MDAV2K: A VARIABLE-SIZE MICROAGGREGATION TECHNIQUE FOR PRIVACY PRESERVATION

S. K. Chettri[1] and B. Borah[2]

[1]Department of Computer Science, Saint Mary's College, Shillong, India
s.chettri@smcs.ac.in
[2]Department of Computer Science and Engineering, Tezpur University, Tezpur, India
bgb@tezu.ernet.in

ABSTRACT

Public and private organizations are collecting personal data regarding day to day life of individuals and accumulating them in large databases. Data mining techniques may be applied to such databases to extract useful hidden knowledge. Releasing the databases for data mining purpose may lead to breach of individual privacy. Therefore the databases must be protected through means of privacy preservation techniques before releasing them for data mining purpose. Microaggregation is a privacy preservation technique used by statistical disclosure control community as well as data mining community for microdata protection. The Maximum distance to Average Vector (MDAV) is a very popular multivariate fixed-size microaggregation technique studied by many researchers. The principal goal of such techniques is to preserve privacy without much information loss. In this paper we propose a variable-size, improved MDAV technique having low information loss.

KEYWORDS

Privacy preservation, data mining, microaggregation, variable-size, information loss.

1. INTRODUCTION

Data mining techniques have lead to the rapid extraction of previously unknown knowledge from huge collection of data. Advancement of computer technology has enabled different public and private organization to easily collect, store and manage large amount of data including personal data regarding individuals. Such datasets may be released to data miners for research analyses. However, the privacy of individuals may be affected if the databases got published or outsourced for analysis, as the information contained in the databases may be sensitive. There exist several kinds of disclosure risks to the privacy of the individuals [1]. Thus the problem is, how to extract relevant knowledge from large amount of data while protecting at the same time sensitive information existing in the database, which may raise various social and ethical issues if got revealed. Therefore, before conducting data mining the data should be protected through some privacy preserving techniques [2]. This problem has been discussed by multiple communities such as privacy preserving data mining (PPDM) community, statistical disclosure control (SDC) community, database community etc. The protection provided by privacy preserving methods normally results in some degree of data modification or masking, which may be perturbative or nonperturbative. The challenge is to tune the modification of data so that both privacy risk and information loss are kept below certain acceptable limits. A survey of perturbative protection methods can be found in [1].

Microaggregation is a perturbative technique for microdata (i.e individual records) protection appearing in statistical disclosure control [3]. Given a dataset, it builds small groups of at least *k* records, with *k* being a user-definable parameter. Then, the microaggregated data set is built by replacing each original record by the centroid of the group it belongs to. The microaggregated data set can be released without jeopardizing the privacy of the individuals which form the original data set because *k* records have an identical protected value. To minimize information loss caused by microaggregation, records within each group should be as homogeneous as possible. Multivariate microaggregation with maximum within group records homogeneity is NP-hard[4], so heuristics are normally used. There exist two main types of heuristics: *fixed-size microaggregation* [5-11] and *data-oriented microaggregation* [12-14]. The former yield groups with a fixed number of records, except possibly one group and the later yield groups whose size varies depending on the distribution of the original records. Fixed-size microaggregation heuristics are computationally very efficient, due to their simplicity. On the other hand, data-oriented heuristics can often achieve lower information loss because they are able to adapt the choice of group sizes to the structure of the dataset.

In [5], optimal microaggregation is defined as the one yielding a *k*-partition maximizing the within-groups homogeneity; the higher the within-groups homogeneity, the lower the information loss, since microaggregation replaces values in a group by the group centroid. They showed that groups in an optimal *k*-partition contain between *k* and 2*k*-1 records. The sum of square error (*SSE*) criterion is common to measure homogeneity in clustering [9]. In terms of *SSE*, maximizing within-groups homogeneity is equivalent to finding a *k*-partition minimizing the within group *SSE*. The goal of microaggregation is to minimize the *SSE* measure, which is defined as:

$$SSE = \sum_{i=1}^{g} \sum_{x_{ij} \in c_i} (x_{ij} - \bar{x}_i)'(x_{ij} - \bar{x}_i)$$

Where g is the total number of clusters (groups), c_i is the i-th cluster and $\bar{x}_i$ is the centroid of c_i. The total sum of square *SST* is the sum of square error within the entire dataset calculated by summing the Euclidean distance of each record x_{ij} to the centroid $\bar{x}$ as follows:

$$SST = \sum_{i=1}^{g} \sum_{x_{ij} \in c_i} (x_{ij} - \bar{x})'(x_{ij} - \bar{x})$$

Microaggregation techniques are often compared on the basis of the *SSE* or the *IL* (information loss) measure. The measure *IL* standardized between 0 and 100 can be obtained from *SST* as:

$$IL = \frac{SSE}{SST} \cdot 100$$

The *Maximum distance to Average Vector* (*MDAV*) is a very popular multivariate fixed-size microaggregation technique. In this paper we present a variable-size *MDAV* technique for numeric data producing low information loss. It shows better results in comparison to other reported methods.

Rest of the paper is organized as follows. In Section 2 we introduce the different variants of *MDAV* microaggregation methods. Section 3 describes the proposed variable-size *MDAV* algorithm. In Section 4, experimental results are presented and the effectiveness of the proposed algorithm is assessed. Finally, in Section 5 conclusions are drawn.

2. *MDAV* MICRODATA PROTECTION ALGORITHMS

The *MDAV* method is one of the best heuristic methods for multivariate microaggregation. The algorithm was proposed in [6, 7] as part of a multivariate microaggregation method implemented in the *μ-Argus* package for statistical disclosure control. Several variant of this fixed-size microaggregation algorithm exists [6, 7, 8, 14] with minor differences. Some of them are presented here for comparison purpose and to build the foundation for the proposed algorithm. Firstly, we present the basic version of the *MDAV* algorithm as presented in [13] below. The dataset *X* with *n* records and value of group size *k* are to be provided as input to the algorithm.
Algorithm-1 (The *MDAV* algorithm):

```
1.     set i=1; $n$=|$X$|;
2.     while ($n \geq 2k$) do
2.1        compute centroid $\bar{x}$ of remaining records in $X$;
2.2        find the most distant record $x_r$ from $\bar{x}$;
2.3        find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
2.4        form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
2.5        remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
2.6        set $n=n-k$; $i=i+1$;
2.7        find the most distant record $x_s$ from $x_r$;
2.8        find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_s$;
2.9        form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
2.10       remove records $y_1,y_2,\ldots,y_k$ from $X$;
2.11       set $n=n-k$; $i=i+1$;
2.12.  end while
3.     if ($n \geq k$) then
3.1        form a cluster $c_i$ with the $n$ remaining records;
3.2        set $n=n-n$; $i=i+1$;
3.3    endif
4.     if ($n>0$) then
4.1        compute centroid $\bar{x}$ of remaining records in $X$;
4.2        find the closest cluster centroid $\bar{c}_j$ from $\bar{x}$;
4.3        add the remaining records to cluster $c_j$;
```

```
4.4    endif
5.     end algorithm
```

Given a dataset X with n records, *MDAV* iterates building two groups, each of size k until number of remaining unassigned records to any group becomes less than $2k$ (step 2). In order to build these groups, the centroid $\bar{x}$ of the remaining unassigned records is computed at the beginning of each iteration. Then the most distant record x_r from $\bar{x}$ is found and a group is built with the k-nearest neighbors of x_r including itself. These k records are removed from the dataset X. Next, the most distant record x_s from x_r is found and another group is built with the k-nearest neighbours of x_s. When the remaining records after termination of iterations is between k and $2k$-1 *MDAV* simply forms a group with all of them (step 3). If less than k records remain all the records of this subgroup are assigned to its closest group determined by computing distance between centroids of the groups (step 4). All groups have k elements except only one. Finally, given the k-partition obtained by *MDAV*, a microaggregated data set is computed by replacing each record in the original dataset by the centroid of the group to which it belongs. This step is not shown in the algorithm.

The *MDAV-generic* [7] algorithm is a variant of the *MDAV* algorithm. This algorithm smoothly handles the remaining records after the iterations terminate (steps 3 and 4). Since the algorithm iterates until there are at least $3k$ remaining records (note the difference with *algorithm*-1 in step 2), there will be between k and $3k$-1 records left unassigned after iterations terminate. If there are at least $2k$ records, a cluster is formed with the k-nearest neighbours of the most distant record from the centroid as usual after which at least k records remain unassigned (step 3). If less than $2k$ records remain, a cluster is formed with all of the remaining records (step 4). The algorithm is produced below.

Algorithm-2 (The *MDAV-generic* algorithm):

```
1.     set $i$=1; $n$=|$X$|
2.     while ($n \geq 3k$) do
2.1        compute centroid $\bar{x}$ of remaining records in $X$;
2.2        find the most distant record $x_r$ from $\bar{x}$ ;
2.3        find $k$-nearest neighbours $y_1, y_2, \ldots, y_k$ of $x_r$;
2.4        form cluster $c_i$ with the $k$-neighbours $y_1, y_2, \ldots, y_k$;
2.5        remove records $y_1, y_2, \ldots, y_k$ from dataset $X$;
2.6        set $n=n-k$;  $i=i+1$;
2.7        find the most distant record $x_s$ from $x_r$;
2.8        find $k$-nearest neighbours $y_1, y_2, \ldots, y_k$ of $x_s$;
2.9        form cluster $c_i$ with the $k$-neighbours $y_1, y_2, \ldots, y_k$;
2.10       remove records $y_1, y_2, \ldots, y_k$ from dataset $X$;
2.11       set $n=n-k$;  $i=i+1$;
2.12.  end while
3.     if ($n \geq 2k$) do
3.1        compute centroid $\bar{x}$ of remaining records in $X$;
```

```
3.2        find the most distant record $x_r$ from $\bar{x}$;
3.3        find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
3.4        form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
3.5        remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
3.6        set $n=n-k$;  $i=i+1$;
3.7     endif
4.      if ( $n>0$) then
4.1        form a cluster $c_i$ with the $n$ remaining records;
4.2        set $n=n-n$; $i=i+1$;
4.3     endif
5.      end algorithm
```

The following algorithm (*MDAV1*) is presented in [14] with minor modifications in steps 3 and 4 resulting in slightly less information loss than the *MDAV* algorithm (algorithm-1). If at least k records remain unprocessed at step 3, this algorithm forms a single group using the group formation process as in step 2. When less than k records remain at last (step 4) it assigns each of the remaining records to its closest cluster among the clusters that are already formed. Thus there may be more than one group having more than k records.

Algorithm-3 (The *MDAV1* algorithm):

```
1.      set $i=1$; $n=|X|$;
2.      while ($n\geq 2k$) do
2.1        compute centroid $\bar{x}$ of remaining records in $X$;
2.2        find the most distant record $x_r$ from $\bar{x}$;
2.3        find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
2.4        form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
2.5        remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
2.6        set $n=n-k$;  $i=i+1$;
2.7        find the most distant record $x_s$ from $x_r$;
2.8        find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_s$;
2.9        form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
2.10       remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
2.11       set $n=n-k$;  $i=i+1$;
2.12.   end while
3.      if ($n\geq k$) then
3.1        compute centroid $\bar{x}$ of remaining records in $X$;
3.2        find the most distant record $x_r$ from $\bar{x}$;
3.3        find $k$ nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
```

```
3.4          form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
3.5          remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
3.6          set $n=n-k$;  $i=i+1$;
3.7      endif
4.       if ($n>0$) then
4.1          for each remaining record $x_r$ in $X$ do
4.1.1            find the closest cluster centroid $\bar{c}_j$ from $x_r$;
4.1.2            add the records $x_r$ to cluster $c_j$;
4.1.3        end for
4.2      endif
5.       end algorithm
```

The above three algorithms construct two groups in each iteration. The authors in [8] presented the *Centroid-based Fixed-size* microaggregation method that constructs one group in each iteration. The algorithm is adaptation of the *MDAV* algorithm so that the iteration continues until there are at least k records that are unassigned (step 2). Then, each of the remaining records is assigned to its closest group. We adapt the *MDAV*-generic algorithm (algorithm-2) in the similar way. We call it *MDAV-single-group* algorithm and present it here as it forms the basis of our proposed variable-size *MDAV* algorithm.

Algorithm-4 (The *MDAV-single-group* algorithm):

```
1.       set $i=1$; $n=|X|$;
2.       while ($n \geq 3k$) do
2.1          compute centroid $\bar{x}$ of remaining records in $X$;
2.2          find the most distant record $x_r$ from $\bar{x}$;
2.3          find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
2.4          form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
2.5          remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
2.6          set $n=n-k$;  $i=i+1$;
2.7.     end while
3.       if ($n \geq 2k$) do
3.1          compute centroid $\bar{x}$ of remaining records in $X$;
3.2          find the most distant record $x_r$ from $\bar{x}$;
3.3          find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
3.4          form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
3.5          remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
3.6          set $n=n-k$;  $i=i+1$;
3.7      endif
```

```
4.        if ( $n$>0) then
4.1          form a cluster $c_i$ with the $n$ remaining records;
4.2          set $n$=$n$-$n$; $i$=$i$+1;
4.3          endif
5.        end algorithm
```

Computational cost of all the four *MDAV* algorithms presented above become $O(n^2)$ [13]. The main problem of these fixed-size algorithms is lack of flexibility. They only generate groups of fixed cardinality k causing higher information loss. *V-MDAV* (*Variable-size Maximum Distance to Average Vector*) is the first variable-size microaggregation method presented in [12, 13]. This algorithm extends the group that is currently formed up to a maximum size of $2k$-1 based on some heuristics. To extend the current group it finds the closest unassigned record, e_{min} outside the group to any record inside the group and the corresponding distance between these two records is termed d_{in}. Then, the closest unassigned record to e_{min} is found with corresponding distance being termed d_{out}. If $d_{in}<\gamma d_{out}$ then the record e_{min} is inserted in the current cluster. The extension process is repeated until the group size is equal to $2k$-1 or when a decision of inclusion is not satisfied. Here γ is a user defined constant. The determination of the best value of γ for a given dataset is not straightforward. Values of γ close to zero are effective when the data are scattered, when the dataset is clustered the best value of γ is usually close to one [12]. To save time *V-MDAV* computes the global centroid of the dataset at the beginning of the algorithm and keeps it fixed instead of recomputing it in each iteration. Each of the remaining records after termination of iterations is inserted to its closest cluster. This may cause a cluster to have number of records in excess of allowed $2k$-1 when the closest cluster already contains $2k$-1 records because of the extension process. The algorithm for building a k-partition using *V-MDAV* is as follows:

Algorithm-5 (The *V-MDAV* algorithm):

```
1.        set $i$=1; $n$=|$X$|;
2.        compute centroid $\bar{x}$ of remaining records in $X$;
3.        while ($n \geq k$) do
3.1          find the most distant record $x_r$ from $\bar{x}$ ;
3.2          find $k$-nearest neighbours $y_1,y_2,\ldots,y_k$ of $x_r$;
3.3          form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\ldots,y_k$;
3.4          remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
3.5          set $n$=$n$-$k$;  $flag$=true;
3.6          if (n==0) then set $flag$ = false;
3.7           while (|$c_i$|<$2k$-$1$ and $flag$==true)
3.7.1            find unassigned record $e_{min}$ which is the closest to any record of the cluster $c_i$ and let
                 $d_{in}$ be the distance between the two records.
3.7.2            let, $d_{out}$ be the distance from $e_{min}$ to the closest unassigned record in $X$;
3.7.3            if ($d_{in} < \gamma d_{out}$) then
3.7.3.1              assign $e_{min}$ to the current cluster $c_i$;
3.7.3.2              set $n$=$n$-$1$;
3.7.3.3              if (n==0) then set $flag$ = false;
```

```
3.7.3.4      else set $flag$=false;
3.8        end while
3.9       $i$=$i$+1;
3.10    end while
4.      if ($n$<$k$) then
4.1        for each remaining record $x_r$ in $X$ do
4.1.1         find the closest cluster centroid $\bar{c}_j$ from $x_r$;
4.1.2         add the records $x_r$ to cluster $c_j$;
4.1.3       end for
4.2     endif
5.      end algorithm
```

Computational cost of *V-MDAV* algorithm also remains O(n^2) [12]. Although the *V-MDAV* algorithm produces microaggregated datasets with lower information loss, other variable-size algorithms can be developed with better performance. In the next section we present a new variable-size microaggregation algorithm with lower information loss.

3. PROPOSED ALGORITHM

Proposed algorithm is an extension of the *MDAV-single-group* algorithm presented in the previous section (*algorithm*-4) to make it variable-size. Experimental study presented in the next section shows that for some situations the single-group algorithm performs better than double-group algorithms, but the reverse is the case for some other situations. Therefore, variable-size algorithms are more likely to become single-group algorithms as it is natural to form a group and then extend it. We have selected *MDAV-single-group* algorithm as the basis for our variable-size algorithm because it smoothly handles less than k residuals records that remain at end of termination of iterations. Also minimum of 3k unassigned records become a requirement for our algorithm to continue group formation and extension. The proposed algorithm called *MDAV2k* is presented below.

Algorithm-6 (The *MDAV2k* algorithm):

```
1.      set $i$=1; $n$=|$X$|;
2.      while ($n \geq 3k$) do
2.1        compute centroid $\bar{x}$ of remaining records in $X$;
2.2       find the most distant record $x_r$ from $\bar{x}$;
2.3       find 2$k$ nearest neighbours $y_1,y_2,\ldots,y_{2k}$ of $x_r$;
2.4       form cluster $c_i$ with first $k$-neighbours $y_1,y_2,\ldots,y_k$;
2.5       remove records $y_1,y_2,\ldots,y_k$ from dataset $X$;
2.6       set $n=n-k$; $j=k+1$;
2.7       compute centroid $\bar{x}_i$ of cluster $c_i$;
2.8       while ($j$<2$k$ and |$c_i$|<2$k$-1 ) do
```

```
2.8.1       find distance $d_1$ of record $x_r$ from $\overline{x}_i$ ;
2.8.2       find distance $d_2$ of record $y_j$ from $\overline{x}_i$ ;
2.8.3       find $k$-nearest neighbours $z_1, z_2,\dots,z_k$ of $y_j$ in $X$;
2.8.4       compute centroid $\overline{z}$ of $z_1,z_2,\dots,z_k$;
2.8.5       compute distance $d_3$ of $y_j$ from $\overline{z}$ ;
2.8.6       compute $\gamma=d_3/d_1$;
2.8.7       if $(\gamma>1.0)$ then $\gamma = 1.0+(1.0/(5.0+\gamma))$;
2.8.8       if ( $d_2< \gamma d_3$) then
2.8.8.1       insert $y_j$ in current cluster $c_i$;
2.8.8.2       recompute centroid $\overline{x}_i$ of cluster $c_i$;
2.8.8.3       remove record $y_j$ from $X$;
2.8.8.4       set $n=n-1$;
2.8.8.5     endif
2.8.9     end while
2.9       set $i=i+1$;
2.10.   end while
3.      if $(n>2k)$ then
3.1       compute centroid $\overline{x}$ of remaining records in $X$;
3.2       find the most distant record $x_r$ from $\overline{x}$ ;
3.3       find $k$ nearest neighbours $y_1,y_2,\dots,y_k$ of $x_r$;
3.4       form cluster $c_i$ with the $k$-neighbours $y_1,y_2,\dots,y_k$ ;
3.5       remove records $y_1,y_2,\dots,y_k$ from dataset $X$;
3.6       set $n=n-k$; $i=i+1$;
3.7     endif
4.      if ( $n>0$) then
4.1         form a cluster $c_i$ with the $n$ remaining records;
4.2         $i=i+1$;
4.3     endif
5.      end algorithm
```

The *MDAV2k* algorithm iterates so long as at least $3k$ records remain unassigned. In each iteration the algorithm finds $2k$ nearest neighbours, denoted by $y_1,y_2,\dots,y_{2k}$ of the farthest record x_r from the centroid $\overline{x}$ of the remaining records in dataset X. Current cluster, c_i is formed with the first k-neighbours $y_1,y_2,\dots,y_k$ of x_r. Each of the other k neighbours is tested for inclusion in the currently formed cluster by computing a heuristic. This algorithm also uses a constant γ in the heuristic but it is computed dynamically for every situation instead of being a user defined constant as in *V-MDAV*. Let, $\overline{x}_i$ be the centroid of the cluster c_i. Consider the $(k+1)$-th neighbour, y_{k+1} of x_r.

Compute distance d_1 of x_r from $\bar{x}_i$ and distance d_2 of y_{k+1} from $\bar{x}_i$. Compute centroid $\bar{z}$ of $z_1, z_2, \ldots, z_k$, the k-nearest unassigned neighbours of y_{k+1}. Find distance d_3 of y_{k+1} from $\bar{z}$. Compute $\gamma = d_3/d_1$. If $\gamma > 1.0$ then set $\gamma = 1.0 + 1.0/(5.0 + \gamma)$ to restrict the value of γ to maximum of 1.16. Restriction of γ value is required because higher value of it will favour inclusion that may cause increase of information loss rather than decreasing. Now, if $d_2 < \gamma d_3$ then insert y_{k+1} in cluster c_i and recompute the centroid of the cluster. Then the test is repeated for $y_{k+2}, \ldots, y_{2k}$. For y_{2k}, if the cluster c_i has already $2k$-1 records in it then the test should be skipped and record y_{2k} should not be inserted in the cluster c_i.

3.1. Complexity analysis

In each iterations between k and $2k$-1 records are grouped, on average $(3k-1)/2$ records. The algorithm will perform at most $2n/(3k-1)$ iterations. In each iteration, it needs to compute $2k$ nearest neighbours in the remaining records followed by extension of the cluster created k times. In each of the extension process k nearest neighbours need to be found in the remaining records of the dataset. If we assume that on average $n/2$ unassigned records remain in the dataset, complexity of the algorithm will be $O(2n/(3k-1)(2kn/2+kkn/2))$ i.e. $O(kn^2)$.

4. EXPERIMENTAL RESULTS

In this section we present experimental results performed on the proposed method. We have implemented in C++ under LINUX environment all the six microaggregation algorithms namely *MDAV*, *MDAV-generic*, *MDAV1*, *MDAV-single-group*, *V-MDAV* presented in section 2 and our proposed algorithm *MDAV2k* presented in section 3. Experiments are performed on the following three datasets proposed as reference microdata datasets during the "CASC" project [15].

- The "Tarragona" dataset contains 834 records with13 numerical attributes.
- The "Census" dataset contains 1,080 records with 13 numerical attributes.
- The "EIA" dataset contains 4,092 records with 11 numerical attributes.

Attributes of the datasets are standardized by subtracting their mean and dividing by their standard deviation, so that they have equal weights when computing distances.
The results are presented in Table 1. In order to compare the performance of the algorithms both *SSE* and *IL* measures are reported for different values of k and for each of the algorithms and for each of the three datasets.

4.1. Comparing double-group *MDAV* algorithms

It can be seen from the Table 1 that performance of the three double-group fixed-size algorithms *MDAV*, *MDAV1* and *MDAV-generic* (first three rows in the table for each dataset) are almost similar. Although the difference is in fractional parts only, it can be seen that *MDAV1* is slightly better than *MDAV* which is slightly better than *MDAV*-generic in terms of *SSE* as well as *IL* measure. Lower *SSE* and *IL* measures indicate better performance. *MDAV1* obtains lesser information loss because it distributes the less than k remaining records to more than one cluster. If number of records are integral multiple of k all the three algorithm produce exactly similar results.

4.2. Comparing double-group and single-group MDAV algorithms

Comparing any one of the double group algorithms to the single group algorithm *MDAV-single-group* (Fourth row in the table for each dataset) it can be concluded that neither of the double-group or single-group algorithm performs better than the other. Performance of any double-group algorithm is better than the single-group algorithm for the *Tarragona* dataset, but reverse is the case for *Census* dataset. Results for the *EIA* dataset shows that performance depends upon different values of *k* also.

4.3. Comparing variable-size MDAV algorithms

Fifth and sixth rows for each dataset in Table 1 present the results for variable-size algorithms *V-MDAV* and the proposed algorithm *MDAV2k*. It is clear that *MDAV2k* performs far better than *V-MDAV* producing lesser information loss. We have extended the *MDAV*-single group algorithm for developing the variable-size *MDAV2k* algorithm, so performance of the proposed algorithm should be compared to this algorithm. It can be seen from Table 1 that the proposed algorithm produces lower information loss than the *MDAV-single-group* algorithm. In fact for the *EIA* as well as *Census* datasets the *MDAV2k* algorithm shows better results than any of the presented algorithms. Variable-size algorithms show better performance for datasets with clustering tendency. This is the reason for achieving greater reduction of information loss for *EIA* dataset. *Tarragona* is a scattered dataset exhibiting no tendency for clustering for lower value of *k*, that is why reduction of information loss by proposed algorithm is very less.

For the *V-MDAV* algorithm γ value is user specified. The results for this algorithm are presented in Table 1, taking γ=0.2 for *Tarragona* and *Census* datasets and γ=1.1 for the *EIA* dataset as these two values of γ are suggested by authors of *V-MDAV* in [12]. The *MDAV2k* algorithm dynamically adjusts the value of γ.

Table 1. Experimental results.

Dataset	***Method***	***K=3*** ***SSE : (IL)***	***K=4*** ***SSE : (IL)***	***K=5*** ***SSE : (IL)***	***K=10*** ***SSE : (IL)***
Tarragona	1. *MDAV*	*1835.8318 (16.9326)*	*2119.1678 (19.545)*	*2435.2796 (22.461)5*	*3598.7743 (33.1929)*
	2. *MDAV1*	*1835.8318 (16.9326)*	*2119.1549 (19.5458)*	*2435.2534 (22.4613)*	*3598.7173 (33.1924)*
	3. *MDAV-gen*	*1835.8318 (16.9326)*	*2119.1740 (19.5460)*	*2435.3160 (22.4619)*	*3598.7743 (33.1929)*
	4. *MDAV-single*	*1839.4617 (16.9661)*	*2139.1554 (19.7303)*	*2473.9951 (22.8186)*	*3601.2138 (33.2154)*
	5. *VMDAV*	*1839.6440 (16.9678)*	*2135.5903 (19.6974)*	*2481.3201 (22.8862)*	*3607.2572 (33.2711)*
	6. *MDAV2k*	*1839.4617 (16.9661)*	*2139.1497 (19.7302)*	*2418.5713 (22.3074)*	*3600.4316 (33.2082)*
Census	1. *MDAV*	*799.1827 (5.6922)*	*1052.2557 (7.4947)*	1276.0162 (9.0884)	*1987.4925 (14.1559)*
	2. *MDAV1*	*799.1827 (5.6922)*	*1052.2557 (7.4947)*	1276.0162 (9.0884)	*1987.4925 (14.1559)*

	3. *MDAV-gen*	*799.1827 (5.6922)*	*1052.2557 (7.4947)*	*1276.0162 (9.0884)*	*1987.4925 (14.1559)*
	4. *MDAV-single*	*793.7595 (5.6536)*	*1044.7749 (7.4414)*	*1247.3171 (8.8840)*	*1966.5216 (14.0066)*
	5. *VMDAV*	*794.9373 (5.6619)*	*1054.9675 (7.5140)*	*1264.5801 (9.0070)*	*1975.8520 (14.0730)*
	6. *MDAV2k*	*791.5291 (5.6377)*	*1037.6860 (7.3909)*	*1243.5027 (8.8569)*	*1957.0561 (13.9391)*
EIA	1. *MDAV*	*217.3804 (0.4829)*	*302.1859 (0.6713)*	*750.1957 (1.6667)*	*1728.3120 (3.8397)*
	2. *MDAV1*	*217.3804 (0.4829)*	*302.1859 (0.6713)*	*750.1957 (1.6667)*	*1728.309 (3.8397)*
	3. *MDAV-gen*	*217.3804 (0.4829)*	*302.1859 (0.6713)*	*750.2037 (1.6667)*	*1728.3120 (3.8397)*
	4. *MDAV-single*	*215.1095 (0.4779)*	*301.9676 (0.6709)*	*783.0258 (1.7396)*	*1580.8008 (3.5120)*
	5. *VMDAV*	*229.2986 (0.5094)*	*437.8020 (0.9726)*	*588.0341 (1.3064)*	*1264.4328 (2.8091)*
	6. *MDAV2k*	*191.6008 (0.4257)*	*289.4685 (0.6431)*	*405.1972 (0.9002)*	*1188.4501 (2.6403)*

The experimental results presented here show that proposed *MDAV2k* is a good algorithm producing microaggregated datasets with lower information loss.

5. Conclusion

In this paper we have made an experimental study on several variants of the *MDAV* microaggregation technique for privacy preservation. Performances of the double-group fixed-size algorithms are found to be almost similar. Performance of single-group and double group algorithms depend on datasets as well as value of *k*, so it cannot be judged which is better. Variable-size algorithms outperform fixed-size algorithms.

We also proposed an inproved variable-size *MDAV* algorithm that produce lower information loss with little increase in computational complexity (O(kn^2)). Fixed-size algorithms have complexity O(n^2). This is acceptable as *k* is usually a small integer.

Proposed algorithm is a modification of the *MDAV* algorithm to make it variable-size. The algorithm computes 2*k* nearest neighbours of the farthest record from the centroid of the remaining unassigned records in the dataset. First *k* of the 2*k* neighbours form a cluster and it is extended up to a size of 2*k*-1 records by including some of the remaining *k* neighbours based on a heuristic. The *V-MDAV* algorithm requires a user defined global constant γ to be used for the cluster extension process. In our algorithm the value of γ is dynamically computed for each locality. Experimental results on standard datasets show better performance of this algorithm in comparison to other existing algorithms.

To form a single cluster 2*k* nearest neighbours of the currently selected record for cluster formation is considered. It is possible to consider 3*k* neighbours instead of 2*k* as the algorithm iterates so long as there are at least 3*k* neighbours yet to be assigned to any cluster. This will increase computation time slightly while producing better results as more records are considered for inclusion in the cluster extension. Another possibility for modification of the algorithm is to form the current cluster by finding *k* nearest neighbours of the present record considered and then test *k* or 2*k* nearest neighbours of the centroid of the current cluster for inclusion in the extension.

ACKNOWLEDGEMENTS

This paper is an outcome of AICTE funded project under Research Promotion Scheme (RPS).

REFERENCES

[1] J. Domingo-Ferrer & V. Torra, (2001) "A quantitative comparison of disclosure control methods for microdata", in: P. Doyle, J. Lane, J. Theeuwes, L. Zayatz (Eds.), Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, Elsevier Science, pp. 111-133.

[2] C. Aggarwal & P. Yu, (2008) "Privacy-Preserving Data Mining: Models and Algorithms (Advances in Database Systems)", Springer Science and Business Media L.L.C.: Berlin, Heidelberg.

[3] E. Fayyoumi & J. Oommen, (2010) "A Survey on statistical disclosure control and micro-aggregation techniques for secure statistical databases", Software - Practice and Experience, 2010:40, pp. 1161-1188.

[4] A. Oganian & J. Domingo-Ferrer, (2000) "On the complexity of optimal microaggregation for statistical disclosure control", Statistical Journal of United Nations Economic Commission for Europe, 18 (4), pp. 345-354.

[5] J. Domingo-Ferrer & J. Mateo-Sanz, (2002) "Practical data-oriented microaggregation for statistical disclosure control", IEEE Transactions on Knowledge and Data Engineering, 14(1), pp. 189-201.

[6] A. Hundepool, A. V. deWetering, R. Ramaswamy, L. Franconi, A. Capobianchi, P.-P. DeWolf, J. Domingo-Ferrer, V. Torra, R. Brand & S. Giessing, (2003) "μ-ARGUS version 3.2 Software and User's Manual", Voorburg NL: Statistics Netherlands, http://neon.vb.cbs.nl/casc.

[7] J. Domingo-Ferrer & V. Torra, (2005) "Ordinal, continuous and heterogenerous k-anonymity through microaggregation", Data Mining Knowl. Discov. 11(2), pp. 195-212.

[8] M. Laszlo & S. Mukherjee, (2005) "Minimum spanning tree partitioning algorithm for microaggregation", IEEE Transactions on Knowledge and Data Engineering, 17(7), pp. 902-911.

[9] J. Domingo-Ferrer, A. Martínez-Ballesté, J.M. Mateo-Sanz & F. Sebé, (2006) "Efficient multivariate data-oriented microaggregation", The VLDB Journal 15, pp. 355-369.

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

[11] J. Nin, J. Herranz & V. Torra, (2008) "On the Disclosure Risk of Multivariate Microaggregation", Data and Knowledge Engineering (DKE), Elsevier, volume 67, issue 3, pp. 399-412.

[12] A. Solanas & A. Mart´ınez-Ballest´e, (2006) "V-MDAV: A multivariate microaggregation with variable group size", Seventh COMPSTAT Symposium of the IASC, Rome.

[13] J. Domingo-Ferrer, A. Solanas & A. Mat´_nez-Ballest´e, 2006 "Privacy in statistical databases: k-anonymity through microaggregation", in IEEE Granular Computing' 06. Atlanta. USA, pp. 774-777.

[14] Jun-Lin Lin, Tsung-Hsien Wen, Jui-Chien Hsieh & Pei-Chann Chang, (2010) "Density-based microaggregation for statistical disclosure control", Expert Systems with Application, 37(4), pp. 3256-3263.

[15] R. Brand, J. Domingo-Ferrer & J. M. Mateo-Sanz, (2002) "Reference data sets to test and compare sdc methods for protection of numerical microdata", European Project IST-2000-25069 CASC, http://neon.vb.cbs.nl/casc.

Authors

Sarat Kumar Chet**tri** is an Assistant Professor in the Department of Computer Science, Saint Mary's College, Shillong, Meghalaya, India. He is currently pursuing Ph.D. degree in computer science from the Department of Computer Science and Engineering, Tezpur University, Tezpur, India. His research interests include database technology, data mining and knowledge discovery, machine learning and data privacy preservation.

Dr. B. Borah is currently an Associate Professor of Computer Science and Engineering at Tezpur University, Tezpur, India. He received his MS degree in Systems and Information from Birla Institute of Technology and Science, Pilani, India in 1996. He got his Ph. D. degree in Data mining from Tezpur University in 2008. He also guides Ph. D. research scholars at Tezpur University. His research interests include Data mining, Pattern Recognition and Image Analysis.