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. So far, practical microaggregation consisted of taking fixed-size microaggregates (size k). We consider in this paper a new approach to multivariate microaggregation in which the size of aggregates is a variable taking values $\geq k$ depending on data.
Keywords: Statistical disclosure control; Microaggregation; Hierarchical clustering; Microdata protection.

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 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 k-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 (this problem is dealt with in Gordon and Henderson, 1977, for instance). In the k-partition problem, groups cannot have a size smaller than k. 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

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

Defays and Nanopoulos (1993) have proposed a mathematical algorithm to find an optimal solution of 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.

Practical heuristic microaggregation methods are proposed in the same paper (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, then 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. 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, then 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.

The rest of this paper is organized as follows. In section 2, a new heuristic method for multivariate microaggregation which is based on hierarchical clustering is presented. A performance comparison with known practical microaggregation algorithms is given in section 3. Finally, section 4 contains some conclusions.

2 Multivariate Ward microaggregation

In Domingo and Mateo (1997) 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 to find a grouping that minimizes SSE.

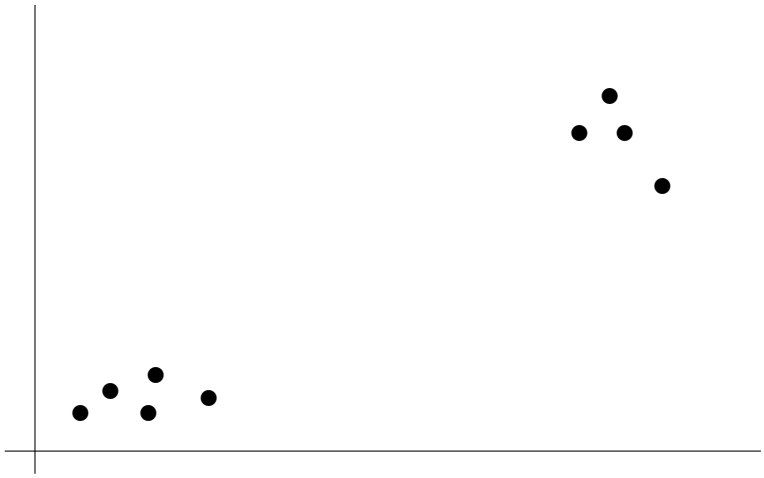

Fig. 1. Variable-sized groups versus fixed-sized groups

In Domingo and Mateo (1997) two alternative heuristic approaches to variable-size univariate microaggregation are presented:

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

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. Unfortunately, such a genetic approach is not so easy to adapt to the multivariate case: 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.

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 into a Modified Ward's Algorithm (MWA) to be useful for microaggregation. 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, MWA can be naturally turned into a truly multivariate microaggregation method.

2.1 Previous work: univariate Modified Ward's Algorithm (MWA)

MWA is a microaggregation method for quantitative data or for qualitative data where a distance has been defined. In what follows, MWA 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 (1997). 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 (MWA) is as follows:

Algorithm 1 (MWA) *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 (1997); 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 MWA, but a good behaviour is expected given that MWA is built on top of Ward's method. See section 3 for computational results.

2.2 Multivariate Modified Ward's Algorithm

Both MWA and genetic algorithms can be used for multivariate microaggregation if they are combined with single-axis or individual sorting of multidimensional data (see section 1). 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. The strong point of MWA is that it can be easily adapted into a multivariate Modified Ward's 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 MWA, 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.

Rather than using single-axis or individual sorting to perform microaggregation, such procedures can be used as sorting criteria to determine which are the first and last k vectors in step 1 of algorithm 1. The multivariate data vectors can be ranked according to their first principal component, their sum of z-scores or a particular variable.

An additional sorting criterion which is specific for MWA 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 3.2).

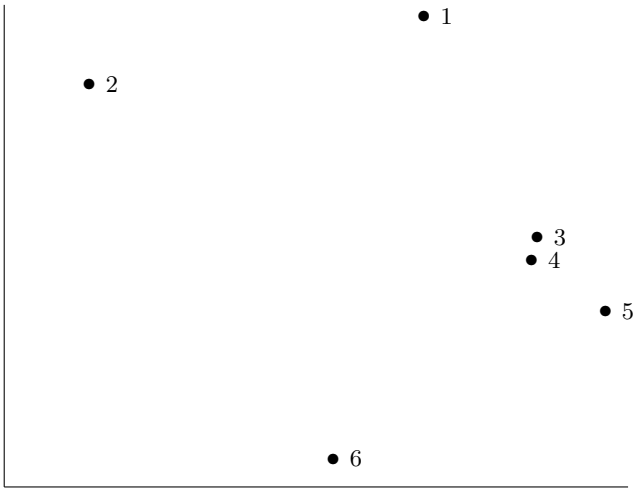

Fig. 2. Grouping with the maximum-distance criterion

2.3 Complexity of the method

The storage complexity $S(MWA)$ of MWA is mainly related to the fact that Ward's clustering method requires a distance matrix containing the distance between each pair of data. Such a distance matrix is symmetrical and has zeroes on its diagonal, so the storage actually needed for a data set of size n is a quadratic function of n:

$$S(MWA) = (n-1)n/2 \tag{2}$$

The strong point of MWA that its information loss is usually smaller than the information loss of other heuristics (see section 3). Unfortunately, the storage requirements and the computing time tend to grow quadratically with the size n of the data set. A way to soften this drawback is to partition the initial data set into several data subsets whose size is more manageable using MWA. For instance, the initial data set could be partitioned into subsets of about 1000 data elements.

3 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.

3.1 Computing time

Several versions of multivariate MWA have been tested; versions differ from each other in the way the k "first" and "last" elements are chosen at step 1 of algorithm 1. Using a Sun Ultra2 Sparc 2160 computer at 166MHz, all versions need about 70 seconds to microaggregate the whole data set of 834 data vectors. If fixed-size microaggregation is run, the computing time is negligible. Anyway, it can be seen that computing time is not an issue for either method; microaggregation is usually performed off-line and even a few hundred seconds would be an acceptable time. The really interesting comparison is in terms of information loss and data quality.

3.2 Information loss and data quality

To compare the information loss caused by multivariate fixed-size microaggregation and MWA, we have considered the loss L (see expression (1)). It must 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 multivariate fixed-size microaggregation (denoted by FS) and for multivariate MWA. For FS, several sorting criteria have been considered: first principal component (FPC), sum of z-scores (SZ), a particular variable (PV) and individual sorting (I). For step 1 of MWA the following sorting criteria have been considered: FPC, SZ, PV, I and maximum-distance (MD). In the tables mentioned in this subsection, the sorting criterion appears as a subscript of the microaggregation method.

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

A range of losses is given for the PV and MD sorting criteria. With those criteria, the information loss depends on the particular variable chosen for sorting and the ordering being ascending or descending. Thus the lower limit of the range corresponds to the best combination (leading to a minimal loss) and the

Method	$100L$ $(k=3)$	$100L$ $(k=4)$	$100L$ $(k=5)$
FS_{FPC}	23.87	30.62 or 25.99	33.29 or 30.74
FS_{SZ}	28.92	32.15 or 32.08	35.20 or 32.56
FS_{PV}	from 30.11 to 48.48	from 34.14 to 56.99	from 37.59 to 60.82
FS_{I}	2.24	5.04	8.54
MWA_{FPC}	16.21	21.58 or 20.94	23.91 or 23.86
MWA_{SZ}	19.16	23.56 or 24.31	26.65 or 27.60
MWA_{PV}	from 16.23 to 21.61	from 20.45 to 29.35	from 22.38 to 31.78
MWA_{MD}	from 16.01 to 16.75	from 21.13 to 21.24	from 21.83 to 22.77
MWA_{I}	2.37	3.28	4.46

Table 1. Percentage information loss for FS and MWA

upper limit to the worst combination. It can be seen that the range for MWA_{PV} is narrower than for FS_{PV}; this is due to the fact that any sorting criterion has a smaller influence on MWA (where it is only used at step 1) than on FS. In this sense, the MD sorting criterion can be termed *robust*, because there is little difference in the information loss between the best and worst combinations variable-ascending/descending.

Table 2 compares the standard deviations of each variable under each method. Original data have been standardized, so 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 standard deviations are to 1, the better is a method.

Var.	Orig.	FS_{FPC}	FS_{SZ}	FS_{PV}	FS_{I}	MWA_{FPC}	MWA_{SZ}	MWA_{PV}	MWA_{DM}	MWA_{I}
V1	1.00	.87	.82	.84	.96	.91	.88	.91	.91	.96
V2	1.00	.94	.89	.88	1.00	.93	.92	.92	.93	1.00
V3	1.00	.82	.81	.79	1.00	.95	.90	.96	.94	1.00
V4	1.00	.95	.95	.99	.99	.94	.93	.94	.94	.99
V5	1.00	.78	.80	.78	.99	.91	.90	.90	.90	.99
V6	1.00	.93	.86	.82	1.00	.92	.92	.92	.93	1.00
V7	1.00	.88	.82	.86	.99	.94	.90	.94	.95	.99
V8	1.00	.67	.49	.61	.99	.81	.78	.81	.82	.99
V9	1.00	.84	.84	.79	1.00	.92	.91	.92	.94	1.00
V10	1.00	.92	.88	.89	.99	.91	.90	.91	.93	.99
V11	1.00	.94	.93	.88	.99	.95	.94	.95	.93	.99
V12	1.00	.89	.91	.85	.98	.91	.90	.91	.89	.98
V13	1.00	.87	.88	.83	.97	.90	.89	.90	.89	.97

Table 2. Standard deviations of original vs. microaggregated variables

For each method, table 3 gives the weight of each variable on the first principal component. The figures enclosed between parentheses are the percentage

change in the weight. The smaller this percentage change, the better the method for that variable. The last row of the table gives the percentage of variability of the whole data set explained by the first principal component under each method. The more similar the percentage explained to 63.4 the better is a method.

Var.	Orig.	FS_{FPC}	FS_{SZ}	FS_{PV}	FS_I	MWA_{FPC}	MWA_{SZ}	MWA_{PV}	MWA_{DM}	MWA_I
V1	.81	.92 (12.9)	.92 (13.5)	.94 (15.8)	.82 (1.0)	.87 (7.2)	.9 (10.4)	.87 (7.5)	.87 (7.4)	.82 (1)
V2	.85	.91 (7.5)	.96 (13.2)	.94 (10.6)	.86 (1.2)	.91 (7.5)	.92 (8.6)	.91 (7.4)	.9 (6.2)	.87 (2.2)
V3	.61	.74 (21.8)	.84 (38.2)	.73 (19.8)	.61 (0.1)	.63 (2.9)	.67 (10.1)	.63 (2.5)	.63 (2.3)	.61 (0.1)
V4	.94	.98 (4.8)	.98 (4.3)	.97 (4.0)	.93 (-0.3)	.97 (3.3)	.97 (3.5)	.97 (3.4)	.97 (3.5)	.93 (-0.2)
V5	.64	.82 (27.3)	.81 (25.9)	.82 (27.6)	.66 (1.9)	.68 (6.1)	.69 (7.3)	.69 (6.6)	.69 (6.7)	.65 (0.4)
V6	.82	.89 (8.2)	.94 (15.3)	.92 (11.9)	.84 (2.0)	.89 (8.2)	.89 (8.9)	.89 (8.3)	.88 (6.9)	.83 (1.6)
V7	.82	.92 (12.7)	.93 (13.6)	.93 (13.9)	.81 (-1)	.87 (6.0)	.91 (10.6)	.87 (5.9)	.87 (6.1)	.81 (-0.8)
V8	-.50	-.8 (60.2)	-.8 (60.9)	-.72 (44.5)	-.54 (8.5)	-.66 (33.1)	-.68 (37.1)	-.66 (32.4)	-.67 (34.5)	-.53 (6.4)
V9	.74	.89 (20.4)	.91 (23.6)	.89 (20.9)	.76 (2.7)	.81 (9.2)	.81 (9.2)	.81 (9.1)	.78 (5.9)	.77 (3.6)
V10	.87	.95 (9.9)	.97 (11.5)	.94 (8.4)	.89 (2.1)	.94 (8.7)	.95 (8.9)	.95 (9.4)	.93 (6.8)	.89 (2.1)
V11	.92	.97 (5.3)	.96 (4.5)	.97 (5.3)	.91 (-1)	.95 (3)	.95 (3.5)	.94 (2.9)	.95 (3.5)	.92 (-0.1)
V12	.87	.95 (9.2)	.95 (9.1)	.96 (10.1)	.86 (-1.7)	.92 (5.1)	.93 (6.6)	.92 (5.3)	.92 (6)	.85 (-2.5)
V13	.85	.95 (11.8)	.94 (10.8)	.95 (12.6)	.84 (-0.5)	.89 (5.7)	.9 (6.2)	.89 (5.6)	.9 (6)	.84 (-0.3)
%	63.4	81.2	84.3	81.4	64.3	72.6	74.7	72.6	72.1	64.3

Table 3. Influence of variables on the first principal component

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 and the standard deviation 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.

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

- With single-axis sorting (FPC, SZ, PV), MWA improves significantly on FS.
- With MD sorting, MWA behaves similarly as with single-axis sorting and is thus better than FS with single-axis sorting.
- With individual sorting, FS and MWA behave similarly and significantly better than with single-axis sorting. However, as pointed out in section 1, individual sorting yields higher disclosure risk.

4 Conclusion and future research

Optimal multivariate microaggregation is a very difficult and time-consuming problem. Conventional practical methods reduce to the one-dimensional case

Method	Average	Std. dev.
FS_{FPC}	0.2037	0.0884
FS_{SZ}	0.2382	0.1031
FS_{PV}	0.2033	0.0827
FS_{I}	0.0251	0.0210
MWA_{FPC}	0.1020	0.0582
MWA_{SZ}	0.1269	0.0584
MWA_{PV}	0.1028	0.0583
MWA_{MD}	0.0969	0.0548
MWA_{I}	0.0241	0.0205

Table 4. Impact of microaggregation methods on correlations

and perform partitioning without looking for natural aggregates. A multivariate microaggregation method (multivariate MWA) has been presented which is data-oriented in that it tries to preserve natural data aggregates. Multivariate MWA is based on Ward's method and in a reasonable computing time it offers a smaller information loss and a better data quality than fixed-size microaggregation.

An interesting line of future research will be to do a performance comparison between the methods presented here and those presented in a forthcoming paper by Nanopoulos, Defays and Kokolakis. The methods in that paper adapt to microaggregation Hanani's algorithm for approximating the optimal solution to the multivariate k-partition problem; such methods share with multivariate MWA the feature of avoiding one-dimensional reduction.

Acknowledgments

We thank the Registre Mercantil de Tarragona for providing the data set used to carry out the tests described in section 3.

References

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

DEFAYS, D., 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., (1997), "Practical Data-Oriented Microaggregation for Statistical Disclosure Control", *Journal of Classification,* (submitted).

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.
NANOPOULOS, P., DEFAYS, D., and KOKOLAKIS, G., "Clustering analysis under a cardinality constraint", *Journal of Classification* (to appear).
WARD, J. H., (1963), "Hierarchical grouping to optimize an objective function", *Journal of the American Statistical Association*, 58, 236-244.