A Comparative Study of Microaggregation Methods

Josep M. Mateo-Sanz and Josep Domingo-Ferrer

Universitat Rovira i Virgili, Departament d'Enginyeria Informàtica, Autovia de Salou s/n, E-43006 Tarragona, Catalonia, e-mail {jmateo,jdomingo}@etse.urv.es [*]

Abstract

Microaggregation is a statistical disclosure control technique for microdata. Raw microdata (*i. e.* individual records) are grouped into small aggregates prior to publication. Each aggregate should contain at least k records to prevent disclosure of individual information. Fixed-size microaggregation consists of taking fixed-size microaggregates (size k). Data-oriented microaggregation (with variable group size) was introduced recently. Regardless of the group size, microaggregates on a multidimensional data set can be formed using univariate techniques on projected data or using multivariate techniques. This paper presents the first method for multivariate fixed-size microaggregation. In addition, a real data set is used to compare the information loss and output data quality of fixed-size vs. data-oriented, and univariate vs. multivariate microaggregation.

Keywords: Statistical disclosure control; Microaggregation; Data-oriented microaggregation; Microdata protection.

AMS Subject Classification: 62H30, 92G30, 68PXX

1 Introduction

A *microdata set* is a set of records containing data of individuals being studied, who can be persons, companies, etc. The individual records of a microdata set are stored in a *microdata file.* Each individual j is assigned a *data vector* V_j, also called data record or data set. A data vector is formed by several variables.

Microaggregation is a family of statistical disclosure control techniques for microdata which belong to the data modification category. The rationale behind microaggregation is that confidentiality rules in use allow publication of

[*] This work is partly supported by the Spanish CICYT under grant no. TIC98-0699-C02-02

microdata sets if the data vectors correspond to groups of k or more individuals, where no individual dominates (*i. e.* contributes too much to) the group and k is a threshold value. Strict application of such confidentiality rules leads to replacing individual values with values computed on small aggregates (microaggregates) prior to publication. This is the basic principle of microaggregation.

To obtain microaggregates in a microdata set with n data vectors, these are combined to form g groups of size at least k. For each variable, the average value over each group is computed and is used to replace each of the original averaged values. Groups are formed using a criterion of maximal similarity. Once the procedure has been completed, the resulting (modified) data vectors can be published.

The partition problem implicit in microaggregation differs from the classical clustering problem whose goal is to split a population into a fixed number of disjoint groups (Hartigan, 1975), regardless of the group size. Partitions resulting from microaggregation cannot consist of groups of size smaller than k; call such partitions k-partitions. To solve the k-partition problem, a measure of similarity between data vectors is needed. Each individual data vector can be viewed as a point and the whole microdata set as a set of multidimensional points. The dimension is the number of variables in data vectors. If data vectors are characterized as points, similarity between them can be measured using a distance.

To be more specific, consider a microdata set with p continuous variables and n data vectors (*i. e.* the result of observing p variables on n individuals). A particular data vector can be viewed as an instance of $\mathbf{X}' = (X_1, \cdots, X_p)$ where the X_i are the variables. With these individuals g groups are formed with n_i individuals in the i-th group ($n_i \geq k$ and $n = \sum_{i=1}^{g} n_i$). Denote by $\mathbf{x}_{ij}$ the j-th data vector in the i-th group; denote by $\bar{\mathbf{x}}_i$ the average data vector over the i-th group, and by $\bar{\mathbf{x}}$ the average data vector over the whole set of n individuals.

The within-groups sum of squares SSE is defined as

$$SSE = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)'(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)$$

The between-groups sum of squares SSA is

$$SSA = \sum_{i=1}^{g} n_i (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})'(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})$$

The total sum of squares is $SST = SSA + SSE$ or explicitly

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

The optimal k-partition is the one that minimizes SSE (or equivalently, maximizes SSA); sums of squares are usual to measure information loss (Gordon and Henderson, 1977). A measure L of information loss standardized between 0 and 1 can be obtained from

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

The rest of this paper is organized as follows. In Section 2, previous work on univariate microaggregation is reviewed (both with fixed-size groups and variable-size groups). Section 3 deals with multivariate microaggregation techniques; a new concept of multivariate fixed-size microaggregation is presented and a method to implement it is specified; also recent work of these authors on multivariate data-oriented microaggregation is briefly recalled. In Section 4, the information loss and output data quality of microaggregation methods are compared based on a real data set; four families of methods are considered: univariate fixed-size, univariate data-oriented, multivariate fixed-size and multivariate data-oriented. Finally, Section 5 is a conclusion and a sketch of future research.

2 Univariate microaggregation

Defays and Nanopoulos (1993) proposed a mathematical algorithm to find an optimal solution for the k-partition problem (minimizing the information loss L). The idea is to choose a suitable set of hyperplanes separating the n data vectors into a number of homogeneous groups. As pointed out by its authors, the proposed algorithm is pretty complicated and difficult to implement in practice. As an alternative, the same paper presents some of the practical alternatives described in Subsection 2.1

2.1 Univariate fixed-size microaggregation

Practical heuristic microaggregation methods were proposed in Defays and Nanopoulos (1993), in Anwar (1993) and in Defays and Anwar (1995). The partition mechanism is the same in all such methods: first, data vectors are sorted in ascending or descending order according to some criterion. Then groups of successive k vectors are combined. Inside each group, the effect for each variable is to replace the k values taken by the variable with their average. If the total number of data vectors n is not a multiple of k, the last group will contain more than k data vectors.

Instead of using a multidimensional distance to sort data vectors, all practical methods quoted above perform straightforward one-dimensional sorting (this fact explains why such methods are called univariate). Two main approaches exist: single-axis sorting and individual sorting.

Single-axis sorting methods are good if all variables are highly correlated. If a particular variable is used for sorting, this variable must reflect somehow the size of the data vector. Vectors are sorted in ascending or descending order by the sorting variable, and then groups of k successive vectors are formed. Inside each group and for each variable, values are replaced by the group average. A natural improvement is to sort data vectors by the first principal component of the microdata set rather than by a particular variable. Principal components are transformed variables such that the first principal component is highly correlated with most original variables. An alternative that, like principal components, also takes all variables into account is based on the sum of z-scores: all variables are standardized and, for each data vector, the standardized values of all variables are added. Vectors are subsequently sorted by their sum of z-scores.

If the individual sorting option is chosen, then each variable is considered independently. Data vectors are sorted by the first variable, then groups of k successive values of the first variable are formed and, inside each group, values are replaced by the group average. A similar procedure is repeated for the rest of variables. Individual sorting usually preserves more information than single-axis sorting, but has a higher disclosure risk. Indeed, with individual sorting any intruder knows that the real value of a variable in a data vector in the i-th group is between the average of the $i-1$-th group and the average of the $i+1$-th group; if these two averages are very close to each another, then a very narrow interval for the real value being searched has been determined. Individual sorting also has a conceptual drawback: it does not partition the n data vectors in the microdata set on a data vector basis; instead, microaggregation is done for each variable in turn so that a different partition is obtained for each variable in the microdata set.

2.2 Data-oriented univariate microaggregation

In Domingo and Mateo (1998), microaggregation using variable-sized groups depending on data (data-oriented microaggregation) is introduced in the univariate case. The idea is that groups need not consist of exactly k data vectors, but of *at least* k data vectors. Methods yielding variable-sized groups are a bit more complex than fixed-size microaggregation (Defays and Nanopoulos, 1993) but they may take advantage from the distribution of original data to obtain a smaller information loss in comparison with fixed-size microaggregation. Figure 1 is a simple graphical example that illustrates the advantages of variable-sized groups. The figure shows two variables and nine data. If fixed-size microaggregation with $k = 3$ is used, we obtain a partition of the data into three groups, which looks rather unnatural for the data distribution given. On the other hand, if variable-sized groups are allowed then the five data on the left can be kept in a single group and the four data on the right in another group; such a variable-size grouping achieves a smaller information loss.

As discussed above, deterministically finding an optimal solution to the k-partition problem is very difficult. Heuristic methods are the only practical alternative and they should attempt to minimize the information loss L specified by expression (1). Since SST is fixed for a given data set, one should attempt

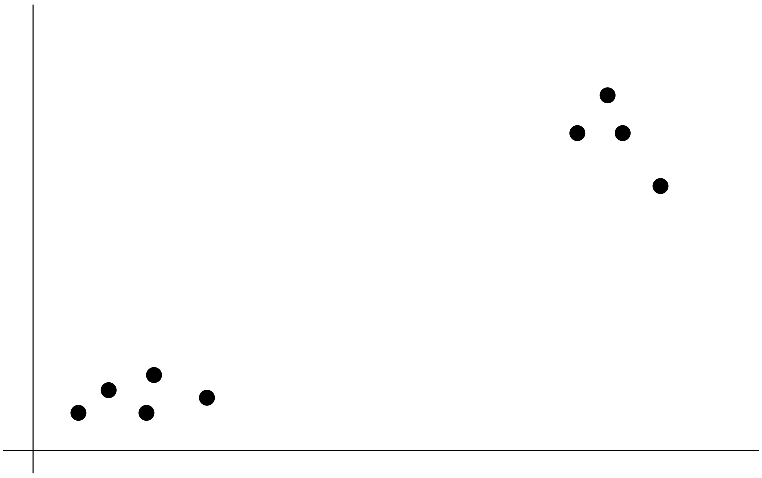

Figure 1: Variable-sized groups versus fixed-sized groups

to find a grouping that minimizes SSE.

In Domingo and Mateo (1998) two alternative heuristic approaches to variable-size univariate microaggregation are presented:

- Microaggregation based on genetic algorithms (GA).
- Modified Ward's Algorithm (k-Ward).

Being univariate, both approaches above must be combined with single-axis or individual sorting (described in Section 1) to deal with a multivariate microdata set.

Genetic microaggregation represents k-partitions as binary strings (also called chromosomes) and combines directed and random search to locate global optima.

Hierarchical classification methods can also be used as building blocks for heuristic microaggregation methods yielding variable-sized groups. Ward's method (Ward, 1963) is attractive because it is stepwise optimal: the two groups or data elements joined at each step are chosen so that the increase in the within-groups sum of squares SSE caused by their union is minimal. However, Ward's method must be adapted to be useful for microaggregation (k-Ward algorithm). The standard method just builds up a grouping hierarchy, whereas a k-partition of the initial data set is desired. As will be shown in this paper, k-Ward can be naturally turned into a truly multivariate microaggregation method.

k-Ward is a microaggregation method for quantitative data or for qualitative data where a distance has been defined. In what follows, k-Ward will be briefly recalled. The following definitions and results are needed:

Definition 1 *For a given data set, a k-partition P is any partition of the data set such that each group in P consists of at least k elements.*

Definition 2 *For a given data set, k-partition P is said to be finer than k-partition P' if every group in P is contained by a group in P'.*

It is straightforward to check that "finer than" is a partial order relationship on the set of k-partitions of a given data set.

Definition 3 *For a given data set, a k-partition P is said to be minimal with respect to the relationship "finer than" if there is no k-partition $P' \neq P$ such that P' is finer than P.*

Proposition 1 *For a given data set, k-partition P is minimal with respect to the relationship "finer than" if and only if it consists of groups with sizes $\geq k$ and $< 2k$.*

Corollary 1 *An optimal solution to the k-partition problem of a set of data exists that is minimal with respect to the relationship "finer than".*

The proofs of the above results can be found in Domingo and Mateo (1998). Now, Ward's hierarchical classification method can be modified to provide a solution that belongs to the set of candidate optimal solutions characterized by Proposition 1 and Corollary 1. Modified Ward's algorithm (k-Ward) is as follows:

Algorithm 1 (k-Ward)

1. *Form a group with the first (smallest) k elements of the data set and another group with the last (largest) k elements of the data set.*
2. *Use Ward's method until all elements in the data set belong to a group containing k or more data elements; in the process of forming groups by Ward's method, never join two groups which have both a size greater than or equal to k.*
3. *For each group in the final partition that contains $2k$ or more data elements, apply this algorithm recursively (the data set to be considered is now restricted to the particular group containing $2k$ or more elements).*

The following property of the above algorithm is proven in Domingo and Mateo (1998); its proof is recalled here because it helps understanding the design of the algorithm.

Property 1 (Convergence) *Algorithm 1 ends after a finite number of recursion steps.*

Proof: By Step 1 of the above algorithm each new recursion step starts splitting the initial data group into at least two groups; the rule in Step 2 ensures that the group formed by the smallest elements and the group formed by the largest elements are never joined thereafter (because of their sizes). In this way, at the end of a recursion step, the final k-partition consists of at least two groups and is therefore finer than the initial k-partition (consisting of a single group). If there is a group of size $\geq 2k$, then the algorithm is recursively applied to it and strictly smaller groups will be obtained (according to the previous argument). Thus after a finite number of recursion steps a k-partition of the initial data set will be obtained such that the maximal group size is less than $2k$. □

As explained above, Ward's algorithm is stepwise optimal in what regards information loss. Stepwise optimality does no longer hold for k-Ward, but a good behaviour is expected given that k-Ward is built on top of Ward's method. See Section 4 for computational results.

3 Multivariate microaggregation

We present in this section the multivariate counterparts of the methods described above. We define multivariate microaggregation to be microaggregation on multivariate *unprojected* data. Of course, if multivariate data are one-dimensionally projected using single-axis or individual sorting (see Section 1), the resulting projected data can be microaggregated with the univariate methods described in Section 2. However, single-axis sorting is a rather coarse technique and individual sorting has a higher disclosure risk and does not really perform microaggregation on a data vector basis.

Work presented in Subsection 3.1 on multivariate fixed-size microaggregation is new. In Section 3.2, previous work of these authors on data-oriented multivariate microaggregation is recalled (Mateo and Domingo, 1998; Mateo, 1998).

3.1 Multivariate fixed-size microaggregation: a new proposal

We introduce in this subsection a new family of microaggregation methods. The idea is to form groups of size k *without* projecting multivariate data in one-dimension. Instead, a multivariate distance is used. The basic algorithm can be described as follows:

Algorithm 2 (Multivariate fixed-size microaggregation)

1. *Form one group with the "first" k data vectors and another group with the "last" k data vectors.*

2. *If there are at least $2k$ data vectors which do not belong to the two groups formed in Step 1, go to Step 1 taking as new data set the previous data set minus the groups formed in the previous instance of Step 1.*

3. *If there are between k and $2k-1$ data vectors which do not belong to the two groups formed in Step 1, form a new group with those elements and exit the Algorithm.*

4. *If there are less than k data vectors which do not belong to the groups formed in Step 1, add them to the closest group formed in Step 1.*

The problem remains of how to decide which are the "first" and "last" data vectors in Step 1 of the above algorithm. If single-axis sorting is used to make that decision, Algorithm 2 is equivalent to performing univariate fixed-size microaggregation on projected multivariate data. A truly multivariate criterion is as follows. Define as extreme data vectors the two vectors in the data set which are most distant according to the distance matrix; then, for each of the extreme data vectors, take the $k-1$ data vectors closest to it following the distance matrix; in this way, a group with the "first" k data vectors and another group with the "last" data vectors are obtained. This criterion for choosing the "first" and the "last" data vectors will be called maximum-distance (MD) criterion. The grouping resulting from MD may depend on which extreme data vector one starts with, *i. e.* which extreme data vector is taken as the "first" data vector. For example, consider the six two-dimensional data vectors in Figure 2 and take $k=3$. The two most distant vectors are labeled 2 and 5. Starting from vector 2, the closest vector is vector 1; now the vector closest to the group (1,2) is vector 3. So starting from vector 2, we get the groups (1,2,3) and (4,5,6). But if we choose to start from the other extreme point (vector 5), the closest vector is vector 4; now the vector closest to the group (4,5) is vector 3. So starting from vector 5, we get the groups (3,4,5) and (1,2,6). Anyway, the differences in the information loss resulting from choosing either extreme vector as "first" or "last" are small (see results in Subsection 4.2). Furthermore, the larger the data set, the less likely is the kind of pathological situation depicted in Figure 2.

3.2 Multivariate data-oriented microaggregation

A natural way to obtain multivariate data-oriented methods is to generalize some of the univariate data-oriented methods quoted in Subsection 2.2. Genetic microaggregation methods are not so easy to adapt for dealing with unprojected multivariate data: the main problem comes from the fact that a multidimensional space is only partially ordered, which makes properly representing multivariate k-partitions as binary strings far from obvious.

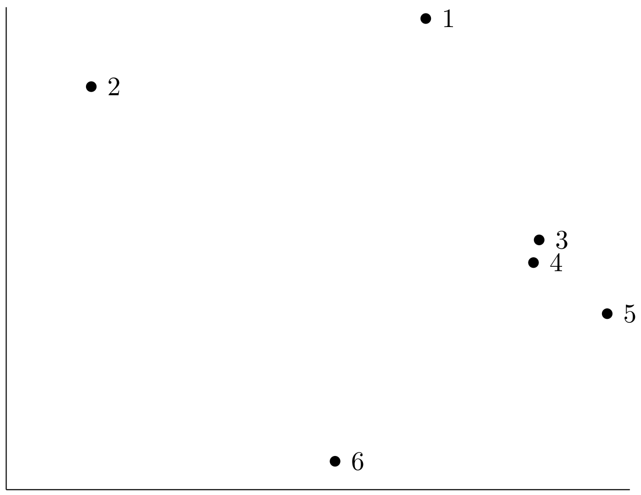

Figure 2: Grouping with the maximum-distance criterion

Luckily enough, the strong point of k-Ward is that it can be easily adapted into a multivariate k-Ward algorithm to directly work with multidimensional (unprojected) data vectors. The reason is that the underlying Ward's method was actually designed as a multivariate clustering algorithm. Thus to obtain a multivariate version of k-Ward, only Step 1 of Algorithm 1 needs to be adapted. Basically, what is needed is a multivariate sorting criterion specifying what is meant by the "first" k data vectors and the "last" k data vectors.

Unlike for multivariate fixed-size microaggregation, it makes sense to use single-axis sorting to determine which are the first and last k vectors in Step 1 of Algorithm 1. The resulting algorithm is *not* equivalent to data-oriented univariate microaggregation on projected data. The multivariate data vectors can be ranked according to their first principal component, their sum of z-scores or a particular variable. The maximum-distance criterion may also be used as an alternative to avoid one-dimensional projection, and then the grouping depends on which extreme vector is taken as first.

4 Computational results

The performance of the multi-dimensional microaggregation methods discussed in this work has been compared using a data set of 834 companies in the Tarragona area for which 13 variables have been collected: fixed assets (V1), current assets (V2), treasury (V3), uncommitted funds (V4), paid-up capital (V5), short-term debt (V6), sales (V7), labour costs (V8), depreciation (V9), operating profit (V10), financial outcome (V11), gross profit (V12) and net profit (V13). The data set corresponds to year 1995.

The methods considered in the comparison include univariate fixed-size microaggregation (denoted by UFS), univariate data-oriented microaggregation using k-Ward (denoted by UDO), multivariate fixed-size microaggregation (MFS), and multivariate data-oriented microaggregation using k-Ward (denoted by MDO). For UFS and UDO (step 1 of k-Ward) several sorting criteria have been considered: a particular variable (PV), sum of z-scores (SZ) and first principal component (FPC). For MFS, only truly multivariate sorting criteria such as maximum-distance (MD) make sense (see Subsection 3.1). For MDO (step 1 of multivariate k-Ward), the PV, SZ, FPC and MD criteria have been considered. In the rest of this section, the sorting criterion appears as a subscript of the microaggregation method. When using the PV criterion, a range of results is obtained depending on which particular variable is used; if a single result is reported in what follows, it corresponds to the *best variable*.

4.1 Computing time

The whole data set of 834 data vectors was microaggregated on a Pentium MMX at 166MHz running under the Linux operating system. The following results for the microaggregation time (excluding sorting time) were obtained:

- With UFS, the computing time is negligible for any $3 \leq k \leq 5$ regardless of the sorting criterion used.
- With UDO, the computing time is about 25 seconds, regardless of the group size and the sorting criterion used.
- With MFS, the computing times is also about 25 seconds.
- With MDO, the computing time is between 35 and 39 seconds.

The above results exhibit no significant differences. It should be noted here microaggregation is usually performed off-line and even a few hundred seconds of computing time would be acceptable. Thus, it can be concluded that the computing time is not an issue for either method, even if MDO turns out to be somewhat slower than the rest of methods. The really interesting comparison is in terms of information loss and data quality.

4.2 Information loss and data quality

To compare the information loss caused by multivariate fixed-size microaggregation and k-Ward, we have considered the loss L (see expression (1)). It must

Method	$100L$ ($k=3$)	$100L$ ($k=4$)	$100L$ ($k=5$)
UFS$_{PV}$	30.11 - 48.48	34.14 - 57.0	37.59 - 60.83
UFS$_{SZ}$	28.92	32.15 or 32.08	35.20 or 32.56
UFS$_{FPC}$	23.87 or 23.89	30.62 or 25.99	33.29 or 30.74
UDO$_{PV}$	31.17 - 53.93	35.28 - 58.23	39.06 - 61.18
UDO$_{SZ}$	30.13	32.56	34.65
UDO$_{FPC}$	25.35	31.71	32.08
MFS$_{MD}$	15.60	19.27	22.67
MDO$_{PV}$	16.23 - 21.61	20.46 - 29.45	22.38 - 31.77
MDO$_{SZ}$	19.16	24.31	27.6
MDO$_{FPC}$	15.87	21.58	23.69
MDO$_{MD}$	16.01 - 16.75	21.13 - 21.24	21.83 - 22.77

Table 1: Comparison of percentage information loss

be pointed out here that the value of L depends on the units used for the variables in the microdata set. Such an undesirable property can be neutralized if all variables are standardized prior to microaggregation: if variable V_i takes a value x, then x is replaced by $(x-\bar{v}_i)/s_{v_i}$, where $\bar{v}_i$ and s_{v_i} are, respectively, the average and the standard deviation of the values taken by V_i. Results presented throughout this section have been obtained on standardized variables. Table 1 shows the percentage values of L obtained for UFS, UDO, MFS and MDO.

For the FPC and SZ sorting criteria, two losses are given (unless both are very similar). The first one is obtained when data are sorted in ascending order following the criterion; the second loss is obtained when data are sorted in descending order.

A range of losses is given for the PV criterion. With this criterion, the information loss depends on the particular variable chosen for sorting. Thus the lower limit of the range corresponds to the best variable (leading to a minimal loss) and the upper limit to the worst variable. It can be seen that the range for MDO is narrower than for UDO. In this sense, multivariate data-oriented methods are more *robust* than their univariate counterparts.

For the MDO method with MD criterion, a range is also given. The reason is that each time Step 1 of the multivariate k-Ward method is run, one must decide which of two extreme data vectors is the “first” one and which is the “last” one. Thus, if Step 1 is run m times, one could in principle obtain as many as 2^m different losses. However, it can be seen that the MD criterion is pretty robust in that the difference between the worst and best losses is very small.

Table 2 compares average standard deviations of all variables under each method and for three group sizes. Original data have been standardized, so

Method	$k = 3$	$k = 4$	$k = 5$
UFS$_{PV}$	.83	.80	.78
UFS$_{SZ}$	.84	.82	.81
UFS$_{FPC}$	.87	.86	.83
UDO$_{PV}$	.83	.80	.77
UDO$_{SZ}$	.83	.81	.80
UDO$_{FPC}$	.86	.82	.82
MFS$_{MD}$	.92	.90	.88
MDO$_{PV}$	.91	.89	.88
MDO$_{SZ}$	.90	.87	.85
MDO$_{FPC}$	.92	.88	.87
MDO$_{MD}$	.92	.89	.88

Table 2: Comparison of average standard deviations

they have standard deviation 1; microaggregated data cannot contain more information, *i. e.* more variability, so standard deviations for variables are less than or equal to 1. The closer the average standard deviation to 1, the better is a method. It can be seen that multivariate methods perform better than univariate methods; a closer look reveals that MFS is slightly better than MDO.

For each method and for each group size k, Table 3 reflects the impact of the various microaggregation methods on the first principal component. The table gives:

ΔR: Percentage change in the average correlation of variables with the first principal component. Using the original data set, the percentage change is 0; therefore, the smaller this figure, the better is a method.

ΔW: Percentage change in the average weight of variables on the first principal component. The smaller this figure, the better is a method.

%FPC: Percentage proportion of variability of the whole microaggregated data set explained by the first principal component. In the original data set, the first component explains 63.4% of the total variability. The more similar the percentage explained to 63.4, the better is a method.

Table 4 summarizes the impact of the various microaggregation methods on the correlations between variables. With 13 variables, the number of unordered variable pairs is $13!/(11!2!) = 78$. Let r_{ij} be the linear correlation coefficient between variables i and j for original data; let r_{ij}^m be the correlation coefficient between the same variables once data have been microaggregated using method m. For each method m, Table 4 gives the average $\overline{\Delta r}$ and the standard deviation

Method	$k = 3$			$k = 4$			$k = 5$		
	ΔR	ΔW	%FPC	ΔR	ΔW	%FPC	ΔR	ΔW	%FPC
UFS$_{PV}$	15.8	11.7	81.4	20.2	15.7	85.8	22.0	17.0	87.8
UFS$_{SZ}$	18.8	14.5	84.3	21.3	15.0	88.0	22.6	16.3	89.4
UFS$_{FPC}$	16.3	13.1	81.3	17.1	13.4	82.9	21.7	16.1	88.3
UDO$_{PV}$	16.6	12.3	82.4	21.3	16.6	86.9	23.7	18.4	89.8
UDO$_{SZ}$	19.8	14.1	86.0	22.1	15.3	88.9	24.4	17.5	91.3
UDO$_{FPC}$	17.7	14.1	83.0	23.2	16.6	90.0	23.6	17.1	90.4
MFS$_{MD}$	7.7	7.1	71.9	9.1	8.1	73.7	9.6	8.0	74.3
MDO$_{PV}$	8.2	8.0	72.6	10.4	9.5	75.1	10.6	9.0	75.5
MDO$_{SZ}$	10.1	9.1	74.7	13.7	10.6	79.2	14.7	11.5	80.1
MDO$_{FPC}$	8.2	8.0	72.6	11.4	9.9	76.7	13.2	10.9	78.6
MDO$_{MD}$	7.8	7.9	72.1	10.7	9.5	75.8	10.5	9.0	75.3

Table 3: Impact of microaggregation on the FPC

$s_{\Delta r}$ of the 78 discrepancies $|r_{ij}^m - r_{ij}|$. The smaller the average discrepancy and the smaller the discrepancy variability, the better is a method.

5 Conclusion and future research

The results shown in Tables 1 through 4 for the microdata set tested can be summarized as follows:

- Multivariate methods behave significantly better than univariate methods. The reason is that in univariate microaggregation there are two sources of information loss: one-dimensional data projection and microaggregation. In multivariate microaggregation, the only source of information loss is microaggregation itself.
- Among univariate methods, UFS$_{FPC}$ is slightly better than the rest for this data set.
- Among multivariate methods, the new method MFS$_{MD}$ is better than the rest for this data set.
- The MD sorting criterion seems to be the best one for multivariate microaggregation. It is very robust, gives best results and requires little computation.

An interesting line of future research would be to repeat the comparative study performed in this paper for a very large data set. Some changes in the implementation of MFS and MDO may be necessary to deal with very large

Method	$k = 3$		$k = 4$		$k = 5$	
	$\overline{\Delta r}$	$s_{\Delta r}$	$\overline{\Delta r}$	$s_{\Delta r}$	$\overline{\Delta r}$	$s_{\Delta r}$
UFS$_{PV}$	.20	.08	.26	.12	.28	.13
UFS$_{SZ}$	.24	.10	.28	.11	.29	.12
UFS$_{FPC}$	.20	.09	.22	.10	.28	.11
UDO$_{PV}$	.21	.09	.27	.13	.30	.15
UDO$_{SZ}$	.26	.10	.19	.12	.32	.13
UDO$_{FPC}$	.22	.10	.30	.12	.31	.13
MFS$_{MD}$	.10	.05	.12	.06	.12	.06
MDO$_{PV}$	.10	.06	.13	.07	.14	.07
MDO$_{SZ}$	.13	.06	.18	.07	.19	.08
MDO$_{FPC}$	.10	.06	.15	.07	.17	.07
MDO$_{MD}$	.10	.05	.14	.06	.13	.06

Table 4: Impact of microaggregation methods on correlations

multivariate data sets: the distance matrix between data vectors cannot be prestored and thus distances between data vectors must be computed when they are needed. This introduces a considerable overhead and may be a computing time penalty for MFS. The reason is that MFS normally tends to create more groups than MDO, and thus requires more distance computations to complete the microaggregation process.

Other research topics include the development of alternative heuristics for multivariate microaggregation. For example, one could think of adapting genetic algorithms to deal with unprojected multivariate data.

Acknowledgments

We thank the Registre Mercantil de Tarragona for providing the data set used to carry out the tests described in Section 4.

References

Anwar, N. (1993). *Micro-Aggregation - The Small Aggregates Method,* internal report, Luxembourg: Eurostat.

Defays, D., and Nanopoulos, P. (1993). “Panels of enterprises and confidentiality: the small aggregates method”, in *Proceedings of the 1992 Symposium on Design and Analysis of Longitudinal Surveys,* Ottawa: Statistics Canada, 195-204.

Defays, D., and Anwar, N. (1995). "Micro-aggregation: a generic method", in *Proceedings of the 2nd International Symposium on Statistical Confidentiality,* Luxembourg: Eurostat, 69-78.

Domingo-Ferrer, J., and Mateo-Sanz, J.M. (1998). *Practical Data-Oriented Microaggregation for Statistical Disclosure Control,* Research Report DEI-RR-98-005, Department of Computer Science, Tarragona: Universitat Rovira i Virgili.

Gordon, A. D., and Henderson, J. T. (1977). "An algorithm for Euclidean sum of squares classification", *Biometrics,* **33**, 355-362.

Hartigan, J. A. (1975). *Clustering Algorithms,* New York: Wiley.

Mateo-Sanz, J. M. (1998). *Contributions to Statistical Disclosure Control*, Faculty of Mathematics, Universitat de Barcelona (in Catalan).

Mateo-Sanz, J. M., and Domingo-Ferrer, J. (1998). "A method for data-oriented multivariate microaggregation", in *Proceedings of Statistical Data Protection'98,* Amsterdam: IOS Press (to appear).

Ward, J. H. (1963). "Hierarchical grouping to optimize an objective function", *Journal of the American Statistical Association*, **58**, 236-244.