The VLDB Journal (2006) 15:355–369
DOI 10.1007/s00778-006-0007-0

SPECIAL ISSUE PAPER

Efficient multivariate data-oriented microaggregation

Josep Domingo-Ferrer · Antoni Martínez-Ballesté · Josep Maria Mateo-Sanz · Francesc Sebé

Received: 30 September 2005 / Accepted: 25 May 2006 / Published online: 29 August 2006

Abstract Microaggregation is a family of methods for statistical disclosure control (SDC) of microdata (records on individuals and/or companies), that is, for masking microdata so that they can be released while preserving the privacy of the underlying individuals. The principle of microaggregation is to aggregate original database records into small groups prior to publication. Each group should contain at least k records to prevent disclosure of individual information, where k is a constant value preset by the data protector. Recently, microaggregation has been shown to be useful to achieve k-anonymity, in addition to it being a good masking method. Optimal microaggregation (with minimum within-groups variability loss) can be computed in polynomial time for univariate data. Unfortunately, for multivariate data it is an NP-hard problem. Several heuristic approaches to microaggregation have been proposed in the literature. Heuristics yielding groups with fixed size k tends to be more efficient, whereas data-oriented heuristics yielding variable group size tends to result in lower information loss. This paper presents new data-oriented heuristics which improve on the trade-off between computational complexity and information loss and are thus usable for large datasets.

Keywords Statistical databases · Privacy · Anonymity · Statistical disclosure control · Microaggregation · Microdata protection

J. Domingo-Ferrer (✉) · A. Martínez-Ballesté · F. Sebé
Department of Computer Engineering & Maths,
Rovira i Virgili University of Tarragona,
Av. Països Catalans 26, Tarragona, Catalonia
e-mail: josep.domingo@urv.cat

A. Martínez-Ballesté
e-mail: antoni.martinez@urv.cat

F. Sebé
e-mail: francesc.sebe@urv.cat

J. M. Mateo-Sanz
Statistics Group, Rovira i Virgili University of Tarragona,
Av. Països Catalans 26, Tarragona, Catalonia
e-mail: josepmaria.mateo@urv.cat

1 Introduction

Statistical disclosure control (SDC) in statistical and administrative databases attempts to balance the societal or corporate right to know with the individual right to privacy. Traditionally, SDC methods have been used to protect *respondent privacy* when releasing data collected by official statistics from individuals or companies: most statistical laws contain confidentiality commitments toward citizens, in order to increase their response rates and the accuracy in their responses. More recently, SDC has found new applications in e-commerce and e-health [2]: indeed, massive automated data collection raises privacy concerns and any transfer of such data to third parties should be SDC-protected.

Statistical disclosure control can be applied to information in several formats: tables, responses to dynamic database queries and microdata (individual respondent records). See Doyle et al. [14], Willenborg and DeWaal [41] for a survey of SDC techniques. The protection provided by SDC normally results in some degree of data modification or *masking*. Modification can be perturbative (perturbing the data to some extent) or non-perturbative (e.g., sampling or partially suppressing the original data rather than perturbing them). *The challenge in SDC is to tune modification so that both privacy*

and information loss are acceptable: both the risk of disclosing private confidential information and the loss of data utility should be kept below reasonable thresholds preset by the data protector.

1.1 Contribution and plan of this paper

This paper is about microaggregation, a class of perturbative SDC methods for microdata. Given an original set of microdata whose respondents (i.e., contributors) must have their privacy preserved, microaggregation yields a protected data set consisting of aggregate information (e.g., mean values) computed on small groups of records in the original dataset. Since this protected dataset contains only aggregate data, its release is less likely to violate respondent privacy. For the released dataset to stay analytically useful, the information loss caused by microaggregation must be minimized: a way to approach this minimization is for records within each group to be as homogeneous as possible. Multivariate microaggregation (for several attributes) with maximum within-groups record homogeneity is NP-hard, so heuristics are normally used.

There is a dichotomy between *fixed-size heuristics* yielding groups with a fixed number of records and *data-oriented heuristics* yielding groups whose size varies depending on the distribution of the original records. Even if the latter heuristics can in principle achieve lower information loss than fixed-size microaggregation (see discussion in Sect. 2 below), they are often dismissed for large datasets due to complexity reasons. For example, the μ-Argus SDC package [21] only features fixed-size microaggregation. Our contribution in this paper is an approach to turn some fixed-size heuristics for multivariate microaggregation of *numerical* data into data-oriented heuristics with little additional computation. The resulting new heuristics improve the trade-off between information loss and computational complexity: the former is reduced without significantly increasing the latter.

Section 2 is a brief survey of microaggregation and previous work on it. Section 3 points out the privacy benefits of microaggregation. The new data-oriented heuristics are described in Sect. 4. Section 5 is an empirical comparison with alternative fixed-size and data-oriented heuristics in the literature. Section 6 is a conclusion.

2 Basics of microaggregation

Microaggregation is a family of perturbative SDC methods originally designed for continuous numerical data [8–10] and recently extended for categorical data [13,36]. Whatever the data type, microaggregation can be operationally defined in terms of two steps:

Partition The set of original records is partitioned into several groups in such a way that records in the same group are *similar* to each other and so that the number of records in each group is at least k. A partition meeting this requirement on minimal group size is called a k-partition.

Aggregation An aggregation operator (for example, the mean for numerical data) is used to compute a centroid for each group. Then, each record in a group is replaced by the group centroid.

In Domingo-Ferrer and a Mateo-Sanz [10], 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.

To be more specific, consider a microdata set V with p continuous numerical attributes and n records (i.e., the result of observing p attributes on n individuals). With these records, groups are formed with n_i records in the i-th group ($n_i \geq k$ and $n = \sum_{i=1}^{g} n_i$, where g is the number of resulting groups). Denote by $\mathbf{x}_{ij}$ the j-th record in the i-th group and by $\bar{\mathbf{x}}_i$ the mean record (centroid) over the i-th group.

The sum of squares criterion is common to measure homogeneity in clustering [15–17,22,25,40]. In terms of sums of squares, maximizing within-groups homogeneity is equivalent to finding a k-partition minimizing the within-groups sum of squares SSE defined as

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

where g is the number of groups in a k-partition (g is not fixed and depends on the particular k-partition). It is shown in Domingo-Ferrer and Mateo-Sanz [10] that the sizes of groups in the optimal k-partition lie between k and $2k - 1$.

Microaggregating categorical data requires some adaptations described in Domingo-Ferrer and Torra [13]. Basically, the distance used for partitioning and the operation used for aggregation must be suitable for categorical data, which precludes the Euclidean distance and the arithmetic mean; thus, SSE minimization as defined above is no longer valid as an optimality criterion for categorical data. This paper is devoted to numerical data, the traditional application field of microaggregation, so we can think of distances being Euclidean,

centroids being mean records and optimality being *SSE* minimization.

In Oganian and Domingo-Ferrer [29], it was shown that, for multivariate records, optimal microaggregation is an NP-hard problem. For univariate data, a polynomial-time optimal algorithm is given in Hansen and Mukherjee [18]. This algorithm has complexity $O(k^2n)$ and solves optimal univariate microaggregation as a shortest-path problem on a graph. Unfortunately, realistic datasets are multivariate, so in practice microaggregation is multivariate and heuristic.

As mentioned in the introduction, there exist two main types of heuristics:

- *Fixed-size microaggregation* These heuristics yield k-partitions where all groups have size k, except perhaps one group which has size between k and $2k-1$. Examples of fixed-size microaggregation heuristics can be found in [8,10,20].
- *Data-oriented microaggregation* These heuristics yield k-partitions where all groups have sizes varying between k and $2k-1$. Examples of such heuristics can be found in [10,23,26,34]. Note that adapting standard clustering techniques (e.g., k-means, [19]) for microaggregation is not trivial: the challenge is how to enforce cardinality constraints on groups without substantially increasing *SSE*.

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. The idea behind variable group size is to avoid unnatural k-partitions [10]. For instance, for the bivariate dataset in Fig. 1, microaggregation using a fixed size of $k = 3$ records is rather unnatural. The optimal 3-partition for this dataset obviously consists of a group with four records and another group with five records.

Fig. 1 Variable-versus fixed-sized groups

3 Privacy benefits of microaggregation

The attributes in an original unprotected microdata set V can be classified in four categories which are not necessarily disjoint:

- *Identifiers* These are attributes that *unambiguously* identify the respondent. Examples are passport number, social security number, full name, etc. Since our objective is to prevent confidential information from being linked to specific respondents, we will assume in what follows that, in a pre-processing step, identifiers in V have been removed/encrypted.
- *Key attributes* Key attributes (called quasi-identifiers in Dalenius [6] and Samarati [32]) are a set of attributes that, in combination, can be linked with external information to re-identify (some of) the respondents to whom (some of) the records in V refer. Examples of key attributes are age, gender, job, zipcode, etc. Unlike identifiers, key attributes cannot be removed from V. The reason is that any attribute in V is potentially a key attribute, depending on the external data sources available to a snooper. Thus one would need to remove all attributes (!) to make sure that the dataset no longer contains key attributes.
- *Confidential outcome attributes* These are attributes which contain sensitive information on the respondent. Examples are salary, religion, political affiliation, health condition, etc.
- *Non-confidential outcome attributes* Those are attributes which contain non-sensitive information on the respondent. Note that attributes of this kind cannot be neglected when protecting a dataset, because they can also be key attributes. For instance, job and town of residence may reasonably be considered non-confidential outcome attributes, but their combination can be a quasi-identifier because everyone knows who is the doctor in a small village.

The usual practice in SDC is for the data protector to apply microaggregation to a restricted set of attributes rather than to entire records in a dataset. From a mathematical point of view, one can say that what is microaggregated are the projections of records on this restricted set of attributes, which may include key attributes and confidential attributes. Further, for a given set of attributes, microaggregation can be carried out jointly for all attributes in the set, independently for each

attribute – a variant known as individual ranking – or jointly within disjoint subsets of attributes [12].

When microaggregation is applied to confidential outcome attributes, respondent re-identification by a snooper using unaltered key attributes may still be possible. The goal in this case is to prevent *attribute disclosure* rather than re-identification: the snooper only obtains perturbed values for the confidential attributes of the re-identified respondents. For confidential outcome attributes masked by adding noise with a known distribution, an information-theoretic measure is proposed in Agrawal and Aggarwal [1] to quantify attribute disclosure. Unfortunately, adapting this measure for microaggregated data is not straightforward; an alternative is for the data protector to quantify attribute disclosure empirically by using the interval disclosure measure proposed in Domingo-Ferrer and Torra [12]: for each attribute, the data protector computes the proportion of records for which the original attribute value falls within an interval centered on the corresponding microaggregated attribute value (the width of the interval is a pre-specified multiple of the standard deviation of the original attribute). The higher the proportion, the higher is the risk of attribute disclosure.

If the key attributes can be precisely ascertained by the data protector (because he knows what attributes in V are also available in external identified data sources), then a possibility is to apply microaggregation to the set of key attributes. The purpose here is to oppose *re-identification*. An important special case is when all key attributes are jointly microaggregated. As pointed out in Domingo-Ferrer and Torra [13], this special case yields a k-anonymous microaggregated dataset V', where k-anonymity is the following property defined in [32,33,35].

Definition 1 (k-anonymity) *A dataset is said to satisfy k-anonymity for $k > 1$ if, for each combination of values of key attributes, at least k records exist in the dataset sharing that combination.*

If a dataset V' satisfies k-anonymity, a snooper attempting re-identification with an identified external source S can only hope to map an identified record in S to a group of k records in V' (each record in the group is seen as an equally likely match by the snooper). Therefore, if, for a given k, k-anonymity is assumed to be enough protection, one can concentrate on minimizing information loss with the only constraint that k-anonymity should be satisfied.

There are some situations in which k-anonymity or microaggregation of key attributes are insufficient and must be supplemented by perturbation of confidential outcome attributes:

- If some key attributes available to snoopers are known to be strongly correlated to a confidential outcome attribute, the latter should also be microaggregated or perturbed in some way. If only k-anonymity was used, a snooper could use the known correlations and the unaltered values of the confidential attribute in the k-anonymous dataset to re-identify the correct record within a group of k with probability greater than $1/k$; this kind of attack is frustrated if the correlated confidential attribute is perturbed.
- If a confidential outcome attribute takes a constant value within a group of k records sharing key attribute values, a snooper knows the value of that attribute for any of the k respondents in the group. For numerical confidential attributes (the ones considered in this paper) this situation is much less likely than for categorical confidential attributes. Anyway, there are at least two possible fixes : (1) to use an SDC method based on noise addition [3] to perturb the values of the confidential attribute; (2) to recompute microaggregation using a different k or a different heuristic, in order to get different groups.

If the precise set of key attributes cannot be anticipated by the data protector or if jointly microaggregating all key attributes results in unacceptable information loss, microaggregation can still be used in ways not leading to k-anonymity (e.g., on confidential outcome attributes or independently on disjoint groups of attributes, regardless of whether these are key or confidential attributes). In such a general application of microaggregation, the degree of anonymity offered by a particular microaggregated dataset is usually assessed using record linkage experiments (see Torra and Domingo-Ferrer [37] for a survey on record linkage algorithms). In those experiments, a variety of possible disclosure scenarios are considered, most of them different from the disclosure scenario used to microaggregate the data. For each disclosure scenario, one attempts re-identification by linking the exact values of the key attributes in the scenario with the microaggregated values for those key attributes. The higher the proportion of correctly re-identified records, the lower is the anonymity provided. Empirical record linkage studies comparing microaggregation with other SDC methods can be found in Domingo-Ferrer and Torra [12], Lenz and Vorgrimler [24]; they conclude that some variants of microaggregation – splitting key attributes in several groups and jointly microaggregating attributes within a group independently from other groups – rank among the best SDC methods in terms of the trade-off between anonymity and information loss.

The relevance of microaggregation for privacy in large statistical databases is endorsed by its use by statistical agencies world-wide. Microaggregation was proposed at Eurostat [9] in the early nineties, and has since then been used by the Italian agency [30], the German federal agency [24,31] and several other national agencies [38,39].

Example 1 This example illustrates the use of microaggregation for SDC and, more specifically, for k-anonymity. We show in Table 1 a dataset giving, for 11 small or medium enterprises (SMEs) in a certain town, the company name, the surface in square meters of the company's premises, its number of employees, its turnover and its net profit. Clearly, the company name is an identifier. We will consider that turnover and net profit are confidential outcome attributes. A first SDC measure is to suppress the identifier "Company name" when releasing the dataset for public use. However, note that the surface of the company's premises and its number of employees can be used by a snooper as key attributes. Indeed, it is easy for anybody to gauge to a sufficient accuracy the surface and number of employees of a target SME. Therefore, *if the only privacy measure taken when releasing the dataset in Table 1 is to suppress the company name*, a snooper knowing that company K&K Sarl has about a dozen employees crammed in a small flat of about 50 m^2 will still be able to use the released data to link company K&K Sarl with turnover 645,223 Euros and net profit 333,010 Euros.

Table 2 is a 3-anonymous version of the dataset in Table 1. The identifier "company name" was suppressed and optimal bivariate microaggregation with $k=3$ was used on the key attributes "Surface" and "No. employees" (in general, if there are p key attributes, multivariate microaggregation with dimension p should be used to mask all of them). Both attributes were standardized to have mean 0 and variance 1 before microaggregation, in order to give them equal weight, regardless of their scale. Due to the small size of the dataset, it was feasible to compute optimal microaggregation by exhaustive search. The information or variability loss incurred for those two attributes in standardized form was $SSE_{opt} = 7.484$. Dividing by the total sum of squares $SST = 22$ – sum of squared Euclidean distances from all 11 pairs of standardized (surface, number of employees) to their average – yielded a variability loss measure $SSE_{opt}/SST = 0.34$ bounded between 0 and 1.

It can be seen that the 11 records were microaggregated into three groups: one group with the 1st, 2nd, 3rd and 10th records (companies with large surface and many employees), a second group with the 4th, 5th and 9th records (companies with large surface and few employees) and a third group with the 6th, 7th, 8th and 11th records (companies with a small surface). Upon seeing Table 2, a snooper knowing that company K&K Sarl crams a dozen employees in a small flat hesitates between the four records in the third group. Therefore, since turnover and net profit are different for all records in the third group, the snooper cannot be sure about their values for K&K Sarl. □

Table 1 Example: SME dataset. "Company name" is an identifier to be suppressed before publishing the dataset

Company name	Surface (m^2)	No. employees	Turnover (Euros)	Net profit (Euros)
A&A Ltd	790	55	3,212,334	313,250
B&B SpA	710	44	2,283,340	299,876
C&C Inc	730	32	1,989,233	200,213
D&D BV	810	17	984,983	143,211
E&E SL	950	3	194,232	51,233
F&F GmbH	510	25	119,332	20,333
G&G AG	400	45	3,012,444	501,233
H&H SA	330	50	4,233,312	777,882
I&I LLC	510	5	159,999	60,388
J&J Co	760	52	5,333,442	1,001,233
K&K Sarl	50	12	645,223	333,010

Table 2 Example: 3-anonymous version of the SME dataset after optimal microaggregation of key attributes

Surface (m^2)	No. employees	Turnover (Euros)	Net profit (Euros)
747.5	46	3,212,334	313,250
747.5	46	2,283,340	299,876
747.5	46	1,989,233	200,213
756.67	8	984,983	143,211
756.67	8	194,232	51,233
322.5	33	119,332	20,333
322.5	33	3,012,444	501,233
322.5	33	4,233,312	777,882
756.67	8	159,999	60,388
747.5	46	5,333,442	1,001,233
322.5	33	645,223	333,010

4 The new heuristics

A multivariate dataset consisting of n records and p numerical attributes can be represented as n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^p$. The new data-oriented heuristics proposed in this paper operate in three steps:

1. Find a path T traversing all n points of the dataset. Let the π_T be the permutation of $\{1, \ldots, n\}$ expressing the order in which the points are traversed by T.

2. Use the ordering π_T to feed the p-variate records to a multivariate version of the Hansen–Mukherjee algorithm [18], called MHM algorithm. Formally, pass the ordered tuple $(\mathbf{x}_{\pi_T(1)}, \ldots, \mathbf{x}_{\pi_T(n)})$ to MHM, which outputs a data-oriented k-partition for that ordered dataset.
3. Microaggregate $(\mathbf{x}_{\pi_T(1)}, \ldots, \mathbf{x}_{\pi_T(n)})$ using the k-partition output by MHM.

We will assume in what follows that $n \gg p$, i.e., the number of records is much larger than the number of attributes. Strictly speaking, algorithm complexities computed in the rest of this paper depend on p. However, since the dependence on p is always linear, p will be omitted for simplicity and complexity will be expressed only as a function of n and k.

The MHM algorithm is described below and yields a data-oriented k-partition of a p-variate ordered dataset. In general, the resulting partition is not optimal in $\mathbb{R}^p$, but it certainly is the best k-partition compatible with the prescribed order of points, where compatibility is defined next.

Definition 2 (k-partition compatible with an order) *Given an ordered set $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of p-variate points, a k-partition is said to be compatible with the order of the points if, for any group G in the k-partition and any three points $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$, such that $i \leq j \leq k$ and $\mathbf{x}_i, \mathbf{x}_k \in G$, it holds that $\mathbf{x}_j \in G$*

Therefore, the choice of the path is critical to obtain a good k-partition, i.e., with low within-groups sum of squares. We first describe the MHM algorithm and then several ways of constructing a path traversing all points (records) in a multivariate dataset.

4.1 The MHM algorithm

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be an ordered dataset with n records where each record $\mathbf{x}_i$ contains the values of p attributes. Let k be an integer group size such that $1 \leq k < n$. Then, a graph $\mathbf{G}_{n,k}$ is constructed as follows:

1. For each value $\mathbf{x}_i$ in $\mathbf{X}$, create a node with label i. Create also an additional node with label 0.
2. For each pair of graph nodes (i, j) such that $i + k \leq j < i + 2k$, create a directed arc (i, j) from node i to node j.
3. Map each arc (i, j) to the group of values $C_{(i,j)} = \{\mathbf{x}_h : i < h \leq j\}$. Let the length $L_{(i,j)}$ of the arc be the within group sum of squares for $C_{(i,j)}$, that is,

$$L_{(i,j)} = \sum_{h=i+1}^{j} (\mathbf{x}_h - \bar{\mathbf{x}}_{(i,j)})'(\mathbf{x}_h - \bar{\mathbf{x}}_{(i,j)})$$

where $\bar{\mathbf{x}}_{(i,j)}$ is a p-dimensional record computed as the centroid (average) of records in $C_{(i,j)}$.

Like in the univariate Hansen–Mukherjee algorithm, the k-partition output by MHM is computed as the one whose groups correspond to the arcs in the shortest path between nodes 0 and n. The complexity of MHM is the same as the complexity of the univariate Hansen–Mukherjee algorithm, that is $O(k^2 n)$.

Lemma 1 *For a fixed path traversing a dataset of multivariate points, MHM yields the best k-partition (with lowest SSE) compatible with the ordering of points induced by the path.*

Proof The lemma follows from the optimality of the univariate Hansen–Mukherjee algorithm proven in [18]. □

4.2 Path construction

We next describe several ways of contructing a path traversing a multivariate dataset, whose records are assumed to contain the values of p numerical attributes and are represented as points in $\mathbb{R}^p$. All path constructions are based on connecting nearby points, which is in line with the purpose of microaggregation: to form groups such that the distances between points in a group are short. Except the first construction (NPN), the rest of constructions are essentially existing fixed-size microaggregation heuristics (maximum distance, MDAV, CBFS) used only for the purpose of ordering multivariate points.

4.2.1 Nearest point next (NPN)

We first proposed this construction in Mateo-Sanz and Domingo-Ferrer [27]. We give here a slightly faster variant based on the dataset centroid. The p-dimensional Euclidean distance is used. The path is constructed as follows:

1. Compute the centroid (average record) $\bar{\mathbf{x}}$ of all points in the dataset.
2. Compute the most distant point $\mathbf{r}$ from $\bar{\mathbf{x}}$ and take $\mathbf{r}$ as the first point in the path.
3. The second point is the closest one to the first (among the remaining points), the third point is the closest one to the second (among the remaining points

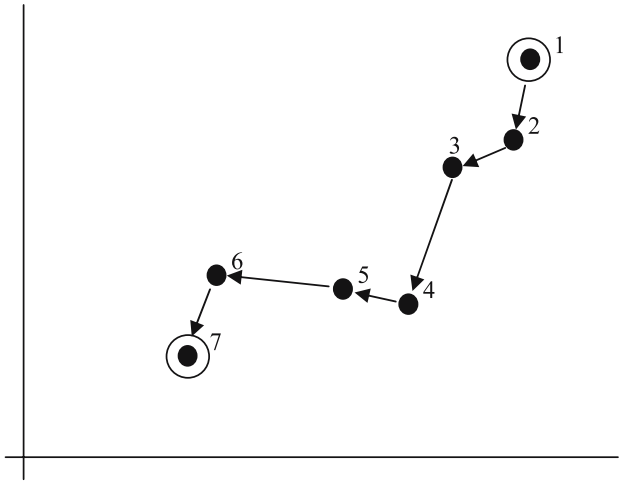

Fig. 2 Path construction on bivariate microdata using NPN

different from the first and the second) and so on until all n points have been added to the path.

Figure 2 illustrates the NPN construction on a dataset with seven points. Point no. 1 is taken as first; then nodes are added to the path in the order indicated by their label.

Lemma 2 *The complexity of the NPN construction is* $O(\frac{n^2}{2})$. *Specifically, the construction requires* $\frac{n(n+1)}{2} - 1$ *distance computations.*

Proof The complexity of the NPN construction can be measured as the number of required distance computations. For a dataset with n points, we must:

- Compute the dataset centroid $\bar{\mathbf{x}}$.
- After computing the dataset centroid $\bar{\mathbf{x}}$, find the most distant point $\mathbf{r}$ from the centroid, which requires computing n distances to choose the maximum one.
- Construct the path starting at $\mathbf{r}$. To determine the second point in the path, $n-1$ distance computations are needed (from $\mathbf{r}$ to the $n-1$ remaining points). To determine the third point, $n-2$ distance computations are required (from the second point to the $n-2$ remaining points), and so on, up to the $n-1$-th point, whose determination requires two distance computations (from the $n-2$-th point to the remaining two points). After that, the n-th point is the remaining one.

Thus, the total number of distance computations is

$$n + \sum_{i=2}^{n-1}(n-i+1) = n + \sum_{i=2}^{n-1} i = \frac{n(n+1)}{2} - 1$$

The centroid computation takes $O(n)$ additions and $O(1)$ divisions, a cost which is negligible in front of the quadratic number of distances to be computed. □

Note 1 Comparisons are not explicitly mentioned in the above complexity assessment for NPN. Distance computation is always done to find a maximum or a minimum distance, so we include the cost of comparisons in the cost of distance computations. The same remark applies to the complexity assessments in the rest of this paper.

The NPN construction is the only path construction among the ones proposed in this paper which orders multivariate points without making any assumption on the minimum group size k to be used later by MHM for microaggregation. This turns out to be a good idea to reduce information loss when data points are skewed or naturally clustered (see results in Sect. 5). For homogeneous datasets (with dense points without any big gaps between them), the constructions in the next sections perform better in terms of information loss.

4.2.2 Maximum distance

We first proposed this construction in Mateo-Sanz and Domingo-Ferrer [26] and later in Domingo-Ferrer and Mateo-Sanz [10] as a standalone multivariate microaggregation method. It was also used in Laszlo, and Mukherjee [23], where the authors called it diameter-based fixed-size microaggregation.

We next adapt the maximum distance algorithm for path construction rather than microaggregation:

1. Compute:
   (a) The two most distant points $\mathbf{r}$ and $\mathbf{s}$ in the dataset, using the Euclidean distance;
   (b) Form a group with $\mathbf{r}$ and the $k-1$ points closest to $\mathbf{r}$; form a group with $\mathbf{s}$ and the $k-1$ points closest to $\mathbf{s}$.
2. If there are at least $2k$ points which do not belong to any of the two groups formed in Step 1, go to Step 1 taking as the new dataset the previous dataset minus the groups formed in the previous instance of Step 1.
3. If there are between k and $2k-1$ points which do not belong to any of the two groups formed in Step 1, form a new group with those points and go to Step 5.
4. If there are less than k points which do not belong to any of the groups formed in Step 1, add them to the closest group formed in Step 1.
5. At this moment, we have a k-partition of the dataset into a number of groups; we construct a group-level path starting at the first group created in the first instance of Step 1; we use the NPN construction

above on the centroids of the groups to add the remaining groups in turn to the group-level path.

6. Finally a point-level path is constructed as follows:
   (a) Points in the first group are sorted so that the point **r** of the group is taken as first and the remaining $k-1$ points are sorted from least to most distant to **r**.
   (b) Points within a group are sorted as follows. The first point is the one closest to the centroid of the preceding group in the group-level path. The rest of points in the group are sorted according to their distance to the first point: the second point is the closest neighbor of the first point, the third point is the second closest neighbor and so on.

Note also that the parameter value k used in maximum distance does not need to be the same value k subsequently used in the MHM algorithm. The reason is that maximum distance forms groups of fixed size k (only used for sorting purposes), whereas MHM yields groups of sizes varying between k and $2k-1$. However, empirical results (Sect. 5) show that using the same value k for both maximum distance and MHM is a wise choice.

Lemma 3 *The complexity of maximum distance is* $O(\frac{n^3}{12k})$.

Proof Like in NPN, the complexity of maximum distance is measured as the number of required distance computations. In the rest of the proof, we assume that the number n of points in the dataset is a multiple of $2k$; this simplifies the derivation and does not affect the resulting complexity, except perhaps by an added constant (the complexity of Steps 3 and 4). Let us analyze the various computations required:

- *Finding the two most distant points* **r** *and* **s** *(Step 1a)* When all n points are still ungrouped, $n(n-1)/2$ distance computations are needed. After the first iteration, $n-2k$ points stay ungrouped, so $(n-2k)(n-2k-1)/2$ distance computations are needed, and so on. Thus, the total number of distances to be computed is

$$\sum_{i=0}^{(n/2k)-1} \frac{(n-2ik)(n-2ik-1)}{2} = \frac{1}{12k}n^3 + \frac{2k-1}{8k}n^2 + \frac{2k-3}{12}n \qquad (1)$$

- *Forming groups around* **r** *and* **s** *(Step 1b)* When $n-2$ points stay ungrouped, $n-2$ distance computations are needed to form a group around **r** with the $k-1$ closest points to **r**; next, $n-k-1$ distance computations are needed to form a group around **s** (from **s** to all remaining ungrouped points). In the next iteration, there are $n-2k$ ungrouped points among which *new* most distant points **r** and **s** are identified; forming groups around the new **r** and **s** takes $n-2k-2$ and $n-3k-1$ distance computations, respectively. This can be carried on. In the end, the total number of distance computations is

$$\sum_{i=0}^{(n/2k)-1} (n-2ik-2) + \sum_{i=0}^{(n/2k)-2} (n-(2i+1)k-1) = \frac{1}{2k}n^2 + \frac{k-3}{2k}n - k + 1 \qquad (2)$$

- *Group-level path construction* At Step 5 of maximum distance, the NPN construction is used on the n/k centroids of the formed groups. According to Lemma 2, the number of distance computations needed is

$$\frac{(n/k)((n/k)+1)}{2} - 1 = \frac{n^2}{2k^2} + \frac{n}{2k} - 1 \qquad (3)$$

- *Point-level path construction* At Step 6 of maximum distance, a point-level path is constructed. Sorting points in the first group requires $k-1$ distance computations (from **r** to all remaining points in the group). Sorting points in the remaining $(n/k)-1$ groups needs $k+(k-1)$ distance computations (k distances from the centroid of the previous group to each point in the current group, plus $k-1$ distances from the first point in the current group to the remaining points in the current group). Therefore, Step 6 takes the following number of distance computations:

$$k-1+(2k-1)(n/k-1) = 2n - \frac{n}{k} - k \qquad (4)$$

Adding up Expressions (1), (2), (3) and (4) we get that the total complexity is $O(\frac{n^3}{12k})$ □

Note 2 The complexity in Lemma 3 has been computed assuming that the distance matrix is not stored. If the number of records n is large, storing $O(n^2)$ distances may be unfeasible. However, if enough storage is available, a stored distance matrix approach can be used. In this case, finding r and s requires $n(n-1)/2$ distance computations; after this, maximum distance can be completed with $O(\frac{n^3}{12k})$ comparisons (instead of distance computations). Alternatively, the $n(n-1)/2$ distances can be sorted, which takes $O(n^2 \log n)$ operations, and no further comparisons are needed if suitable data structures are used.

4.2.3 Maximum distance to average vector (MDAV)

We first proposed this construction in Hundepool et al. [20] as part of a multivariate microaggregation method implemented in the μ-Argus package for statistical disclosure control. A slight modification of same construction was later described in Laszlo and Mukherjee [23] under the name centroid-based fixed size microaggregation.

We give below a novel adaptation of MDAV for path construction:

1. Compute:
   (a) The centroid (average record) $\bar{\mathbf{x}}$ of all points in the dataset;
   (b) The most distant point **r** from the centroid;
   (c) The most distant point **s** from **r**.
2. Form two groups around **r** and **s**: the first group contains **r** and the $k-1$ points closest to **r**; the other group contains **s** and the $k-1$ points closest to **s**.
3. If there are at least $2k$ points which do not belong to any of the two groups formed in Step 2, go to Step 1 taking as new set of points the previous set of points minus the groups formed in the latest instance of Step 2.
4. If there are between k and $2k-1$ points which do not belong to any of the two groups formed in Step 2, form a new group with those points and go to Step 6.
5. If there are less than k remaining points which do not belong to any of the groups formed in Step 2, add them to the group formed in Step 2 whose centroid is closest to the centroid of the remaining points.
6. At this moment, we have a k-partition of the dataset into a number of groups; we construct a group-level path starting at the first group created in the first instance of Step 2; we use the NPN construction above on the centroids of the groups to add the remaining groups in turn to the group-level path.
7. Finally, a point-level path is constructed as follows:
   (a) Points in the first group are sorted so that the point **r** of the group is taken as first and the remaining $k-1$ points are sorted from least to most distant to **r**.
   (b) Points within a group are sorted so that the first point is the one closest to the centroid of the preceding group in the group-level path, the second point is the closest one to the first point, and so on.

Centroid-based fixed-size microaggregation (CBFS, [23]) is very similar to MDAV. At each iteration, CBFS differs from MDAV in that CBFS does not compute **s** and its group. Thus, while MDAV uses the centroid computed in each iteration to form two groups around **r** and **s**, respectively, CBFS only computes one group per iteration and centroid.

It is easy to show that MDAV and CBFS with $k=1$ are equivalent to NPN: indeed, for $k=1$ MDAV and CBFS reduce to a group-level path formed using NPN where groups consist of a single point and the first point is the one farthest from the dataset centroid. MDAV and CBFS can also be regarded as faster variants of maximum distance. Maximum distance needs to find the two most distant points **r** and **s** several times, an operation which requires $O(n^2)$ distance computations for a set of n points. MDAV and CBFS compute a centroid, then find the most distant point **r** from the centroid and, in the case of MDAV, the most distant point **s** from **r**, which makes $O(n)$ distance computations for a dataset of n points. This gives a reduction in complexity:

Lemma 4 *The complexity of MDAV, respectively CBFS, is* $O(\frac{n^2}{2k})$.

Proof We focus on MDAV, the order of complexity for CBFS being the same. Like for the previous path constructions, the complexity of MDAV is measured as the number of required distance computations. In the rest of the proof, we assume that the number n of points in the dataset is a multiple of $2k$; this simplifies the derivation and does not affect the resulting complexity, except perhaps by an added constant (the complexity of Steps 4 and 5). Let us analyze the various computations required:

- *Computing centroids (Step 1a)* At each iteration, a centroid must be computed. This makes $n/2k$ centroid computations. In the first iteration, the centroid is computed over n points; in the second iteration, it is computed over $n-2k$ points, and so on. The total number of points involved in the computation of the $n/2k$ centroids is $n^2/4k + n/2$; thus $O(\frac{n^2}{4k})$ additions and $O(\frac{n}{2k})$ divisions are performed.
- *Finding* **r** *(Step 1b)* If there are n ungrouped points, n distance computations are needed. If there are $n-2k$ ungrouped points left, $n-2k$ distance computations are needed, and so on. Thus, the total number of distance computations to find **r** in the $n/2k$ iterations is

$$\sum_{i=0}^{(n/2k)-1} (n-2ik) = \frac{n^2}{4k} + \frac{n}{2} \tag{5}$$

- *Finding* **s** *(Step 1c)* In the first iteration, $n-1$ distance computations are needed (from **r** to the remaining points). In the second iteration, $n-2k-1$ distance

computations are needed, and so on. The total number of distance computations to find **s** in the $n/2k$ iterations is

$$\sum_{i=0}^{(n/2k)-1} (n - 2ik - 1) = \frac{n^2}{4k} + \frac{(k-1)n}{2k} \quad (6)$$

The remaining steps are analogous to the final steps of maximum distance (group-level path and point-level path construction). The number of distance computations required by those steps can be found in the proof of Lemma 3. Adding up the complexity of Expressions (5), (6), (3) and (4), we get a total complexity of $O(\frac{n^2}{2k})$. Centroid computation does not affect this overall complexity because it requires $O(\frac{n^2}{4k})$ additions and $O(\frac{n}{2k})$ divisions and those operations can be considered simpler than distance computations. □

4.3 The resulting heuristics and their complexity

Table 3 summarizes the complexities of the four heuristics NPN–MHM, MD–MHM, MDAV–MHM and CBFS–MHM resulting from plugging MHM after NPN, maximum distance, MDAV and CBFS, respectively. Since MHM runs in $O(k^2n)$ time, the complexity of the heuristics is dominated by the path construction.

Note 3 MDAV–MHM and CBFS–MHM are the fastest heuristics among those in Table 3. The order of magnitude of their complexity is the same as the one of the most efficient fixed-size microaggregation heuristics in the literature (MDAV and CBFS). One might argue that quadratic time is still too long for very large datasets. If n is very large, a good strategy is to use blocking attributes to split the n records into several smaller subdatasets which can be microaggregated independently. This is no real shortcoming, because statistical agencies routinely use blocking for treatments other than SDC or microaggregation [31,24]. One or several categorical attributes (like province, activity sector, etc.) are used as blocking attributes, and the typical block sizes are a few thousands of records.

Table 3 Complexities of the combinations of the four path constructions with MHM

Heuristic	Complexity
CBF–MHM	$O(\frac{n^2}{2})$
MD–MHM	$O(\frac{n^3}{12k})$
MDAV–MHM	$O(\frac{n^2}{2k})$
CBFS–MHM	$O(\frac{n^2}{2k})$

5 Empirical comparative analysis

Before moving on to results on real datasets, we will use the toy dataset of Fig. 3 to illustrate how MD–MHM, MDAV–MHM and CBFS–MHM can dramatically reduce information loss (measured as within-groups sum of squares *SSE*) with respect to the fixed-size microaggregation heuristics Maximum Distance, MDAV and CBFS, respectively. Table 4 gives the 3-partitions obtained with each method and their corresponding *SSE*. It can be seen that *SSE* is much lower in the 3-partitions resulting from methods using MHM. In fact, MD–MHM, MDAV–MHM and CBFS–MHM all yield the same 3-partition, which turns out to be the optimal one for this particular dataset; it can be checked by exhaustive search that no other partition exists with $SSE < 9.06$.

Next, we present the real and simulated datasets that we have employed to compare the performance of the new heuristics using MHM with previous proposals in the literature. Three reference datasets [4] proposed in the European project CASC have been used:

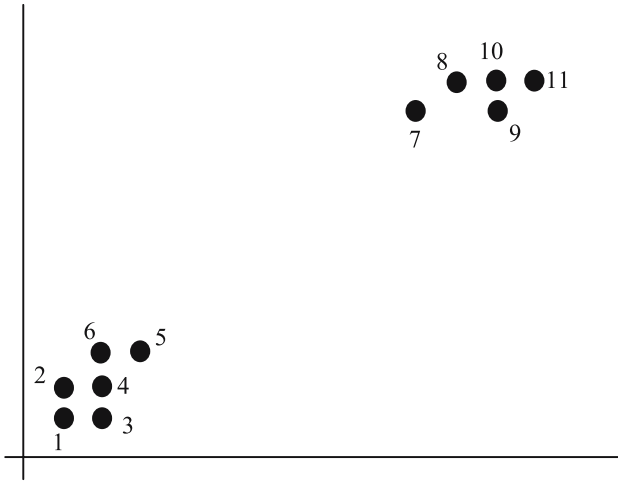

Fig. 3 Example bivariate dataset to illustrate the benefits of using MHM

Table 4 Maximum distance, MDAV and CBFS versus MD–MHM, MDAV–MHM and CBFS–MHM for the dataset in Fig. 3 with $k = 3$

Method	3-partition obtained	SSE
Maximum distance	{{1,2,3},{4,5,6,7,8},{9,10,11}}	178.67
MDAV	{{1,2,3},{4,5,6,7,8},{9,10,11}}	178.67
CBFS	{{1,2,3,4,6},{5,7,8},{9,10,11}}	92.00
MD–MHM	{{1,2,3},{4,5,6},{7,8,9,10,11}}	9.06
MDAV–MHM	{{1,2,3},{4,5,6},{7,8,9,10,11}}	9.06
CBFS–MHM	{{1,2,3},{4,5,6},{7,8,9,10,11}}	9.06

1. The "Tarragona" dataset contains 834 records with 13 numerical attributes. This dataset was used in CASC and in [10,23].
2. The "Census" dataset contains 1,080 records with 13 numerical attributes. This dataset was used in CASC and in [7,11,13,23,42].
3. The "EIA" dataset contains 4,092 records with 11 numerical attributes (plus two additional categorical attributes not used here). This dataset was used in CASC, in Dandekar et al. [7] and partially in Laszlo and Mukherjee [23] (an undocumented subset of 1080 records from "EIA", called "Creta" dataset, was used in the latter paper).

Larger datasets called "LargeCensus" and "Very LargeCensus", with 10^4 and 10^5 records, respectively, were also tried. These are synthetic datasets with the same covariance matrix as "Census", which have been generated using the synthetic data generator described in Mateo-Sanz et al. [28], a variant of the IPSO generator [5].

Additionally, two tests were carried out using entirely simulated datasets:

1. A test was performed on 25 uniform random datasets. Each dataset consisted of $n = 1000$ records with $p = 10$ numerical attributes. These are referred to as "SimU" datasets. Attribute values were drawn from the $[-10,000, 10,000]$ interval by simple random sampling.
2. A second test was performed on 25 random clustered datasets with $p = 10$ attributes. These are referred to as "SimC" datasets. Each random clustered dataset was generated as follows:
   (a) First, 100 points $x_1, \ldots, x_{100}$ were randomly selected in $[-10,000, 10,000]^{10}$;
   (b) For $i = 1$ to 100:
      (i) A cluster cardinality c_i was chosen as a uniform random number c_i between 3 and 20;
      (ii) c_i vectors starting at the origin and ending at a random point in $[-50, 50]^{10}$ were generated;
      (iii) The c_i vectors were added to x_i to get an i-th cluster with $c_i + 1$ points.
   (c) Let n be the total number of points generated in the preceding steps. To prevent the dataset from looking too perfectly clustered, $n/3$ points were randomly chosen in $[-10,000, 10,000]^{10}$ and added to the dataset; in this way, there were some points midway between clusters.

The comparison embraced eight different heuristic methods for multivariate microaggregation, including the new proposals in this paper:

- The new heuristics presented here, i.e., NPN–MHM, MD–MHM, MDAV–MHM and CBFS–MHM.
- The heuristics maximum distance, MDAV and CBFS.
- The minimum spanning tree heuristic M-d from Laszlo and Mukherjee [23], which is the best performer in that paper. In M-d, a minimum spanning tree (MST) is computed in $O(n^2)$ time for a dataset with n points; then the MST is cut to form clusters; finally, clusters with more than $2k$ points are partitioned using the maximum distance heuristic, which yields the k-partition used for microaggregation. Comparable results in Laszlo and Mukherjee [23] are only for the "Tarragona" and the "Census" datasets.

Table 5 shows the information loss for several values of k and for the various datasets and the various methods listed above (methods were used to jointly microaggregate all attributes in the datasets). Information loss is measured as $100 \times SSE/SST$, where SST is the total sum of squares of the dataset; note that the within-groups sum of squares SSE is never greater than SST so that the reported information loss measure takes values in the range $[0, 100]$. For the datasets "SimU" and "SimC", the information loss measure is the average over 25 simulated datasets, so we specify also the standard deviation of the information loss over the 25 datasets (figures after the $\pm$ symbol).

We can observe from Table 5 that

- For "Tarragona", "Census", "LargeCensus" and "VeryLargeCensus", the data-oriented heuristics MD–MHM, MDAV–MHM and CBFS-MHM did not really improve on their maximum distance, MDAV and CBFS fixed-size counterparts. This is due to the fact that these are datasets with little skewness: thus, all dataset points are quite homogeneous and allowing variable-sized groups is no big difference with respect to fixed-sized groups. Still, the three new heuristics were (by a very small margin) the best performers for both datasets and for all values of k considered. In particular, for "Tarragona" and "Census" (except for $k = 3$ in "Tarragona") the minimum spanning tree-based heuristic M-d from Laszlo and Mukherjee [23] was outperformed by maximum distance, MDAV, CBFS, MD–MHM, MDAV–MHM and CBFS–MHM.
- For the more skewed "EIA" dataset, the improvement of MD–MHM, MDAV–MHM and CBFS–MHM on their fixed-size counterparts was more

Table 5 Information loss $100 \times SSE/SST$ for several values of k, several datasets and several microaggregation heuristics

Dataset	Method	$k=3$	$k=5$	$k=10$
Tarragona	NPN–MHM	17.3949	27.0213	40.1831
	MD	16.9835	22.5273	33.1834
	MD–MHM	16.9829	22.5269	**33.1834**
	MDAV	16.9326	22.4619	33.1929
	MDAV–MHM	16.9326	**22.4617**	33.1923
	CBFS	16.9741	22.8277	33.2188
	CBFS–MHM	16.9714	22.8227	33.2188
	M-d	**16.6300**	24.5000	38.5800
Census	NPN–MHM	6.3498	11.3443	18.7335
	MD	5.71999	9.00648	14.3965
	MD–MHM	5.69724	8.98594	14.3965
	MDAV	5.6922	9.0884	14.2239
	MDAV–MHM	**5.6523**	9.0870	14.2239
	CBFS	5.6800	8.9055	13.8963
	CBFS–MHM	5.6734	**8.8942**	**13.8925**
	M-d	6.1100	10.3000	17.1700
EIA	NPN–MHM	0.5525	**0.9602**	**2.3188**
	MD	0.4723	1.6694	3.7141
	MD–MHM	0.4422	1.2627	3.6374
	MDAV	0.4831	1.6782	3.8445
	MDAV + MHM	**0.4081**	1.2563	3.7725
	CBFS	0.4831	1.7482	3.5445
LargeCensus	NPN–MHM	29.2775	**34.5031**	44.8446
	MD	25.6557	34.9761	43.2482
	MD–MHM	25.6496	34.9679	43.2437
	MDAV	25.6495	34.9868	43.1929
	MDAV–MHM	25.6441	34.9806	**43.1912**
	CBFS	25.6422	34.9729	43.2161
	CBFS–MHM	**25.6403**	34.9703	43.2164
VeryLargeCensus	NPN–MHM	24.2500	29.3982	33.4028
	MDAV	20.4097	25.2550	30.1285
	MDAV–MHM	**20.4002**	**25.2536**	**30.1281**
SimU	NPN–MHM	**18.0768 ± 0.2650**	29.6716 ± 0.4151	48.9056 ± 0.7414
	MD	18.2494 ± 0.2509	28.3524 ± 0.4401	41.0628 ± 0.5109
	MD–MHM	18.1665 ± 0.2701	28.2456 ± 0.4270	40.9986 ± 0.5089
	MDAV	18.2215 ± 0.2373	28.1793 ± 0.3521	41.1682 ± 0.5376
	MDAV–MHM	18.1264 ± 0.2368	28.0877 ± 0.3481	41.1050 ± 0.5261
	CBFS	18.1652 ± 0.2262	28.1274 ± 0.3047	40.7392 ± 0.3128
	CBFS–MHM	18.1297 ± 0.2221	**28.0700 ± 0.2882**	**40.7104 ± 0.3155**
SimC	NPN–MHM	**2.6561 ± 0.3126**	**6.2457 ± 0.6236**	**14.5791 ± 0.8726**
	MD	4.3729 ± 0.2825	7.1045 ± 0.4186	16.0232 ± 1.1086
	MD–MHM	3.4268 ± 0.2675	6.7640 ± 0.4446	15.5769 ± 1.0383
	MDAV	4.3407 ± 0.3002	7.0465 ± 0.3945	16.2478 ± 0.9997
	MDAV–MHM	3.3597 ± 0.2660	6.6788 ± 0.4073	15.8361 ± 0.9396
	CBFS	4.3434 ± 0.2774	7.0305 ± 0.4024	15.7962 ± 1.0066
	CBFS–MHM	3.5416 ± 0.2054	6.6857 ± 0.4185	15.5336 ± 0.9515

Lowest information losses are shown in boldface

obvious. For larger k, even NPN–MHM outperformed MD–MHM, MDAV–MHM and CBFS–MHM. This picture changes a bit if we take speed into account: NPN–MHM, MDAV–MHM and CBFS–MHM are faster because they run in $O(n^2)$ time, while MD–MHM runs in $O(n^3)$ time.

- For the simulated datasets "SimU" and "SimC", MD–MHM, MDAV–MHM and CBFS–MHM clearly improved on their fixed-size counterparts. While for "SimU", MDAV–MHM and CBFS–MHM were best among the methods tried, they were substantially outperformed by NPN–MHM for "SimC" (clustered data).
- We can conclude that, being data-oriented, the new heuristics really show their usefulness for skewed and or clustered datasets, but they still behave pretty well for non-clustered, homogeneous datasets. For skewed data or clustered data with moderate gaps, such as the "EIA" dataset, MDAV–MHM and CBFS–MHM are probably the methods yielding the best

Table 6 Number of optimal k-partitions. Maximum and average ratios between SSE_{method} and the optimal SSE_{opt}

Number of optimal partitions ($\frac{SSE_{\text{method}}}{SSE_{\text{opt}}} = 1$)							
k	NPN–MHM	CBFS	CBFS–MHM	MDAV	MDAV–MHM	Max. Dist.	MD–MHM
2	42	38	54	15	45	14	43
3	46	46	57	10	39	10	34
5	52	45	52	37	45	37	45
Maximum($\frac{SSE_{\text{method}}}{SSE_{\text{opt}}}$)							
k	NPN–MHM	CBFS	CBFS–MHM	MDAV	MDAV–MHM	Max. Dist.	MD + MHM
2	2.55	2.26	1.82	2.87	2.01	2.49	2.01
3	1.66	1.68	1.68	1.99	1.7	1.91	1.9
5	1.50	1.47	1.11	1.48	1.13	1.48	1.48
Average($\frac{SSE_{\text{method}}}{SSE_{\text{opt}}}$)							
k	NPN-MHM	CBFS	CBFS–MHM	MDAV	MDAV–MHM	Max. Distance	MD–MHM
2	1.29 ± 0.33	1.25 ± 0.27	1.16 ± 0.20	1.51 ± 0.38	1.21 ± 0.25	1.51 ± 0.37	1.21 ± 0.25
3	1.18 ± 0.18	1.17 ± 0.16	1.12 ± 0.14	1.36 ± 0.20	1.18 ± 0.16	1.37 ± 0.21	1.20 ± 0.17
5	1.11 ± 0.10	1.12 ± 0.10	1.10 ± 0.10	1.14 ± 0.11	1.13 ± 0.11	1.46 ± 0.11	1.13 ± 0.11

trade-off between low information loss and performance. However, when clusters are very separated, as in "SimC", NPN–MHM is the best option.

Regarding computing times, we measured them on a Centrino processor at 1.4 GHz running under a Linux operating system. For $k = 3$, it took about one second to run the quadratic heuristics (NPN–MHM, MDAV, MDAV–MHM, CBFS and CBFS–MHM) on "Census" and "EIA". The cubic heuristics MD and MD–MHM took one second on "Census", but on "EIA" they took 30 and 32 s., respectively. On a large dataset like "Very-LargeCensus" and also for $k = 3$, the quadratic heuristics were still affordable: 721 s. for NPN–MHM, 1,421 seconds for MDAV and 1,450 s. for MDAV–MHM. However, as explained in Note 3, microaggregation is normally used by statistical agencies after blocking large datasets into smaller sub-datasets of a few thousands of records (about the size of "EIA").

Finally, we give some experimental results on how close to optimality are the partitions obtained using the new heuristics and their fixed-size counterparts. We simulated 100 datasets consisting of 11 records and 3 numerical attributes. Attributes were generated by drawing from a normal distribution $N(0, 5,000)$. For each dataset, the optimal k-partition was found by exhaustive search for $k = 2, 3, 5$.

Table 6 shows, for $k = 2, 3, 5$:

- The number of datasets (out of 100) for which the optimal k-partition was obtained with each method.
- The maximum ratio $SSE_{\text{method}}/SSE_{\text{opt}}$ for the 100 datasets, that is, the maximum ratio between the within-groups sum of squares yielded by a certain method and the optimum within-groups sum of squares. The closer that maximum to 1, the better.
- The average value $SSE_{\text{method}}/SSE_{\text{opt}}$ for the 100 datasets. The closer that average to 1, the better.

The following can be observed from Table 6:

- For all three criteria in the table, the new data-oriented heuristics MD–MHM, MDAV–MHM and CBFS–MHM clearly outperformed their fixed-size counterparts. Among the new heuristics, CBFS-MHM was the best performer, closely followed by MDAV-MHM.
- For the third criterion (average $SSE_{\text{method}}/SSE_{\text{opt}}$ ratio), NPN–MHM reached a performance similar to the more sophisticated MD–MHM, MDAV–MHM and CBFS–MHM.

The above observations on how close we get to optimality still held when we considered several different numbers of attributes. Table 7 gives the number of optimally 3-partitioned datasets (that is, $k = 3$) and the average $SSE_{\text{method}}/SSE_{\text{opt}}$ when the number of attributes was $p = 2, 3, 5, 7$.

6 Conclusions

Microaggregation is an SDC technique used world-wide to preserve the privacy of respondents contributing to microdata sets. The level of privacy required is controlled by a parameter k (minimum group size). Once k has been chosen, the data protector (and the data users) is interested in minimizing information loss. We have shown in this paper how Hansen–Mukherjee's algorithm for optimal univariate microaggregation can be used to enhance the existing heuristics for multivariate microaggregation, so as to reduce information loss.

Table 7 Number of optimal 3-partitions and average ratio between SSE_{method} and the optimal SSE_{opt} for several numbers of attributes

Number of optimal partitions ($\frac{SSE_{\text{method}}}{SSE_{\text{opt}}} = 1$)							
p	NPN–MHM	CBFS	CBFS–MHM	MDAV	MDAV–MHM	Max. Dist.	MDAV–MHM
2	54	55	72	11	44	8	10
3	46	46	57	10	39	10	34
5	41	48	58	18	42	15	41
7	52	38	57	14	44	14	50
Average($\frac{SSE_{\text{method}}}{SSE_{\text{opt}}}$)							
p	NPN–MHM	CBFS	CBFS–MHM	MDAV	MDAV–MHM	Max. Dist.	MD–MHM
2	1.26 ± 0.40	1.21 ± 0.39	1.21 ± 0.31	1.61 ± 0.43	1.28 ± 0.33	1.61 ± 0.45	1.29 ± 0.32
3	1.18 ± 0.18	1.17 ± 0.16	1.12 ± 0.14	1.36 ± 0.20	1.18 ± 0.16	1.37 ± 0.21	1.20 ± 0.17
5	1.14 ± 0.10	1.13 ± 0.12	1.10 ± 0.10	1.22 ± 0.14	1.14 ± 0.13	1.23 ± 0.13	1.13 ± 0.12
7	1.11 ± 0.09	1.13 ± 0.09	1.09 ± 0.08	1.19 ± 0.09	1.11 ± 0.07	1.19 ± 0.09	1.09 ± 0.07

The basic idea is to use fixed-size heuristics or other algorithms such as NPN to construct a path traversing all points in a multivariate dataset; then the MHM multivariate adaptation of Hansen–Mukherjee's algorithm is used on that path. The result is a data-oriented heuristic yielding the best possible k-partition compatible with the ordering induced by the path.

Since MHM runs in linear time, plugging it after a fixed-size microaggregation heuristic is not a computational burden, because such heuristics run in at least quadratic time. Thus, plugging MHM is a fast way of turning a fixed-size microaggregation into a data-oriented, variable-sized heuristic running in quadratic time. Other data-oriented heuristics in the literature also run in quadratic time [23].

For very homogeneous datasets (without obvious clusters), the MHM-enhanced heuristics display a performance as good as their fixed-sized counterparts regarding information loss and nearly as good regarding speed. For mildly skewed or clustered datasets, MDAV–MHM and CBFS–MHM do pretty well in exploiting their data-orientedness to reduce information loss with respect to fixed-size alternatives. For heavily skewed or clustered datasets, NPN–MHM is the best heuristic among those presented.

Acknowledgements The presentation of this paper was substantially improved thanks to the comments and suggestions by two anonymous referees. This work has been partly supported by the Spanish Ministry of Science and Education under project SEG2004-04352-C04-01 "PROPRIETAS" and by the Government of Catalonia under grant 2005 SGR 00446.

References

1. Agrawal, D., Aggarwal, C.C.: On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the Symposium on Principles of Database Systems-PODS'2001, Santa Barbara. Association for Computing Machinery, (2001)
2. Boyens, C., Krishnan, R., Padman, R.: On privacy-preserving access to distributed heterogeneous healthcare information. In: Proceedings of the 37th Hawaii International Conference on System Sciences HICSS-37, Big Island, HI IEEE Computer Society (2004)
3. Brand, R.: Microdata protection through noise addition. In: Domingo-Ferrer, J. (ed.) Inference Control in Statistical Databases, vol 2316 of LNCS, pp. (97–116). Springer, Berlin Heidelberg New York (2002)
4. Brand, R., Domingo-Ferrer, J., Mateo-Sanz, J.M.: 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 (2002)
5. Burridge, J.: Information preserving statistical obfuscation. Stat. Comput. **13**, 321–327 (2003)
6. Dalenius, T.: Finding a needle in a haystack–or identifying anonymous census records. J. Official Stat. **2**3, 329–336 (1986)
7. Dandekar, R., Domingo-Ferrer, J., Sebé, F.: LHS-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection. In: Domingo-Ferrer, J. (ed.) Inference Control in Statistical Databases, vol. 2316 of LNCS, pp. 153–162. Springer, Berlin Heidelberg NewYork (2002)
8. Defays, D., Anwar, N.: Micro-aggregation: a generic method. In: Proceedings of the 2nd International Symposium on Statistical Confidentiality, pp. 69–78. Eurostat, Luxemburg (1995)
9. Defays, D., Nanopoulos, P.: Panels of enterprises and confidentiality: the small aggregates method. In: Proceedings of 1992 Symposium on Design and Analysis of Longitudinal Surveys, pp. 195–204. Statistics Canada, Ottawa (1993)
10. Domingo-Ferrer, J., Mateo-Sanz, J.M. : Practical data-oriented microaggregation for statistical disclosure control. IEEE Trans. Knowl. Data Eng. 14(1), 189–201 (2002)
11. Domingo-Ferrer, J., Mateo-Sanz, J.M., Torra, V.: Comparing SDC methods for microdata on the basis of information loss and disclosure risk. In: Pre-proceedings of ETK-NTTS'2001 (vol. **2**), pp. 807–826. Luxemburg, Eurostat (2001)
12. Domingo-Ferrer, J., Torra, V.: A quantitative comparison of disclosure control methods for microdata. In: Doyle P., Lane J.I., Theeuwes J. J. M., Zayatz, L. (eds.) Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, pp. 111–134. Amsterdam North-Holland, http://vneumann.etse.urv.es/publications/bcpi (2001)
13. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogenerous k-anonymity through microaggregation. Data Mining Knowl. Discov. **11**(2),195–212 (2005)
14. Doyle, P., Lane, J.I., Theeuwes, J.J., Zayatz, L.V.: (eds). Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies. North-Holland, Amsterdam (2001)

15. Edwards, A.W.F., Cavalli-Sforza, L.L.: A method for cluster analysis. Biometrics, **21**, 362–375 (1965)
16. Gordon, A.D., Henderson, J.T.: An algorithm for Euclidean sum of squares classification. Biometrics, **33**, 355–362 (1977)
17. Hansen, P., Jaumard, B., Mladenovic, N.: Minimum sum of squares clustering in a low dimensional space. J. Classifi. **15**, 37–55, (1998)
18. Hansen, S.L., Mukherjee, S.: A polynomial algorithm for optimal univariate microaggregation. IEEE Trans. Knowl. Data Eng. **15**(4), 1043–1044 (2003)
19. Hartigan, J.A.: Clustering Algorithms. Wiley, New York (1975)
20. Hundepool, A., Van de Wetering, A., Ramaswamy, R., Franconi, L., Capobianchi, A.: DeWolf, P.-P., Domingo-Ferrer, J., Torra, V., Brand, R., Giessing, S.: μ-ARGUS version 3.2 Software and User's Manual. Statistics Netherlands, Voorburg NL, http://neon.vb.cbs.nl/casc (2003)
21. Hundepool, A., Van de Wetering, A., Ramaswamy, R., Franconi, L., Capobianchi, A., DeWolf, P.-P., Domingo-Ferrer, J., Torra, V., Brand, R., Giessing, S.: μ-ARGUS version 4.0 Software and User's Manual. Statistics Netherlands, Voorburg NL, http://neon.vb.cbs.nl/casc (2005)
22. Jancey, R.C.: Multidimensional group analysis. Aust. J. Bot. **14**, 127–130 (1966)
23. Laszlo, M., Mukherjee, S.: Minimum spanning tree partitioning algorithm for microaggregation. IEEE Trans. Knowl. Data Eng. **17**(7), 902–911 (2005)
24. Lenz, R., Vorgrimler, D.: Matching German turnover tax statistics. In: Technical Report FDZ-Arbeitspapier Nr. 4, Statistische Ämter des Bundes und der Länder-Forschungsdatenzentren (2005)
25. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol., **1**, 281–297 (1967)
26. Mateo-Sanz, J.M., Domingo-Ferrer, J.: A method for data-oriented multivariate microaggregation. In: Domingo-Ferrer, J., (ed.) Statistical Data Protection, (pp. 89–99) Luxemburg, (1999) Office for Official Publications of the European Communities
27. Mateo-Sanz, J.M., Domingo-Ferrer, J.: Heuristic techniques for multivariate microaggregation. In: COMPSTAT'2000, Utrecht. CBS-Statistics, Netherlands (2000)
28. Mateo-Sanz, J.M., Martínez-Ballesté, A., Domingo-Ferrer, J.: Fast generation of accurate synthetic microdata. In: Domingo-Ferrer, J., Torra, V., (eds.) Privacy in Statistical Databases, volume 3050 of LNCS, (pp.298–306) Springer, Berlin Heidelberg New York (2004)
29. Oganian, A., Domingo-Ferrer, J.: On the complexity of optimal microaggregation for statistical disclosure control. Stat. J. United Nat. Econ. Com. Eur. **18**(4), 345–354 (2001)
30. Pagliuca, D., Seri, G.: Some results of the individual ranking method on the system of enterprise accounts annual survey. In: Technical report, ESPRIT SDC Project, Deliverable MI-3/D2.11 (1999)
31. Rosemann, M.: Erste Ergebnisse von vergleichenden Untersuchungen mit anonymisierten und nicht anonymisierten Einzeldaten am Beispiel der Kostenstrukturerhebung und der Umsatzsteuerstatistik. In: Ronning, G., Gnoss, R., (eds.), Anonymisierung wirtschaftsstatistischer Einzeldaten, (pp.154–183) Wiesbaden, Germany, Statistisches Bundesamt (2003)
32. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Trans. Know. and Data Eng. **13**(6), 1010–1027 (2001)
33. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: *k*-anonymity and its enforcement through generalization and suppression. In: Technical report, SRI International, (1998)
34. Sande, G.: Exact and approximate methods for data directed microaggregation in one or more dimensions. Int. J. Uncert. Fuzziness Know. Based Sys. **10**(5), 459–476 (2002)
35. Sweeney, L.: *k*-Anonimity: a model for protecting privacy. Int. J. Uncert. Fuzziness Knowl. Based Sys. **10**(5), 557–570 (2002)
36. Torra, V.: Microaggregation for categorical variables: a median based approach. In: Domingo-Ferrer, J., Torra, V., (eds.), Privacy Stat. Databases vol. 3050 of LNCS, (pp.162–174) Springer, Berlin Heidelberg New York (2004)
37. Torra, V., Domingo-Ferrer, J.: Record linkage methods for multidatabase data mining. In: Torra, V. (eds.) Information Fusion in Data Mining, (pp.101–132) Springer, Germany (2003)
38. UNECE. United Nations Economic Commission for Europe: Questionnaire on disclosure and confidentiality–summary of replies. In: 2nd Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, Skopje, Macedonia (2001)
39. UNECE. United Nations Economic Commission for Europe: 2003 Questionnaire on statistical confidentiality – summary of replies from Central and Eastern Europe. In: 4th Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, Luxemburg (2005)
40. Ward, J.H.: Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. **58**, 236–244 (1963)
41. Willenborg, L., DeWaal, T.: Elements of Statistical Disclosure Control. Springer. Berlin Heidelberg New York (2001)
42. Yancey, W.E., Winkler, W.E., Creecy, R.H.: Disclosure risk assessment in perturbative microdata protection. In: Domingo-Ferrer, J., (ed.) Inference Control in Statistical Databases, vol. 2316 of LNCS (pp.135–152) Springer, Berlin Heidelberg New York (2002)