based Fixed-size microaggregation. This method is however not a fixed size method because it allows more than one cluster to have more than k records.

The MDAV method is the most widely used microaggregation method [15]. MDAV is the same as MD except in the first step. MDAV finds the record r that is furthest from the current centroid of the dataset and the record s that is furthest from r instead of finding the two records that are most distant to each other, as is done in MD. Then form a cluster with r and its $(k-1)$ nearest records and form another cluster with s and its $(k-1)$ nearest records. For the remaining records, repeat this process until fewer than $2k$ records remain. If between k and $(2k-1)$ records remain, MDAV simply forms a new group with all of the remaining records. On the other hand, if the number of the remaining records is below k, it adds all of the remaining records to their nearest clusters. So MDAV is a fixed size method. Lin *et al.* [28] proposed a modified MDAV, called MDAV-1. The MDAV-1 is similar to MDAV except when the number of the remaining records is between k and $(2k-1)$, a new cluster is formed with the record that is the furthest from the centroid of the remaining records, and its $(k-1)$ nearest records. Any remaining records are then added to their respective nearest clusters. Experimental results indicate that MDAV-1 incurs slightly less information loss than MDAV [28]. Another variant of the MDAV method, called MDAV-generic, is proposed by Domingo-Ferrer and Torra [8], where by the threshold $2k$ is altered to $3k$. If between $2k$ and $(3k-1)$ records remain, then find the record r that is furthest from the centroid of the remaining records and form a cluster with r and its $(k-1)$ nearest records and another cluster with the remaining records. Finally when fewer than $2k$ records remain, this algorithm then forms a new cluster with all the remaining records. Laszlo and Mukherjee [13] proposed another method, called Centroid-based Fixed-size microaggregation that is also based on a centroid but builds only one cluster during each iteration. This algorithm first find a record r that is furthest from the current centroid of the dataset and then find a cluster with r and its $(k-1)$ nearest records. For the remaining records repeat the same process until fewer than k records remain. Finally add each remaining record to its nearest clusters. This method is not a fixed-size method as more than one cluster has more than k records. Solanas *et al.* [16] proposed a variable-size variant of MDAV, called V-MDAV. V-MDAV first builds a new cluster of k records and then tries to extend this to up to $(2k-1)$records based on some criteria. V-MDAV adopts a user-defined parameter to control the threshold of adding more records to a cluster. Chang *et al.* [27] proposed the Two Fixed Reference Points (TFRP) method to accelerate the clustering process of k-anonymization. During the first phase, TFRP selects two extreme points calculated from the dataset. Let N_{min} and N_{max} be the minimum and maximum values over all attributes in the datasets, respectively, then one reference point C_1 has N_{min} as its value for all attributes, and another reference point C_2 has N_{max} as its value for all attributes. A cluster of k records is then formed with the record r that is the furthest from C_1 and the $(k-1)$ nearest records to r. Similarly another cluster of k records is formed with the record s that is the furthest from C_2 and $(k-1)$ nearest records to s. These two steps are repeated until fewer than k records remain. Finally, these remaining records are assigned to their respective nearest clusters. This method

is quite efficient as C_1 and C_2 are fixed throughout the iterations. When all clusters are generated, TFRP applies a enhancement step to determine whether a cluster should be retained or decomposed and added to other clusters.

Lin *et al.* [28] proposed a density-based algorithm (DBA) for microaggregation. The DBA has two different scenarios. The first state of DBA (DBA-1) repeatedly builds a new cluster using the k-neighborhood of the record with the highest k-density among all records that are not yet assigned to any cluster until fewer than k unassigned records remain. These remaining records are then assigned to their respective nearest clusters. The DBA-1 partitions the dataset into some clusters, where each cluster contains no fewer than k records. The second state of DBA (DBA-2) attempts to fine-tune all clusters by checking whether to decompose a cluster and merge its content with other clusters. Notably, all clusters are checked during the DBA-2 by the reverse of the order that they were added to clusters in the DBA-1. After several clusters are removed and their records are added to their nearest clusters in the DBA-2, some clusters may contain more than $(2k-1)$ records. At the end of the DBA-2, the MDAV-1 algorithm is applied to each cluster with size above $(2k-1)$ to reduce the information loss. This state is finally called MDAV-2. Experimental results show that the DBA attains a reasonable dominance over the latest microaggregation methods.

All of the microaggregation methods described above repeatedly choose one/two records according to various heuristics and form one/two cluster(s) with the chosen records and their respective $(k-1)$ other records. However there are other microaggregation methods that build all clusters simultaneously and work by initially forming multiple clusters of records in the form of trees, where each tree represent a cluster. The multivariate k-Ward algorithm [3] first finds the two records that are furthest from each other in the dataset and build two clusters from these two records and their respective $(k-1)$ nearest records. Each of the remaining record then forms its own cluster. These clusters are repeatedly merged until all clusters have at least k records. Finally the algorithm is recursively applied to each cluster containing $2k$ or more records. Domingo-Ferrer *et al.* [10] proposed a multivariate microaggregation method called μ-Approx. This method first builds a forest and then decomposes the trees in the forest such that all trees have sizes between k and $\max(2k-1, 3k-5)$. Finally, for any tree with size greater than $(2k-1)$, find the node in the tree that is furthest from the centroid of the tree. Form a cluster with this node and its $(k-1)$ nearest records in the tree and form another cluster with the remaining records in the tree.

Hansen an Mukherjee [12] proposed a microaggregation method for univariate datasets called HM. After that Domingo-Ferrer *et al.* [9] proposed a multivariate version of the HM method, called MHM. This method first uses various heuristics, such as nearest point next (NPN), maximum distance (MD) or MDAV to order the multivariate records. Steps similar to the HM method are then applied to generate clusters based on this ordering. Domingo-Ferrer *et al.* [7] proposed a microaggregation method based on fuzzy c-means algorithm (FCM) [1]. This

method repeatedly runs FCM to adjust the two parameters of FCM (one is the number of clusters c and another is the exponent for the partition matrix m) until each cluster contains at least k records. The value of c is initially large (and m is small) and is gradually reduced (increased) during the repeated FCM runs to reduce the size of each cluster. The same process is then recursively applied to those clusters with $2k$ or more records.

5 The Proposed Approach

This section presents the proposed least information loss clustering algorithm based on minimum and maximum pairs of pairs of instances that minimizes the information loss and satisfies the k-anonymity requirement. It has been observed that the reason many of the existing techniques has high information loss is due to some clusters containing very $different$ observations which increases the information of a cluster. Therefore, the initial choice of cluster(s) is often difficult since these observations are not known in advance. The proposed technique solves this problem by creating the lower information loss cluster using the proposed Min-Max technique as explained in Section 5.1. Next, this process is incorporated in an iterative pairwise clustering algorithm that takes the minimum or maximum distant instances to create two clusters repeatedly by minimizing information loss and observing k-anonymity. The algorithm is described in Section 5.2.

5.1 Min distance and Max Distance

It has been observed that arbitrarily choosing cluster centroids (e.g., K-Means, MDAV, V-MDAV, MD, etc.) has its disadvantages. In particular, there is a possibility that the clustering process may include an outlier in a cluster in order to obey K-anonymity. However, this has the undesired effect of noticeably increasing the information loss. It has been shown [21] that by simultaneously building clusters whose centroids are farthest from the centroid of the dataset helps to improve the information loss. However, this technique still has drawbacks. For example, in some cases the two farthest points from the centroid of the dataset may fall in the same cluster, at other times they may fall in entirely different clusters, thus limiting the performance of the algorithms in these circumstances. This paper proposes a deterministic technique based on maximum and minimum distance points in the dataset in order to create clusters with lowest information loss in all cases. In order to achieve the lowest information loss, the algorithm iteratively chooses either two most distant points or two closest points in the dataset depending on which clustering would result in the lowest information loss. The Least MinMax distance based algorithm is described in the next section.

Table 1. Least Min-Max distance microaggregation algorithm

Input: a dataset T of n records and a positive integer k
Output: a partitioning $C = C_1, C_2, ..., C_c$ of T, where $c = |C|$
and $|C| \geq k$ for $i = 1$ to c

1. Let $C = \phi$, and $T' = T$;
2. Let Max_1, Max_2, and Min_1, Min_2 such that distance
$D(Max_1, Max_2) \leq D(i, j), \forall i, j \in 1, ..., n$;
3. Form a cluster C_1 containing first record Max_1 adn its $(k-1)$ nearest records in T';
and another cluster C_2 containing Max_2 record and its $(k-1)$ nearest records in T';
Let IL_{Max_1} IL_{max_2} represent the information loss calculated using equation 1
of clusters C_1 and C_2;
4. if $IL_{Max_1} \leq IL_{Max_2}$ then $LeastMaxCluster = C_1$ else $LeastMaxCluster = C_2$;
5. Repeat steps 3 and 4 by replacing Max_1 and Max_2 with Min_1 and Min_2 to create
$LeastMinCluster$;
6. Set $C = C \cup LeastMaxCluster \cup LeastMinCluster$ and
$T' = T' - LeastMaxCluster - LeastMinCluster$;
7. Repeat steps 2-6 until $|T'| < 3k$;
8. if $2k \leq |T'| \leq (3k-1)$;
(i) Go to step 2;
(ii) Form the $LeastMaxCluster$ cluster with k records in T';
(iii) Form the $LeastMinCluster$ cluster with the remaining $(> k)$ records in T';
9. else;
10. if $T' < 2k$;
(i) Form a new cluster with all the remaining records in T';

5.2 Least Min-Max distance microaggregation algorithm

Based on the information loss measure in equation (1), the notion of minimum and maximum distance in Section 5.1 and the definition of the microaggregation problem, the Least Min-Max (LMMD) microaggregation algorithm is as follows:.

According to this method, first find the two most distant (Max_1 and Max_2) records and the two closest (Min_1 and Min_2) records in the dataset T using a distance metric. In this paper, the well-known Euclidean distance metric was used, but other distance metric including Manhattan or City-block distances could also be used. The algorithm (see Table 1) first builds two clusters using the Max_1 and Max_2 records as seeds. The first cluster C_{max_1} is built using Max_1 and choosing the nearest $(k-1)$ records from the dataset for which the information loss of the cluster C_{max_1} is the lowest. Similarly, the second cluster C_{max_2} is built using Max_2 and choosing the nearest $(k-1)$ records from the dataset. Now, the information loss is calculated for both C_{max_1} and C_{max_2}. The cluster with the lower information loss is retained and the other one is discarded. Next, two clusters C_{min_1} and C_{min_2} are created in a similar way but this time using Min_1 and Min_2 instead of using Max_1 and Max_2 records. Like before, the cluster with the lower information loss resulting from the two nearest points is

kept while the other one is discarded. Therefore, at the end of the first iteration the algorithm will create two clusters (one from Max and the other from Min distant records). This process is repeated until fewer than $3k$ records remain (see steps 2-7 of Table 1). The nearest records in a cluster are chosen in such a way that the inclusion of these records causes less SSE than the other records in the dataset. If between $2k$ and $(3k-1)$ records remain, then first cluster will be formed as before with k records and the second cluster with the remaining records having $k+1$ records to satisfy k-anonymity (see step 8 of Table 1). Finally, if fewer than $2k$ records remain, then just one new cluster is formed with all the remaining records (see step 10 of Table 1).

The proposed algorithms stated above endeavours to repeatedly build two clusters simultaneously using the Min- Max distance based approach which results in significantly reduced information loss than existing techniques (see Section 6).

Definition 2 (Least error clustering-based microaggregation decision problem) In a given dataset T of n records, there is a clustering scheme $C = \{C_1, C_2, ..., C_c\}$ such that

1. $\mid C_i \mid \geq k, 1 < k \leq n$: the size of each cluster is greater than or equal to a positive integer k, and
2. $\sum_{i=1}^{g} IL(C_i) \leq \epsilon, \epsilon > 0$: the total information loss of the clustering scheme is less than a positive integer ϵ.

where each cluster $C_i (i = 1, 2, ..., p)$ contains the records that are more similar to each other such that the cluster means are close to the values of the clusters and thus cause the least information loss.

6 Experimental Results

This section presents the experimental results and compares the results with several existing techniques. The objective of this experiment is to investigate the effectiveness of the proposed algorithm in terms of measured information loss of represented cluster data. The following three datasets [9], which have been used as benchmarks in previous studies to evaluate various microaggregation methods, were adopted in the experiments.

1. The "Tarragona" dataset contains 834 records with 13 numerical attributes.
2. The "Census" dataset contains 1,080 records with 13 numerical attributes.
3. The "EIA" dataset contains 4,092 records with 11 numeric attributes (plus two additional categorical attributes not used here).

To accurately evaluate our approach, the performance of the proposed algorithm is compared in this section with various microaggregation methods. Tables 2-4 show the information losses of these microaggregation methods. The lowest information loss for each dataset and each k value is shown in bold face.

Table 2. Information loss comparison using Tarragona dataset

Method	$k = 3$	$k = 4$	$k = 5$	$k = 10$
MDAV-MHM	16.9326		22.4617	33.1923
MD-MHM	16.9829		22.5269	33.1834
CBFS-MHM	16.9714		22.8227	33.2188
NPN-MHM	17.3949		27.0213	40.1831
M-d	16.6300	19.66	24.5000	38.5800
μ-Approx	17.10	20.51	26.04	38.80
TFRP-1	17.228	19.396	22.110	33.186
TFRP-2	16.881	19.181	21.847	33.088
MDAV-1	16.93258762	19.54578612	22.46128236	33.19235838
MDAV-2	16.38261429	19.01314997	22.07965363	33.17932950
DBA-1	20.69948803	23.82761456	26.00129826	35.39295837
DBA-2	16.15265063	22.67107728	25.45039236	34.80675148
LeastMinMaxDisPts	**2.16**	**5.12**	**7.01**	**9.19**

Table 3. Information loss comparison using Census dataset

Method	$k = 3$	$k = 4$	$k = 5$	$k = 10$
MDAV-MHM	5.6523		9.0870	14.2239
MD-MHM	5.69724		8.98594	14.3965
CBFS-MHM	5.6734		8.8942	13.8925
NPN-MHM	6.3498		11.3443	18.7335
M-d	6.1100	8.24	10.3000	17.1700
μ-Approx	6.25	8.47	10.78	17.01
TFRP-1	5.931	7.880	9.357	14.442
TFRP-2	5.803	7.638	8.980	13.959
MDAV-1	5.692186279	7.494699833	9.088435498	14.15593043
MDAV-2	5.656049371	7.409645342	9.012389597	13.94411775
DBA-1	6.144855154	9.127883805	10.84218735	15.78549732
DBA-2	5.581605762	7.591307664	9.046162117	13.52140518
LeastMinMaxDisPts	**1.3**	**2.21**	**2.27**	**2.66**

Table 4. Information loss comparison using EIA dataset

Method	$k = 3$	$k = 4$	$k = 5$	$k = 10$
MDAV-MHM	0.4081		1.2563	3.7725
MD-MHM	0.4422		1.2627	3.6374
NPN-MHM	0.5525		0.9602	2.3188
μ-Approx	0.43	0.59	0.83	2.26
TFRP-1	0.530	0.661	1.651	3.242
TFRP-2	0.428	0.599	0.910	2.590
MDAV-1	0.482938725	0.671345141	1.666657361	3.83966422
MDAV-2	0.411101515	0.587381756	0.946263963	3.16085577
DBA-1	1.090194828	0.84346907	1.895536919	4.265801303
DBA-2	0.421048322	0.559755523	0.81849828	2.080980825
LeastMinMaxDisPts	**2.21**	**0.64**	**5.52**	**4.2**

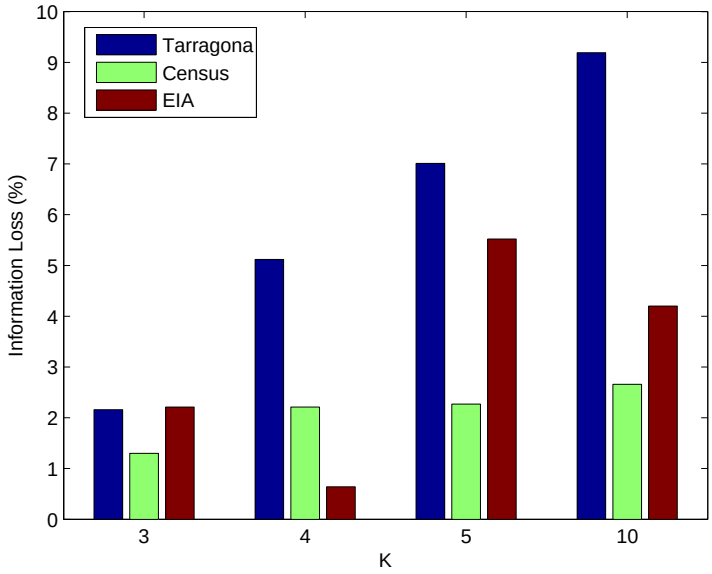

Fig. 2. Information Loss vs k for Tarragona, Census, and EIA datasets

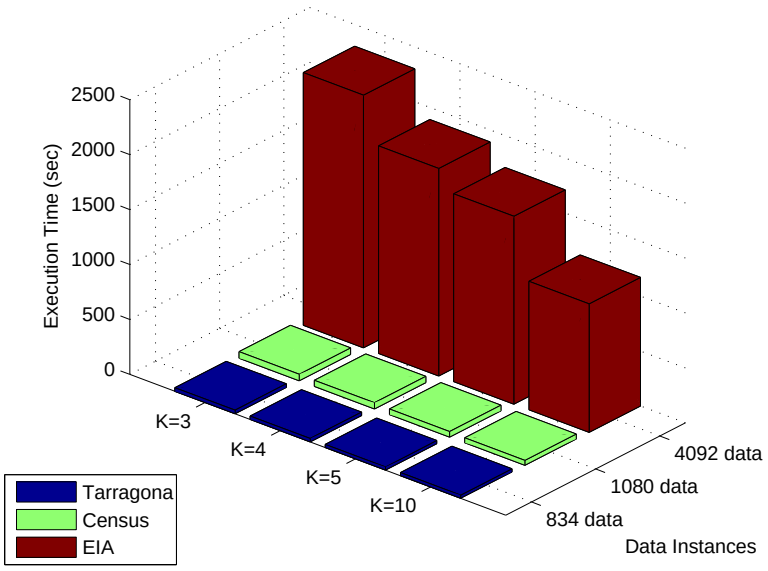

Fig. 3. Execution time vs k

The information losses of methods DBA-1, DBA-2, MDAV-1 and MDAV-2 are quoted from [28]; the information losses of methods MDAV-MHM, MD-MHM, CBFS-MHM, NPN-MHM and M-d (for $k = 3, 5, 10$) are quoted from [9]; the information losses of methods μ-Approx and M-d (for $k = 4$) are quoted from [10], and the information losses of methods TFRP-1 and TFRP-2 are quoted from [27]. TFRP is a two-stage method and its two stages are denoted as TRFP-1 and TRFP-2 respectively. The TFRP-2 is similar to the DBA-2 but disallows merging a record to a group of size over $(4k - 1)$.

Tables 2-4 show the information loss for several values of k and the Tarragona, Census and for the EIA datasets respectively. The information loss is compared with the proposed algorithm among the latest microaggregation methods listed above. Information loss is measured as $\frac{\text{SSE}}{\text{SST}} \times 100$, where SST is the total sum of the squares of the dataset. Note that the within-groups sum of squares SSE is never greater than SST so that the reported information loss measure takes values in the range [0,100]. Tables 2-4 illustrate that in all of the test situations, the proposed algorithm causes significantly less information loss than any of the microaggregation methods listed in the table. This shows the utility and the effectiveness of the proposed algorithm.

Analysis: Figure 2 shows how the information loss values changes with k for each dataset. Results indicate that information loss increases with k. This is obvious since the higher number of records in each cluster results in higher sum-of-squared-error (SSE) values due to the fact that each cluster now has more observations and possibly larger variance. Interestingly, there is little correlation between overall information loss of a dataset and its size as evident from the fact that the information loss for CIA dataset (containing 4092 instances) is much lower than the information loss for Tarragona dataset (containing 1082 instances). This may be due to the lower variance in EIA dataset resulting in clusters with lower SSE, hence lower information loss.

Figure 3 shows the how the execution time varies with k and different file sizes. Again, results show that the execution time depends on the value of k. It shows that the execution time increases slightly due to the increased number of permutations that need to be calculated for each cluster for the higher k. Furthermore, as expected the execution is also related to the file size. As shown in Figure 3 it takes the longest time to find k-anonymous clusters for the EIA dataset (4092 instances) and quickest time for the census dataset (834 instances).

7 Conclusion

Microaggregation is an effective method in SDC for protecting privacy in microdata and has been extensively used world-wide. The level of privacy required is controlled by a parameter k, often called the anonymity parameter. For k-anonymization, k is basically the minimum number of records in a cluster. Once the value of k has been chosen, the data protector and the data users are interested in minimizing the information loss. This work has presented a new multi-dimensional sorting technique for numerical attributes.The new method consists

of two stages. In the first stage it finds two pairs of Minimum and Maximum distant points. From this, the algorithm creates two k element clusters with the least information loss. In the second stage, it repeatedly creates these clusters until there are $p(k < p \leq 2k)$ records left. In which case, a single cluster is formed with the p points to preserve k-anonymity. A comparison has been made of the proposed algorithm with the most widely used microaggregation methods using the popular benchmark datasets. The experimental results show that the proposed algorithm **out-performs** all the tested microaggregation methods with respect to information loss. Thus the proposed method is very effective in preserving the privacy microdata sets and can be used as an effective privacy preserving k-anonymization method for Statistical Disclosure Control.

References

1. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Academic Publishers, Norwell (1981).
2. Domingo-Ferrer, J., Torra, V.: Privacy in data mining. Data Mining and Knowledge Discovery. 11 (2), 117–119 (2005)
3. Domingo-Ferrer, J., Mateo-Sanz, J.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and Data Engineering. 14(1), 189-201 (2002)
4. Domingo-Ferrer, J., Torra, V.: Extending microaggregation procedures using defuzzification methods for categorical variables. In: 1st international IEEE symposium on intelligent systems, pp. 44–49. Verna (2002)
5. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
6. Domingo-Ferrer, J., Torra, V.: Towards fuzzy c-means based microaggregation. In: Grzegorzewski, P., Hryniewicz, O., Gil, A. (eds.) Soft methods in probability, statistics and data analysis. Advances in soft computing, vol. 16, pp. 289-294. Heidelberg: Physica-Verlag (2002)
7. Domingo-Ferrer, J. and Torra, V.: Fuzzy microaggregation for microdata protection. Journal of Advanced Computational Intelligence and Intelligent Informatics. 7(2), 153-159 (2003)
8. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous kanonymity through microaggregation. Data Mining and Knowledge Discovery. 11(2), 195-212 (2005)
9. Domingo-Ferrer, J., Martinez-Balleste, A., Mateo-Sanz, J.M., Sebe, F.: Efficient multivariate data-oriented microaggregation. The VLDB Journal. 15(4), 355-369 (2006)
10. Domingo-Ferrer, J., Sebe, F., Solanas, A.: A polynomial-time approximation to optimal multivariate microaggregation. Computer and Mathematics with Applications. 55(4), 714-732 (2008)
11. Han, J.-M., Cen, T.,-T., Yu, H.-Q., Yu, J.: A multivariate immune clonal selection microaggregation algorithm. IEEE international conference on granular computing. pp. 252–256. Hangzhou (2008)

12. Hansen, S., Mukherjee, S.: A polynomial algorithm for optimal univariate microaggregation. IEEE Transactions on Knowledge and Data Engineering. 15(4), 1043-1044 (2003)
13. Laszlo, M., Mukherjee, S.: Minimum spanning tree partitioning algorithm for microaggregation. IEEE Transactions on Knowledge and Data Engineering. 17(7), 902-911 (2005)
14. Oganian, A., Domingo-Ferrer, J.: On the complexity of optimal microaggregation for statistical disclosure control. Statistical Journal of the United Nations Economic Commission for Europe. 18, 345-354 (2001)
15. Solanas, A.: Privacy protection with genetic algorithms. In: Yang, A. Shan, Y., Bui, L.T. (eds.) Success in evolutionary computation. Studies in Computional Intelligence, vol. 92, pp. 215-237. Heidelberg: Springer (2008)
16. Solanas, A., Martinez-Balleste, A., Domingo-Ferrer, J.: $V-MDAV$: A multivariate microaggregation with variable group size. In: 17th COMPSTAT Symposium of the IASC. Rome (2006).
17. Samarati, P.: Protecting respondent's privacy in microdata release. IEEE Transactions on Knowledge and Data Engineering. 13(6), 1010–1027 (2001)
18. Sweeney, L.: k-Anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems. 10(5), 557-570 (2002)
19. Torra, V.: Microaggregation for categorical variables: A median based approach. In: Domingo-Ferrer, J., Torra, V. (eds.)PSD 2004. LNCS, vol. 3050, pp. 162-174. Heidelberg: Springer , (2004)
20. Kabir, M.E., Wang,H.: Systematic Clustering-based Microaggregation for Statistical Disclosure Control. In: IEEE International Conference on Network and System Security, pp. 435-441, Melbourne (2010)
21. Kabir, M.E., Wang, H., Bertino, E., Chi, Y.: Systematic Clustering Method for l-diversity Model. In Australasian Database Conference, pp. 93-102, Brisbane (2010)
22. Kabir, M.E., Wang, H.: Microdata Protection Method Through Microaggragation: A Median Based Approach. Information Security Journal: A Global Perspective, 20(1), 1–8 (2011)
23. Ward, J.H.J.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association, 58 (301), 236-244 (1963)
24. Wang, H., Zhang, Y., Cao, J.: Effective collaboration with information sharing in virtual universities. IEEE Transactions on Knowledge and Data Engineering, 21(6), 840–853 (2009)
25. Willenborg, L., Waal, T.D.: Elements of statistical disclosure control. Lecture notes in statistics. 155 (2001)
26. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers, C-20 (1), 68-86 (1971).
27. Chang, C.-C., Li, Y.-C., Huang, W.-H.: TFRP: An efficient microaggregation algorithm for statistical disclosure control. Journal of Systems and Software, 80 (11), 1866–1878 (2007)
28. Lin, J.-L., Wen, T.-H., Hsieh, J.-C., Chang, P.-C.: Density-based microaggregation for statistical disclosure control. Expert Systems with Applications, 37(4), 3256–3263 (2010)