Microdata Protection Method Through Microaggregation: A Systematic approach

Md Enamul Kabir[1], Hua Wang[2]
[1]School of Engineering and Information Technology
University of New South Wales at the
Australian Defence Force Academy (UNSW@ADFA)
Northcott Drive, Canberra ACT 2600
E-mail: m.kabir@adfa.edu.au
[2]Department of Mathematics and Computing
University of Southern Queensland
Toowoomba, QLD 4350, Australia
Email: wang@usq.edu.au

Abstract— **Microdata protection in statistical databases has recently become a major societal concern and has been intensively studied in recent years. Statistical Disclosure Control (SDC) is often applied to statistical databases before they are released for public use. Microaggregation for 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. 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. This paper presents a clustering-based microaggregation method to minimize the information loss. The proposed technique adopts to group similar records together in a systematic way and then anonymized with the centroid of each group individually. The structure of systematic clustering problem is defined and investigated and an algorithm of the proposed problem is developed. Experimental results show that our method attains a reasonable dominance with respect to both information loss and execution time than the most popular heuristic algorithm called Maximum Distance to Average Vector (MDAV).**

Index Terms— **Privacy, Microaggregation, Microdata protection, k-anonymity, Disclosure control**

I. Introduction

In recent years, the phenomenal advance 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. To protect personal data from individual identification, SDC is often applied before the data are released for analysis [3], [27]. 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 to suppress or alter 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.

Microaggregation is a family of SDC methods for protecting microdata sets that have been extensively studied recently [4], [5], [7], [8], [11]–[14], [16]. 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 [24], 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 a 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 been extended to handle categorical and ordinal attributes [5], [8], [25]. In this paper we proposed a microaggregated method that also only applicable for the numeric attributes.

The effectiveness of a microaggregation method is measured by calculating its information loss. A lower information loss implies that the anonoymized dataset is less distorted from the original dataset, and thus provides better data quality for analysis. k- anonymity [15], [23], [24] provides sufficient protection of personal confidentiality of microdata, while to ensure the quality of the anonymized dataset, an effective microaggregation method should incur information loss as minimum as possible. In order to be useful in practice, the dataset should keep as much informative as possible. Hence, it is

This paper is based on "Systematic Clustering-based Microaggregation for Statistical Disclosure Control," by M.E. Kabir and H. Wang, which appeared in the Proceedings of the IEEE 4th International Conference on Network and System Security (NSS), Melbourne, Australia, September 2010. © 2010 IEEE.

doi:10.4304/jsw.7.11.2415-2423

necessary to consider deeply 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 at each variable are the same. 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. So 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 [4] proposed univariate and multivariate k-Ward algorithms that extend the agglomerative hierarchical clustering method of Ward et al. [26]. Domingo-Ferrer and Torra [6], [7] proposed a microaggregation method based on the fuzzy c-means algorithm [1], and Laszlo and Mukherjee [17] extended the standard minimum spanning tree partitioning algorithm for microaggregation [28]. 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 cluster at a time. Notable examples include Maximum Distance [21], Diameter-based Fixed-Size microaggregation and centroid-based Fixed-size microaggregation [17], Maximum Distance to Average Vector (MDAV) [4], [8], MHM [9] and the Two Fixed Reference Points method [29]. Most recently, Lin *et al.* [30] proposed a density-based microaggregation method that forms records by the descending order of their densities, then fine-tunes these clusters in reverse order.

All the works stated above proposed different microaggregation algorithms to form the clusters, where within clusters the records are homogeneous but between clusters the records are heterogeneous such that information loss is low. However, no single microaggregation method outperform other methods in terms of information loss. This work presents a new clustering method for microaggregation, where all clusters are made simultaneously in a systematic way. According to this method, sort all records by using a sorting function and partitions all records into $[\frac{n}{k}]$ clusters, where n is the total number of records and k is the k-anonymity parameter. Randomly select a record r from first k records to form the first cluster and the first records of the subsequent clusters form in a systematic way. Then adjusts the records in each cluster in a systematic way such that each cluster contains at least k records. Performance of the proposed method is compared against the MDAV [4] as MDAV is the most widely used microaggregation method. The experimental results show that the proposed microaggregation method outperforms MDAV with respect to both information loss and computational efficiency.

The reminder of this paper is organized as follows. Section II introduces the basic concept of microaggregation. Section III reviews previous microaggregation methods. We present a brief description of our proposed microaggregation method in Section IV. Section V shows experimental results of the proposed method. Finally, concluding remarks are included in Section VI.

II. 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. A microdata set $\boldsymbol{V}$ can be viewed as a file with n records, where each record contains p attributes on an individual respondent. The attributes in an original unprotected dataset can be classified in four categories which are not necessarily disjoint:

- **Identifiers**: These are attributes that unambiguously identify the respondent. Examples are passport number, social security number, full name, etc. Since our objective is to prevent confidential information from being linked to specific respondents, we will assume in what follows that in a pre-processing step, identifiers in $\boldsymbol{V}$ have been removed.
- **Quasi-identifiers**: A quasi-identifiers is a set of attributes in $\boldsymbol{V}$ that in combination can be linked with external information to re-identify (some of) the respondents to whom (some of) the records in $\boldsymbol{V}$ refer. Unlike identifiers, quasi-identifiers cannot be removed from $\boldsymbol{V}$. The reason is that any attribute in $\boldsymbol{V}$ potentially belongs to a quasi-identifiers depending on the external data sources available to the user of $\boldsymbol{V}$. As releasing microdata about individuals poses a privacy threat due to quasi-identifiers, microaggregation only consider the quasi-identifiers.
- **Confidential outcome attributes**: These are attributes which contain sensitive information on the respondent. Examples are salary, religion, political affiliation, health condition, etc.
- **Non-confidential outcome attributes**: Those attributes which contain non-sensitive information on the respondent. Examples are town and country of residence, etc. Note that attributes of this kind cannot be neglected when protecting a dataset because they can be a part of a quasi-identifier.

The purpose of microdata SDC can be stated more formally by saying that given an original microdataset $\boldsymbol{V}$, the goal is to release a protected microdataset $\boldsymbol{V}^{'}$ in such a way that

1) Disclosure risk (i.e., the risk that a user or an intruder can use $\boldsymbol{V}^{'}$ to determine confidential attributes on a specific individual among those in $\boldsymbol{V}$) is low.
2) User analysis (regressions, means, etc.) on $\boldsymbol{V}^{'}$ and $\boldsymbol{V}$ yield the same or at least similar results. This is equivalent to requiring that information loss caused

by SDC should be low, i.e., that the utility of the SDC-protected data should stay high.

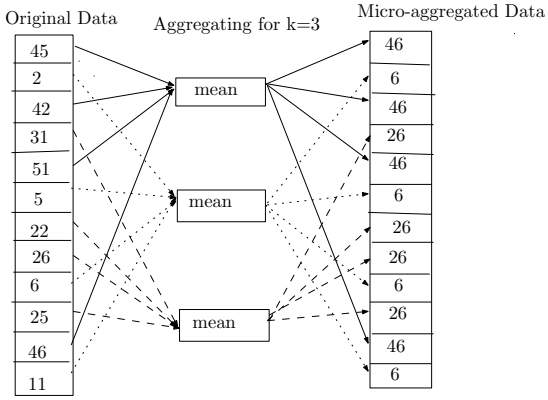

Figure 1. Example of Microaggregation using mean

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 for 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 x_i is calculated as the average vector of all the records in the ith cluster. In order to determine whether two records are similar, a similarity function such as the Euclidean distance, Minkowski distance or Chebyshev distance can be used. A common measure is the Sum of Squared Errors (SSE). The SSE is the sum of squared distances from the centroid of each cluster to every record in the cluster, and is defined as:

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

The lower the SSE, the higher the within cluster homogeneity and higher the SSE, the lower the within cluster homogeneity. If all the records in a cluster are 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. Thus SSE can be treated as a measurement of information loss due to microaggregation. In this paper, we used SSE as a measure of information loss during the microaggregation process. 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 $\boldsymbol{G} = \{G_1, G_2, ..., G_g\}$ of T such that

1) $G_i \cap G_j =$, for all $i \neq j = 1, 2, ..., p$,
2) $\cup_{i=1}^{p} G_i = T$,
3) SSE is minimized,
4) for all $G_i \in T$, $\mid G_i \mid \geq k$ for any $G_i \in \boldsymbol{G}$.

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 [19]. 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 [4] showed that no cluster should contain more than $(2k-1)$ records since such clusters can always be partitioned to further reduce information loss.

III. Previous Microaggregation Methods

Previous microaggregation methods have been roughly divided into two categories, namely fixed-size and data-oriented microaggregation [4], [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 [30]. 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.

The Maximum Distance (MD) method [21] 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 [17] 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.

Domingo-Ferrer and Mateo-Sanz [4] proposed a multivariate fixed-size microaggregation method, called MDAV

which is the most widely used microaggregation method. 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 cluster 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. [30] 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 [30]. 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 [17] 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 finds 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. Kabir and Wang [13] proposed a systematic clustering-based microaggregation for SDC. Extending the systematic idea, Kabir et.al. [14] proposed a pairwise-systematic microaggregation that incurs less information loss than the latest microaggregation methods for all the test situations.

Solanas et al. [22] 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 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. [29] 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 G_1 has N_{min} as its value for all attributes, and another reference point G_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 G_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 G_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 G_1 and G_2 are fixed throughout the iterations. When all clusters are generated, TFRP applies an enhancement step to determine whether a cluster should be retained or decomposed and added to other clusters.

Lin et al. [30] 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. Heuristics are then applied to either decompose a tree to reduce the cluster size to be fewer than $2k$ or merge trees to raise the cluster size to be greater than or equal to k. Instead of using trees, other methods may adaptively adjust the number of clusters to ensure that the size of each cluster is between k and $(2k-1)$.

The multivariate k-Ward algorithm [4] 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. The k-Ward algorithm tends to generate large clusters, consequently increasing the information loss. For instance, this method could merge two clusters, each with $(k-1)$ records to form a large cluster of $(2k-2)$ records. The minimum spanning tree microaggregation method [17] first builds a minimal spanning tree (MST) of the dataset using the Prim method [2]. Then, as in the standard MST partitioning algorithm [28], the longest edge is recursively removed to form a forest of subtrees of the MST. However, unlike in the standard MST partitioning algorithm, the longest edge is removed only if both the resulting subtrees contain at least k nodes. Finally, another microaggregation method (such as MDAV) is applied to those groups containing more than $2k$ records. According to the experimental results reported by Laszlo and Mukherjee [17], this method has the same complexity as the multivariate k-Ward algorithm but causes less information loss. However, it still tends to generate large groups and works well only if the dataset has well-separated clusters.

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 a 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 and Mukherjee [12] proposed a microaggregation method for univariate dataset called, HM. This method converts a dataset into a directed acyclic graph based on the ordering of the records and then transforms the microaggregation problem into the shortest path problem, which can be solved in polynomial time. This method cannot be applied directly to multivariate datasets since these only have a partial ordering among records. 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 a 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. Genetic algorithms (GAs) have also been applied to the microaggregation problem. Solanas et al. [20] encoded a partitioning of a dataset as a chromosome of n genes, where n is the number of records in the dataset and the value of the ith gene indicates the cluster number of the ith record in the dataset. Since each cluster contains at least k records, each cluster number is an integer in the interval $[1, \lfloor \frac{n}{k} \rfloor]$. When generating the initial population of chromosomes and performing genetic operations on these chromosomes, special care must be taken to avoid generating a chromosome where any cluster numbers appear fewer than k or more than $2k$ times in their n genes. The experimental results showed that this method works well for small datasets ($n \leq 50$). Therefore they recommended first using a fixed-size microaggregation method such as MDAV to generate clusters with $k = 50$ and then applying GA for the real intended k value for each cluster. This two-step method was later studied by Martnez-Ballest et al. [18] and was also published in Solanas [21].

IV. The Proposed Approach

This section presents the proposed systematic clustering-based algorithm for microaggregation that minimizes the information loss and satisfy the k-anonymity requirement. The proposed approach builds and refines all clusters simultaneously.

A. Sorting Function

According to the proposed approach, first sort all records with respect to the attributes. So it is necessary to define a sorting function to sort all the records in the dataset. Consider a microdata set T with p numeric attributes, namely $Y_1, Y_2, ..., Y_p$ and n records. Thus each record is represented as a vector in a p-dimensional space. To sort all the records with respect to the numeric attributes, we define the jth sorted record in the dataset T is as follows:

$$SF_j = \sum_{i=1}^{p} (y_{ij} - y_i), \quad j = 1, 2..., n. \tag{2}$$

where, y_{ij} is the jth record of the ith attribute and y_i is the centroid of the ith attribute. The SF stated above measures the distance between the records and their corresponding centroid. In this study, the SF are arranged in ascending order indicating records are arranged in order of magnitude. The lower the values of SF, the records are below their respective centroid and the higher the values of SF, the records are above their respective centroid. Thus the records in the dataset T sorted in ascending order based on the SF and the first and the last record are most distant among all other records in the dataset T.

B. Systematic microaggregation algorithm

Based on the information loss measure in equation (1) and the definition of microaggregation problem, we are now ready to discuss the systematic clustering-based microaggregation algorithm. The general idea of the algorithm is as follows.

TABLE I.
SYSTEMATIC CLUSTERING-BASED MICROAGGREGATION ALGORITHM

Input: a dataset T of n records and a positive integer k
Output: a partitioning $\boldsymbol{G} = \{G_1, G_2, ..., G_g\}$ of T
where $g = |\boldsymbol{G}|$ and $G_i \geq k$ for $i = 1$ to g.
1. Sort all records in T in ascending order by using the SF in equation (2);
2. Let g:= int$\lfloor \frac{n}{k} \rfloor$;
3. Get randomly k distinct records $r_1, r_2, ..., r_k$ from first 1 to k;
4. Let x_{ij} is the jth record in the ith cluster;
5. For $i = 1$ to g;
6. Let $x_{i1} := T_{[r_1+k(i-1)]}$;
7. Next i;
8. For $j := 2$ to k;
9. For $i := 1$ to g;
10. Let IL_i: = InfoLoss($T_{[r_j+k(i-1)]}$);
11. Let N:= Find cluster number with lowest IL_i;
12. where cluster size $\leq k$;
13. Add $T_{[r_j+k(i-1)]}$ to g_n;
14. Next i;
15. Next j;
16. Let $e := (n - gk)$;
17. Find extra element $E_1, E_2, ..., E_e \in E$;
18. For $k := 1$ to e;
19. For $m := 1$ to g;
20. Let $IL_m :=$ InfoLoss(E_k) in cluster m;
21. Next m;
22. Let $N :=$ Find cluster with lowest IL;
23. Add E_k to g_n;
24. Next k;

According to this method first sort all records in ascending order by using the sorting function in equation (2). Then identify the equivalence class and the number of clusters by, $g = \frac{n}{k}$, where n is the total number of records in the dataset T, k is anonymity parameter for k-anonymization. Round this as integer and randomly select a record r_i from first k records as seed to form the first cluster. If there are g clusters to be formed then select the $(r_i + k)$th, $(r_i + 2k)$th,..., $\{r_i + (g-1)k\}$th records in a systematic way to form 2nd, 3rd, ..., gth cluster respectively. Select another record $r_j (j \neq i)$ from the first k records and add this record to the cluster which causes least information loss. Similar in a systematic way select $(r_j + k)$th, $(r_j + 2k)$th,..., $\{r_j + (g-1)k\}$th records and add these records to their respective clusters that cause least information loss. If any cluster size is exactly k, stop adding records to that cluster and continue the same process until all records of first k records finish. If n is not exactly divisible by k and still there are some records left, add these records to their closest clusters that incur least information loss. Systematic microaggregation algorithm endeavor to build all clusters simultaneously, whereas most of the microaggregation algorithms in the literature build one/two cluster(s) at a time. The algorithm selects first record randomly and the subsequent records from in a systematic way. As the records in the dataset T are arranged in ascending order and the first record of each cluster forms in every kth distance, the first record of each cluster contains non identical value, so this algorithm easily captures if there are any extreme values in the dataset. The systematic microaggregation algorithm is shown in Table I.

Definition 2 (Systematic clustering-based microaggregation decision problem) In a given dataset T of n records, there is a clustering scheme $\boldsymbol{G} = \{G_1, G_2, ..., G_g\}$ such that

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

where each cluster $G_i (i = 1, 2, ..., p)$ 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 causes least information loss.

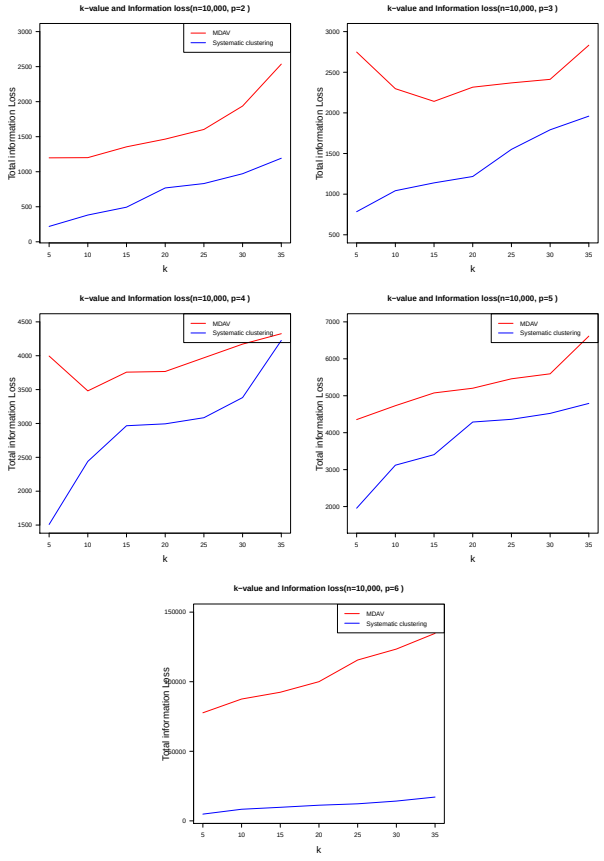

Figure 2. Information Loss comparison for no. of attributes between 2 and 6

V. EXPERIMENTAL RESULTS

The objective of our experiment is to investigate the recital of our approach in terms of data quality and the computational efficiency. This section experimentally evaluates the effectiveness and efficiency of the systematic clustering-based microaggregation algorithm. For this purpose, we utilize a real dataset CENSUS[1] containing personal information of 500 thousands American adults. The dataset has 9 discrete attributes.

[1]Downloadable at *http://www.ipums.org.*

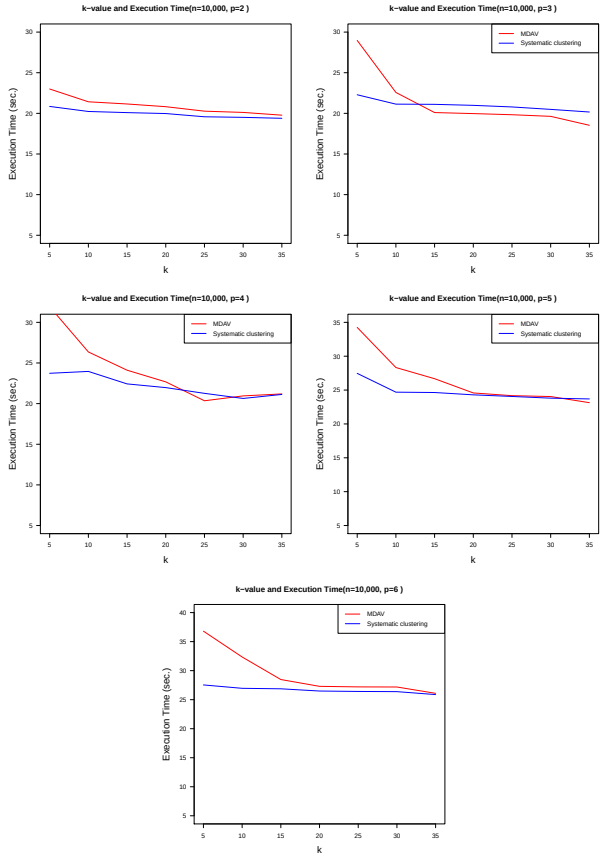

Figure 3. Running time comparison using census datset for no. of attributes between 2 and 6

To accurately evaluate our approach, the performance of the proposed algorithm is compared in this section with MDAV [4] as until now MDAV is the most widely used microaggregation method. For the experiment we have selected 10 thousands records randomly from the whole dataset and run the experiment for $k = 5, 10, ..., 35$ and for different situations of number of attributes, $p = 2, 3, ..., 6$.

A. Data Quality and Efficiency

In this section, we report experimental results on the systematic clustering-based microaggregation algorithm for data quality and execution efficiency. In this paper, SSE defined in equation (1) is used to measure the information loss due to microaggregation.

Figure 2 reports the information loss of both the MDAV and the systematic clustering-based microaggregation algorithms for increasing the values of k and p, where p is the number of attributes in the dataset. With the increase of k, the information loss is increasing for both the algorithms. As the figure illustrates, the systematic clustering-based microaggregation algorithm results in the least cost of the information loss for both all k and p values. The superiority of our algorithm over the MDAV algorithm results from the fact that our algorithm easily captures if there are any extreme values because of sorting

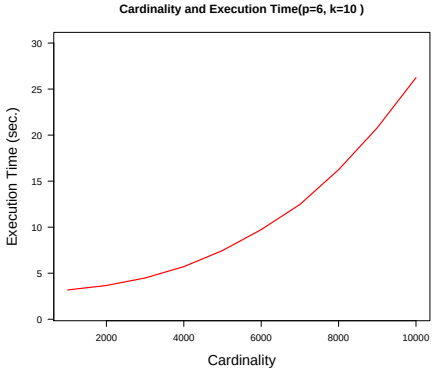

Figure 4. Cardinality and Runtime

function and the systematic way of selecting records in the clusters.

On the other hand, Figure 3 displays the execution (running) time of both the algorithms. In general the running time is decreasing with the increase of k in all scenarios. Figure 3 clearly shows that the running time of the proposed algorithm with all different scenarios are much lesser than the MDAV algorithm for almost all values of k. However, as shows in Figure 3, for some moderate values of k, the running time of the proposed algorithm is little bit more (in some situations) than the MDAV. We believe that that is still acceptable in practice considering its better performance with respect to the information loss.

B. Scalability

Figure 4 shows the execution time behaviors of the systematic clustering-based microaggregation algorithm for various cardinalities with $p = 6$ and $k = 10$. For this experiment we used subsets of the Census dataset with different sizes. As shown, the running time increases almost linearly with the size of the dataset for our proposed algorithm. Again the proposed algorithm introduces the least information loss for any p and k. This shows that our approach preserves the quality of the data and is highly scalable.

VI. Conclusion

Microaggregation is an effective method of protecting privacy in microdata. This work presents a new systematic clustering-based microaggregation method for numerical attributes. The new method consists of clustering individuals records in microdata in a number of disjoint clusters in a systematic way prior publication and then publish the mean over each cluster instead of individual records. A comparison is made on the proposed algorithm with the most widely used microaggregation method, called MDAV through experiment. In the microaggregation problem, the performance of a method is judged by both information loss and the running time. A method that incurs less information loss and has less execution time is the powerful method. The experimental results show

that the proposed algorithm has a significantly reasonable dominance over the MDAV with respect to both information loss and execution time. Finally it has shown by experiment that the proposed algorithm is highly scalable.

ACKNOWLEDGMENT

The authors thankfully acknowledge valuable contribution of Mr. Xiaoxun Sun for assistance with the experiments.

REFERENCES

[1] J.C. Bezdek, *Pattern recognition with fuzzy objective function algorithms.* Norwell, MA: Academic Publishers, 1981.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to algorithms (2nd ed.).* The MIT Press, 2001.

[3] J. Domingo-Ferrer and V. Torra, "Privacy in data mining," *Data Mining and Knowledge Discovery,* vol. 11, no. 2, pp. 117–119, 2005.

[4] J. Domingo-Ferrer and J. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, no. 1, pp. 189-201, 2002.

[5] J. Domingo-Ferrer and V. Torra, "Extending microaggregation procedures using defuzzification methods for categorical variables," in *Proc. 1st international IEEE symposium on intelligent systems, Verna, Sept. 2002,* pp. 44–49.

[6] J. Domingo-Ferrer and V. Torra, "Towards fuzzy c-means based microaggregation," in *Soft methods in probability, statistics and data analysis,* P. Grzegorzewski, O. Hryniewicz and M.A. Gil, Eds. Heidelberg: Physica-Verlag, 2002, Advances in soft computing, vol. 16, pp. 289-294.

[7] J. Domingo-Ferrer and V. Torra, "Fuzzy microaggregation for microdata protection," *Journal of Advanced Computational Intelligence and Intelligent Informatics,* vol. 7, no. 2, pp. 153-159, 2003.

[8] J. Domingo-Ferrer and V. Torra, "Ordinal, continuous and heterogeneous kanonymity through microaggregation," *Data Mining and Knowledge Discovery,* vol. 11, no. 2, pp. 195-212, 2005.

[9] J. Domingo-Ferrer, A. Martinez-Balleste, J.M. Mateo-Sanz and F. Sebe, "Efficient multivariate data-oriented microaggregation," *The VLDB Journal,* vol. 15, no. 4, pp. 355-369, 2006.

[10] J. Domingo-Ferrer, F. Sebe and A. Solanas, "A polynomial-time approximation to optimal multivariate microaggregation," *Computer and Mathematics with Applications,* vol. 55, no. 4, pp. 714-732, 2008.

[11] J.-M. Han, T.,-T. Cen, H.-Q. Yu and J. Yu, "A multivariate immune clonal selection microaggregation algorithm," in *Proc. IEEE international conference on granular computing, Hangzhou, Feb. 2008,* pp. 252-256.

[12] S. Hansen and S. Mukherjee, "A polynomial algorithm for optimal univariate microaggregation," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, no. 4, pp. 1043-1044, 2003.

[13] M.E. Kabir and H. Wang, "Systematic Clustering-based Microaggregation for Statitical Disclosure Control," in *Proc. 4th International Conference on Network and System Security, Melbourne, Sept. 2010,* pp. 435–441.

[14] M.E. Kabir, H. Wang and Y. Zhang, "A Pairwise-Systematic Microaggregation for Statitical Disclosure Control," in *Proc. 10th IEEE International Conference on Data Mining, Sydney, Dec. 2010,* pp. 266–273.

[15] M.E. Kabir, H.Wang and E. Bertino, "Efficient systematic clustering method for k-anonymization," *Acta Informatica,* vol. 48, no. 1, pp. 51–66, 2011.

[16] M.E. Kabir and H. Wang, "Microdata Protection Method Through Microaggragation: A Median Based Approach," *Information Security Journal: A Global Perspective,* vol. 20, no. 1, pp. 1–8, 2011.

[17] M. Laszlo and S. Mukherjee, "Minimum spanning tree partitioning algorithm for microaggregation," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, no. 7, pp. 902-911, 2005.

[18] A. Martinez-Balleste, A. Solanas, J. Domingo-Ferrer and J.M. Mateo-Sanz, "A genetic approach to multivariate microaggregation for database privacy," in *Proc. IEEE 23rd international conference on data engineering workshop, Cancun, April, 2007,* pp. 180-185.

[19] A. Oganian and J. Domingo-Ferrer, "On the complexity of optimal microaggregation for statistical disclosure control," *Statistical Journal of the United Nations Economic Commission for Europe,* vol.18, 345-354, 2001.

[20] A. Solanas, A. Martinez-Balleste, J. Mateo-Sanz and J. Domingo-Ferrer, "Multivariate microaggregation based genetic algorithms," in *Proc. IEEE third international conference on intelligent systems, Varna, Sept., 2006,* pp. 65-70.

[21] A. Solanas, "Privacy protection with genetic algorithms," in *Success in evolutionary computation,* A. Yang, Y. Shan and L.T. Bui, Eds. Heidelberg: Springer, 2008, Studies in Computional Intelligence, vol. 92, pp. 215-237.

[22] A. Solanas, A. Martinez-Balleste and J. Domingo-Ferrer, "$V - MDAV$: A multivariate microaggregation with variable group size," in *Proc. 17th COMPSTAT Symposium of the IASC, Rome, Aug. 2006*.

[23] P. Samarati, "Protecting respondent's privacy in microdata release," *IEEE Transactions on Knowledge and Data Engineering,* vol. 13, no. 6, pp. 1010–1027, 2001.

[24] L. Sweeney, "k-Anonymity: A model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,* vol. 10, no. 5, pp. 557-570, 2002.

[25] V. Torra, "Microaggregation for categorical variables: A median based approach," in *PSD 2004,* J Domingo-Ferrer and V. Torra, Eds. Heidelberg: Springer, 2004, LNCS, vol. 3050, pp. 162-174.

[26] J.H.J. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association,* vol. 58, no. 301, pp. 236-244, 1963.

[27] L. Willenborg and T.D. Waal, "Elements of statistical disclosure control," *Lecture notes in statistics,* vol. 155, 2001.

[28] C.T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers,* vol. C-20, no. 1, pp. 68-86, 1971.

[29] C.-C. Chang, Y.-C. Li and W.-H. Huang, "TFRP: An efficient microaggregation algorithm for statistical disclosure control," *Journal of Systems and Software,* vol. 80, no. 11, pp. 1866–1878, 2007.

[30] J.-L. Lin, T.-H. Wen, J.-C. Hsieh and P.-C. Chang, "Density-based microaggregation for statistical disclosure control," *Expert Systems with Applications,* vol. 37, no. 4, pp. 3256–3263, 2010.

Md Enamul Kabir is a postdoctoral research fellow at the University of New South Wales at the Australian Defence Force Academy (UNSW@ADFA). He completed his Ph.D. degree in Data Mining from the Department of Mathematics and Computing, University of Southern Queensland. He has served as an Assistant Professor at the Department of Statistics, University of Dhaka, Bangladesh. His research interests are in the field of privacy preservation, and particularly in access control, data anonymization and Statistical disclosure control.

Hua Wang is a professor at the University of Southern Queensland, after having earned a Ph.D. degree from that same university. He has been active in the areas of information systems management, distributed database management systems, access control, and data mining. He has participated in research projects on mobile electronic systems,Web service, and role-based access control for electronic service systems. Professor Wang has published over 100 research papers.