Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 55 (2008) 714–732
An International Journal
computers & mathematics
with applications
www.elsevier.com/locate/camwa

A polynomial-time approximation to optimal multivariate microaggregation

Josep Domingo-Ferrer*, Francesc Sebé, Agusti Solanas

Rovira i Virgili University of Tarragona, UNESCO Chair in Data Privacy, Department of Computer Engineering and Maths, Av. Països Catalans 26, Tarragona, Catalonia, Spain

Received 24 November 2006; received in revised form 8 March 2007; accepted 3 April 2007

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 without disclosing private information on the underlying individuals. Microaggregation techniques are currently being used by many statistical agencies. The principle of microaggregation is to group original database records into small aggregates prior to publication. Each aggregate should contain at least k records to prevent disclosure of individual information, where k is a constant value preset by the data protector. In addition to it being a good masking method, microaggregation has recently been shown useful to achieve k-anonymity. In k-anonymity, the parameter k specifies the maximum acceptable disclosure risk, so that, once a value for k has been selected, the only job left is to maximize data utility: if microaggregation is used to implement k-anonymity, maximizing utility can be achieved by microaggregating optimally, *i.e.* with minimum within-groups variability loss. Unfortunately, optimal microaggregation can only be computed in polynomial time for univariate data. For multivariate data, it has been shown to be NP-hard. We present in this paper a polynomial-time approximation to microaggregate multivariate numerical data for which bounds to optimal microaggregation can be derived at least for two different optimality criteria: minimum within-groups Euclidean distance and minimum within-groups sum of squares. Beyond the theoretical interest of being the first microaggregation proposal with proven approximation bounds for any k, our method is empirically shown to be comparable to the best available heuristics for multivariate microaggregation.

Keywords: Microaggregation; Statistical databases; Privacy; Microdata protection; Statistical disclosure control; Approximation algorithms; k-anonymity

1. Introduction

Releasing statistical databases for public use can jeopardize the privacy of the individual respondents on which the statistical data were collected. Statistical disclosure control (SDC), also known as statistical disclosure limitation (SDL), seeks to transform statistical data in such a way that they can be publicly released while preserving data utility and statistical confidentiality. The latter refers to avoiding disclosure of information that can be linked to specific

* Corresponding author.
E-mail addresses: josep.domingo@urv.cat (J. Domingo-Ferrer), francesc.sebe@urv.cat (F. Sebé), agusti.solanas@urv.cat (A. Solanas).

0898-1221/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.04.034

individual or corporate respondent entities. SDC is important for legal, ethical and economical reasons, as illustrated by the following application domains:

- *Official statistics*. Most countries have legislation which compels national statistical institutes to guarantee statistical confidentiality when they release data collected from citizens or companies. This justifies the research on SDC undertaken by several countries, including the United States, Canada and the European Union (*e.g.* the CASC project [1]).
- *Health information*. This is one of the most sensitive areas regarding privacy. For example, in the U. S., the Privacy Rule of the Health Insurance Portability and Accountability Act [2] requires the strict regulation of protected health information for use in medical research. In most western countries, the situation is similar, which has caused e-health to become a hot topic in privacy research (see *e.g.* [3]).
- *E-commerce*. Electronic commerce results in the automated collection of large amounts of consumer data. This wealth of information is very useful to companies, which are often interested in sharing it with their subsidiaries or partners. Such consumer information transfer should not result in public profiling of individuals and is subject to strict regulation; see [4] for regulations in the European Union and [5] for regulations in the U.S.

To protect against disclosure, SDC techniques normally perform some data modification, which, for the sake of utility, cannot go as far as to completely encrypt or scramble data. The *SDC problem* is to solve the inherent tradeoff between disclosure risk and information loss: data must be modified in such a way that sufficient protection is provided while minimizing information loss, *i.e.* the loss of the accuracy sought by database users. SDC techniques can be classified as follows depending on the format of data they intend to protect:

- *Tabular data protection*. This is the oldest and best established part of SDC, because tabular data have been the traditional output of national statistical institutes. The goal here is to publish tables with *static* aggregate information, in such a way that they do not leak any confidential information on specific individuals. See [6] for a conceptual survey and [7] for a survey on practical tabular protection methods.
- *Queried databases*. The scenario here is a database to which the user can submit statistical queries (sums, averages, etc.). The aggregate information obtained by a user as a result of successive queries should not allow him to infer information on specific individuals. Since the 80s, this has been known to be a difficult problem, subject to the tracker attack [8]. Currently employed strategies rely on perturbing, restricting or replacing at intervals the answers to certain queries. Examples of these three strategies can be found in [9–11], respectively.
- *Microdata protection*. This subdiscipline is about protecting static individual data, also called microdata. It is only recently that data collectors (statistical agencies and the like) have been persuaded to publish microdata. The current prevalence of microdata and the short history of microdata protection make research in this area especially necessary.

This paper deals with microdata protection and, specifically, with a family of SDC methods for numerical microdata known as microaggregation. Microaggregation was proposed at Eurostat [12] in the early nineties, and has since then been used in Italy [13], Germany [14,15] and several other countries [16]. The remainder of this introduction gives some high-level concepts on microdata protection with a view to justifying the relevance of microaggregation as an SDC method; then, the basics of microaggregation are recalled; a reminder on the recently realized usefulness of microaggregation for k-anonymity follows; the introduction concludes with a sketch of the contribution and the plan of the rest of this paper.

1.1. Microdata protection concepts

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

- *Identifiers*. These are attributes that *unambiguously* identify the respondent. Examples are passport number, social security number, full name, etc. Since our objective is to prevent confidential information from being linked to specific respondents, we will assume in what follows that, in a pre-processing step, identifiers in V have been removed/encrypted.

Table 1
Running example: SME data

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

"Company name" is an identifier to be suppressed before publishing the dataset.

- *Quasi-identifiers*. Borrowing the definition from [17,18], a quasi-identifier is a set of attributes in V 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 quasi-identifier attributes are age, gender, job, zipcode, etc. Unlike identifiers, quasi-identifiers cannot be removed from V. The reason is that any attribute in V potentially belongs to a quasi-identifier (depending on the external data sources available to the user of V). Thus one would need to remove all attributes (!) to make sure that the dataset no longer contains quasi-identifiers.
- *Confidential outcome attributes*. These are attributes which contain sensitive information on the respondent. Examples are salary, religion, political affiliation, health condition, etc.
- *Non-confidential outcome attributes*. Those 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 be part of a quasi-identifier. 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 (this problem is less likely if attributes with coarser categories are used, *e.g.* "Region" instead of "Town of residence").

Example 1. The dataset in Table 1 will be used as a running example throughout this paper. For each of 11 small or medium enterprises (SMEs) in a certain town, the table gives the company name, the surface in square meters of the company's premises, its number of employees, its turnover and its net profit. The number of records in the dataset is unrealistically small, but this has the advantage of allowing the entire dataset and its protected versions to be listed within the paper, which greatly facilitates understanding the operation of algorithms. Results on real datasets are reported in Section 6 below. The company name is normally a good identifier (even if it is not impossible for two companies to have the same name). We will consider that turnover and net profit are confidential outcome attributes. Furthermore, we will assume that quasi-identifier attributes are limited to the surface of the company's premises and its number of employees. Indeed, it is easy for any snooper 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. □

The purpose of microdata SDC mentioned in the previous section can be stated more formally by saying that, given an original microdata set V, the goal is to release a protected microdata set V' in such a way that:

(1) Disclosure risk (*i.e.* the risk that a user or a snooper can use V' to determine confidential attributes on a specific individual among those in V) is low.
(2) User analyses (regressions, means, etc.) on V' and V yield the same or at least similar results. This is equivalent to requiring that information loss caused by SDC should be low, *i.e.* that the utility of the SDC-protected data should stay high.

Table 2
Multivariate microaggregation heuristics vs other microdata protection methods

	Multivariate microaggregation heuristics	Rank swapping	Simple noise addition	Noise addition with nonlinear transformation
Tradeoff data utility vs disclosure risk	Good	Good	Poor	Good
Computational complexity	Low	Very low	Very low	High
Preservation of attribute relationships	Good	Poor	Poor	Very good
Implementability	Easy	Easy	Easy	Difficult
Understandability	Easy	Easy	Easy	Difficult

Sources: [22,24].

Masking, *i.e.* generating a modified version V' of the original microdata set V is the most usual approach in microdata protection. Normally, masking is targeted at the quasi-identifier attributes, in order to prevent re-identification. Masking methods can in turn be divided in two categories depending on their effect on the original data:

- *Non-perturbative*. Non-perturbative methods protect data without distorting them. They rely on the principles of suppression and generalization [18,6]. Suppression consists of deleting some of the original data before publication. Generalization, also known as recoding, consists of coarsening the granularity of data: numerical data are categorized and categorical data are recoded using broader, more general categories. Thus, for the case of numerical data, non-perturbative methods cause numerical (continuous) attributes to become either incomplete or categorical.
- *Perturbative*. The microdata set is distorted before publication. The perturbation method used should be such that statistics computed on the perturbed dataset do not differ significantly from the statistics that would be obtained on the original dataset. Microaggregation, rank swapping and noise addition are examples of perturbative methods (see [6,19–21] for further information).

In [22], SDC methods for numerical microdata were compared using a score combining measures of data utility loss (basically the impact on first and second-order moments) and disclosure risk. Multivariate microaggregation and rank swapping [23] were identified as the best performing methods, that is, the ones whose application offered the best tradeoff between low data utility loss and low disclosure risk. Subsequent work has shown that some sophisticated forms of noise addition (*e.g.* noise addition with non-linear transformation, see [24] for an introduction) can match the performance of both rank swapping and microaggregation. If we take ease of implementation as an additional criterion, it turns out that multivariate microaggregation heuristics and rank swapping are far easier to implement, use and understand than noise addition methods similar in performance [24]. A difference between microaggregation and rank swapping is that multivariate microaggregation deals with several attributes at a time, unlike rank swapping which deals with one attribute at a time. Thus, as a rule, multivariate microaggregation is better at preserving the relationships between attributes (correlations, etc.).

To sum up, while there exist useful masking methods other than microaggregation, the latter has certain advantages in applications where limited data utility loss in quasi-identifier attributes is acceptable. The above comparison criteria are summarized in Table 2.

1.2. Basics of microaggregation

Microaggregation was originally designed for continuous numerical data [12,25] and recently extended for categorical data [26,27]. 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 or the median for categorical data) is used to compute a centroid for each group. Then, each record in a group is replaced by the group centroid.

In the above two steps, one may also use the projections of the records on a particular set of attributes rather than the entire records (in fact this is the most usual practice, see Section 1.3 below).

In [25], optimal microaggregation is defined as the one yielding a k-partition maximizing the within-group 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 n records containing p attributes to be microaggregated. A record may contain additional attributes which will not be microaggregated or will be microaggregated separately, but we will ignore these for the sake of simplicity: in what follows, we consider records with p attributes all of which are microaggregated together. 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 resulting number of 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 [28–33]. In terms of sums of squares, maximizing within-groups homogeneity is equivalent to finding a k-partition minimizing the within-groups sums of squares SSE defined as

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

where the number g of groups in the k-partition is not fixed and depends on the particular k-partition. It is shown in [25] that the sizes of groups in the optimal k-partition lie between k and $2k - 1$.

Microaggregating categorical data requires some adaptations described in [27]. 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 [34], 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 [35]. Unfortunately, realistic datasets are multivariate, so in practice multivariate microaggregation is heuristic [36,25,37–39].

1.3. Microaggregation and k-anonymity

The k-anonymity approach [40,18,41,42], is an elegant way of facing the conflict between information loss and disclosure risk in SDC.

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

As explained above, the attributes in a quasi-identifier are those used by a snooper to attempt re-identification, *i.e.* to link records in the protected dataset V' with identified records in an external data source S. If V' satisties k-anonymity, a snooper attempting record linkage with 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.

When k-anonymity was originally proposed as a concept, the suggested computational procedure to implement it was a combination of generalizations and suppressions. It has recently been shown in [27] that microaggregation provides a more unified computational way to reach k-anonymity. The point is that k-anonymity can be obtained by using a special case of microaggregation in which the set of attributes that are jointly microaggregated coincides with the set of quasi-identifier attributes. (Note that, in general, microaggregation as a masking method can be applied on any set of attributes, including quasi-identifier attributes and confidential outcome attributes; further, for a given set of attributes, microaggregation can be carried out jointly for all attributes in the set, independently for each

Table 3
Running example: 3-anonymous version of the SME dataset after optimal microaggregation of quasi-identifier 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

attribute – a variant known as individual ranking –, or jointly by disjoint subsets of attributes [22].) The advantages of microaggregation over generalization/suppression for k-anonymity are that: (i) microaggregation is applicable to any data type, including numerical data; (ii) microaggregation does *not* complicate analysis by returning a k-anonymous dataset with new categories or censored data; (iii) microaggregation does not turn numerical data into categorical data.

In k-anonymity, parameter k implicitly specifies a maximum acceptable disclosure risk: for a random record, the re-identification risk is upper-bounded by $1/k$, since there are at least k target records being equal concerning the k-anonymized attributes (for a specific record, the upper bound is the reciprocal of the number of its equal records; in the case of k-anonymity via microaggregation this can be as low as $1/(2k-1)$). Therefore, after a value for k has been selected, the job left is to maximize data utility: if microaggregation is used to implement k-anonymity, maximizing utility can be achieved by microaggregating optimally, *i.e.* with minimum within-groups variability loss. This is an extra motivation to find polynomial-time approximations to the NP-hard problem of optimal microaggregation. The case of microaggregation of numerical data is particularly interesting, because for this kind of data microaggregation stands out as the only known computational approach which can lead to k-anonymity while preserving the semantics of continuous numerical data [27].

Note 1. If in practice it turns out that the analytical utility of some k-anonymous microdata file is not sufficiently maintained, the data distributor cannot avoid further investigation in the usual trade-off between the two objectives "maximization of analytical utility" and "minimization of re-identification risk". Possible strategies include choosing a smaller k or trying to find a better estimation of the re-identification risk (*e.g.* via matching masked data with publicly available databases containing the necessary quasi-identifier attributes).

Example 2. Table 3 is a 3-anonymous version of the dataset in Table 1. We must note here that the ratio between the minimum group size k ($k = 3$) and the dataset size n ($n = 11$) lacks realism due to the small size of our example dataset. In datasets of realistic size (such as those in Section 6), one can afford to take $k \ll n$ (*e.g.* n at least two orders of magnitude greater than k) in order for the data coarsening caused by k-anonymity to stay reasonable.

The identifier "Company name" was suppressed and optimal bivariate microaggregation with $k = 3$ was used on the quasi-identifier attributes "Surface" and "No. employees" (in general, if there are p quasi-identifier 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 $\mathrm{SSE}_{\mathrm{opt}} = 7.484$. The total sum of squares – sum of squared Euclidean distances from all 11 pairs of standardized (surface, number of employees) to their average (0, 0) – is

$$\mathrm{SST} = \sum_{i=1}^{11} (\mathrm{surf}_i^2 + \mathrm{emp}_i^2) = 22$$

where surf_i and emp_i are, respectively, the standardized surface and number of employees of the i-th company.

Dividing SSE_{opt} by SST 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 3, 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.

In general, it could happen that a confidential attribute were constant within a group resulting from microaggregation, even if this is more likely for categorical confidential attributes than for numerical confidential attributes as the ones in this example. There are at least two ways out from such a pathological situation: (i) to inject some within-group variability for the confidential attribute via noise addition; (ii) to re-compute microaggregation using a different k or a different heuristic, in order to get different groups. □

1.4. Contribution and plan of the rest of this paper

In the original application of microaggregation as a masking method, the set of masked attributes is not necessarily the set of quasi-identifier attributes, but can include confidential outcome attributes; further, not all masked attributes need to be microaggregated together: individual attributes or groups of attributes can be masked independently of each other. In this setting, optimality understood as information loss minimization is not necessarily interesting, as it might result in unacceptable disclosure risk (*e.g.* this might happen if microaggregation is applied to the confidential outcome attributes). However, as argued in the previous subsection, the application of microaggregation to k-anonymity motivates a renewed interest in optimal microaggregation.

As mentioned above, the literature on multivariate microaggregation only provides heuristic methods. Empirical work [22] shows that some of those heuristics yield quite good solutions (*e.g.* [25,27,38,39]). However, no approximation algorithms to optimal microaggregation exist so far which, for any k, yield k-partitions whose SSE is *never* greater than a known multiple of the optimal SSE.

We present in the rest of this paper an approximation algorithm to optimal microaggregation of multivariate numerical data. The foundations of the new approximation algorithm are explained in Section 2. The approximation algorithm for microaggregation re-uses two procedures given in [43] for approximating optimal k-anonymity *via suppression* and is described in Section 3. In Section 4, the approximation error vs optimal microaggregation is bounded when the optimality criterion is to minimize the within-groups sums of Euclidean distances. In Section 5, a bound for the approximation error is derived when the optimality criterion is to minimize the within-groups sums of squares SSE. To complement the theoretical approximation bounds, Section 6 reports on empirical work to compare the typical variability loss of the new algorithm vs the best known heuristics for multivariate microaggregation. Practical implications and conclusions are summarized in Section 7. The Appendix contains the proofs of the lemmata in the paper plus some auxiliary lemmata for the proofs.

2. Background: Forests with trees of upper- and lower-bounded size

Given a set of records and a distance function between those records, a polynomial-time algorithm is given in [43] to k-anonymize those records *via suppression*. The resulting heuristic is an approximation to the optimal suppression pattern. This algorithm will be adapted in this paper to approximate optimal microaggregation. The algorithm creates a directed forest such that:

(1) Records are vertices;
(2) Each vertex has at most one outgoing edge;
(3) (u, v) is an edge only if v is one of the $k - 1$ nearest neighbors of u (according to the distance function);
(4) The size of every tree in the forest, *i.e.* the number of vertices in the tree, is between k and $\max(2k - 1, 3k - 5)$ (as mentioned below, these bounds are due to the constructions used).

The algorithm consists of two procedures, FOREST and DECOMPOSE-COMPONENT. The first one creates edges in such a way that no cycle arises; also, the out-degree of each vertex is at most one and the size of the trees in the resulting forest is at least k. The second procedure decomposes trees with size greater than $\max(2k - 1, 3k - 5)$ into

smaller component trees of size at least k. We reproduce both procedures from [43], with the slight adaptation that the weight of an edge between two records is the Euclidean distance between both records (in the original [43] version, the edge weight is the number of attributes in which both records differ). Such an adaptation is required to use those procedures for microaggregation rather than suppression.

Procedure 1 (*Forest*).

(1) *Start with an empty edge set so that each vertex is its own connected component.*
(2) *Repeat until all components are of size at least k:*
   (a) *Pick* any *component T having size smaller than k.*
   (b) *Let u be a vertex in T without any outgoing edge. Since there are at most $k-2$ other vertices in T, one of the $k-1$ nearest neighbors of u, say v, must lie outside T. Add a directed edge (u, v) to the forest whose cost is the Euclidean distance $d(u, v)$ between u and v.*

Observe that the resulting forest consists of trees with size at least k. Also, (u, v) is an edge of the forest only if v is one of the $k-1$ nearest neighbors of u.

The procedure to break trees of size $s > \max(2k-1, 3k-5)$ follows:

Procedure 2 (*Decompose-Component*).
While *components of size $s > \max(2k-1, 3k-5)$ exist* **do**:

(1) *Select one of those components.*
(2) *Pick* any *vertex of the component as the candidate vertex u.*
(3) *Root the tree at the candidate vertex u. Let U be the set of subtrees rooted at the children of u. Let ϕ be the size of the largest subtree of U, rooted at vertex v.*
(4) *If $s - \phi \geq k-1$ then*
   (a) *If $\phi \geq k$ and $s - \phi \geq k$ then partition the component into the largest subtree and the rest (by removing the edge between u and v). Clearly, the size of both resulting components is at least k.*
   (b) *If $s - \phi = k-1$, partition the tree into a component containing the subtrees rooted at the children of v and another component containing the rest. To connect the children of v create a dummy vertex v' to replace v. Note that v' is only a Steiner vertex and does not contribute to the size of the first component. Clearly, the sizes of both components are at least k.*
   (c) *If $\phi = k-1$, then partition into a component containing the subtree rooted at v along with the vertex u and another component containing the rest. In order to connect the children of u in the second component, create a Steiner vertex u'.*
   (d) *Otherwise all subtrees have size at most $k-2$. In this case, create an empty component and keep adding subtrees of u to it until the first time its size becomes at least $k-1$. Clearly, at this point, its size is at most $2k-4$. Put the remaining subtrees into a second component; this second component contains at least $k-1$ vertices, since there are at least $s = \max(2k, 3k-4)$ vertices in all (for $k \leq 4$, we have $s = 2k$ and a second component with at least 3 vertices; for $k > 4$, we have $s = 3k-4$ and a second component with at least $k-1$ vertices). Now, since $s \geq 2k$, at most one of the two components has size equal to $k-1$. If such a component exists, add u to that component, else add u to the first component. In order to keep the partition not containing u connected, a Steiner vertex u' corresponding to u is placed in it.*
(5) *Otherwise pick the root v of the largest subtree as the new candidate vertex and go to Step* (3).

End while

In [43], procedure DECOMPOSE-COMPONENT is shown to terminate. Using both procedures in this section, a forest is created which consists of trees of sizes between k and $\max(2k-1, 3k-5)$. The property that an edge (u, v) exists only if v is one of the $k-1$ nearest neighbors of u still holds.

Table 4
Running example: 3-anonymous version of the SME dataset after microaggregating quasi-identifier attributes with Algorithm μ-Approx

Surface (m^2)	No. employees	Turnover (Euros)	Net profit (Euros)
753.3	50	3 212 334	313 250
753.3	50	2 283 340	299 876
830	17	1 989 233	200 213
830	17	984 983	143 211
830	17	194 232	51 233
360	27	119 332	20 333
360	27	3 012 444	501 233
360	27	4 233 312	777 882
360	27	159 999	60 388
753.3	50	5 333 442	1 001 233
360	27	645 223	333 010

3. An approximation algorithm for optimal multivariate microaggregation

Given a multivariate numerical microdata set V, consider each microdata record as a vertex in a graph $\mathbf{G}_V = (V, \emptyset)$ with no edges and use the following algorithm for microaggregation:

Algorithm (μ-*Approx*($\mathbf{G}_V, k$)).

(1) Call procedure FOREST
(2) Call procedure DECOMPOSE-COMPONENT. Taking the resulting trees as groups yields a k-partition P where groups have sizes between k and $\max(2k-1, 3k-5)$.
(3) In a candidate optimal k-partition, no group size should be larger than $2k-1$ [25]. If $3k-5 > 2k-1$ (that is for $k \geq 5$), Step 2 can yield groups with more than $2k-1$ records, which are partitioned as follows:
   (a) Compute the group centroid.
   (b) Take the record u that is farthest from the centroid.
   (c) Form a first group with u and its $k-1$ nearest neighbors.
   (d) Take the rest of records (at least k and at most $2k-5$) as the second group.
   This splitting step yields a k-partition P' where groups have sizes between k and $2k-1$ (those groups in P with size $\leq 2k-1$ are kept unchanged in P').
(4) Microaggregate using the groups in P'.

Note 2. Any bipartition procedure yielding two groups of size at least k can actually be used in Step 3. Whatever the bipartition, the resulting k-partition will always have a lower SSE than the k-partition output by Step 2 [25]. The bipartition proposed above is the one used in the MDAV heuristic (see [27,38] and Section 6 below), which has proven to perform especially well.

Depending on whether randomness is desired, component and vertex selection within FOREST (Step 2a) and DECOMPOSE-COMPONENT (Steps 1 and 2) can be random or follow some prescribed rule. The randomized version has been used in Section 6.

The variability loss incurred when P' is used for microaggregation is no greater than when using P. The reason is that, if a group in P is split into two groups in P', members of the original group will be replaced by two centroids rather than by a single one, which implies that variability loss either decreases or stays the same.

Even though P' is at least as good as P in terms of variability loss, it no longer has the nice property that its groups are trees in the sense of [43]. This property is necessary to prove the bounds in Sections 4 and 5, so we use the (pessimistic) k-partition P in those sections. However, P' is used to obtain the empirical results reported in Section 6.

Example 3. Algorithm μ-Approx was used to obtain a 3-anonymous version of the dataset in Table 1. Like in Example 2, the company name was suppressed and microaggregation was carried out on the standardized versions of the quasi-identifier attributes "Surface" and "No. employees". Fig. 1 depicts the 11 records projected on those two attributes and grouped after the first two steps of Algorithm μ-Approx, that is, after procedure FOREST and

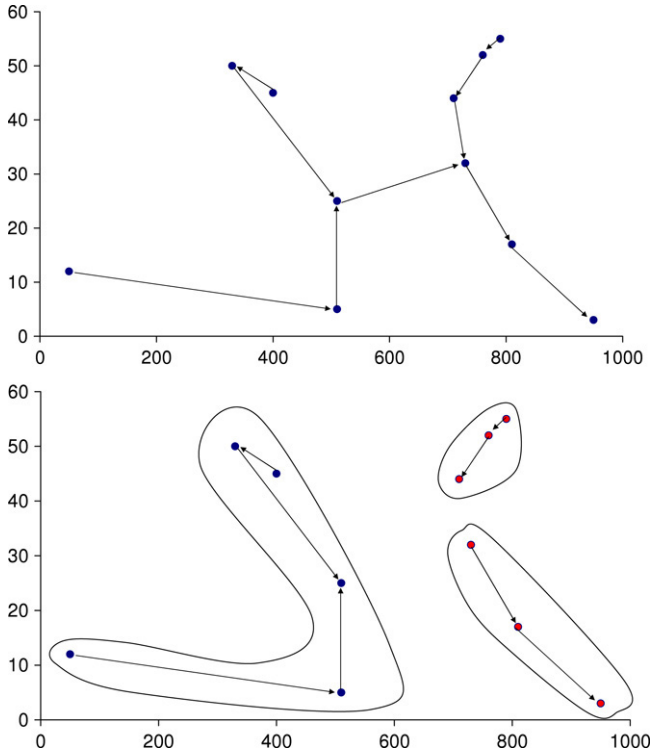

Fig. 1. Top, tree obtained with FOREST on the SME dataset. Bottom, 3-partition P obtained after DECOMPOSE-COMPONENT. In this example P coincides with the 3-partition P' output by the overall Algorithm μ-Approx. Abscissae: surface; ordinates: number of employees.

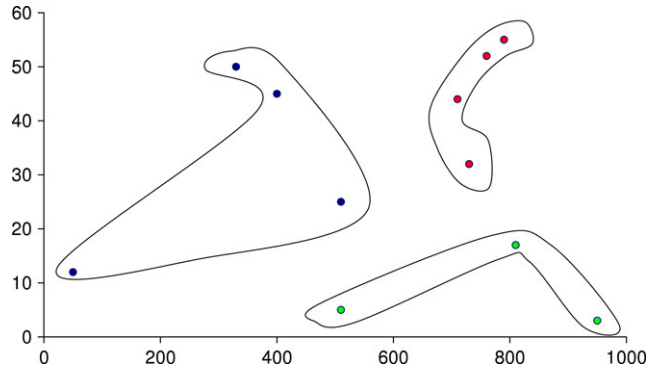

Fig. 2. Optimal k-partition of the SME dataset (Example 2). Abscissae: surface; ordinates: number of employees.

after procedure DECOMPOSE-COMPONENT. The latter grouping corresponds to k-partition P, which in this example coincides with the k-partition P' output by the overall Algorithm μ-Approx. Fig. 2 depicts the optimal k-partition obtained in Example 2.

For k-partition P', the $[0, 1]$-bounded variability loss for the standardized versions of the two quasi-identifier attributes is $\mathrm{SSE}_{P'}/\mathrm{SST} = 8.682/22 = 0.394$, quite close to $\mathrm{SSE}_{\mathrm{opt}}/\mathrm{SST} = 0.34$ obtained for the optimal k-partition in Example 2.

Table 4 gives the 3-anonymous version of the SME dataset output by Algorithm μ-Approx based on k-partition P'. In this case, the snooper looking for the turnover and net profit of a company K&K Sarl with crammed, small premises would hesitate between the 6th, the 7th, the 8th, the 9th or the 11-th records of Table 4. □

3.1. Computational complexity of the μ-Approx algorithm

The μ-Approx algorithm consists of four distinct steps, which are performed in sequence and only once.

The first step is a call to the FOREST procedure. This procedure adds edges between pairs of vertices to grow cycle-free components of size at least k vertices. Basically, FOREST is a loop where each iteration adds a new edge linking two vertices. Since the grown components must be cycle-free, for a graph containing n vertices, no component can contain more than $n-1$ edges; thus, the loop will iterate at most $n-1$ times. At each iteration, a vertex u is selected and linked to one of its $k-1$ nearest neighbor vertices. Such a linkage requires computing $O(n)$ distances to find the $k-1$ neighbors of u among the $n-1$ vertices other than u. Therefore the cost of FOREST is $O(n^2)$.

The second step of μ-Approx is a call to the DECOMPOSE-COMPONENT procedure. This procedure iterates until no components of size greater than $\max(2k-1, 3k-5)$ are left. Each iteration divides a component of size greater than $\max(2k-1, 3k-5)$ into two smaller components of size at least k. In the worst case, DECOMPOSE-COMPONENT starts from a forest consisting of a single tree including all n vertices. Since the resulting components must have size at most $\max(2k-1, 3k-5)$, the maximum number of iterations to be made is $O(n/k)$. Let us analyze the computation in each iteration:

- Step 1 within procedure DECOMPOSE-COMPONENT selects a component of size greater than $\max(2k-1, 3k-5)$, which requires counting its vertices. This is done with a cost $O(n)$, because one might need to count all vertices to find such a component.
- Step 2 within DECOMPOSE-COMPONENT picks a vertex u, which has constant cost $O(1)$.
- Step 3 checks the suitability of u. For this, computing the size ϕ of the largest subtree rooted at u is required, which can be done with cost $O(n)$.
  If $s-\phi < k-1$, Step 5 selects a new u (with cost $O(1)$) and jumps back to Step 3. By construction, a suitable u will be found in at most $k-2$ steps. Therefore, selecting u can be done in $O(kn)$ cost.
- After u is selected, Step 4 partitions the largest subtree rooted at u. If done in Steps 4a, 4b or 4c, this has a cost $O(1)$; if done in Step 4d, this has a cost $O(k)$.

In summary, DECOMPOSE-COMPONENT consists of $O(n/k)$ iterations whose cost is $O(kn)$, so that the overall cost of the procedure is $O(n^2)$.

For $k \leq 4$, the second step yields groups of size at most $2k-1$, so the third step can be skipped. For $k > 4$, the third step is used to split groups of size between $2k$ and $3k-5$ into two groups of size less than $2k$. To do so, the MDAV algorithm is applied, which has a known quadratic cost of $O((3k-5)^2)$ [27]. Since there can be $O(n)$ components to be split, the total cost is $O(k^2 n)$.

The fourth step consists of microaggregating using the groups in the computed k-partition. For each group, the centroid is computed and used to replace the records in the group. For the entire set of groups, this takes $O(n)$ cost.

Since $k \ll n$, the overall complexity of μ-Approx is dominated by $O(n^2)$, that is, quadratic in the number of vertices n.

4. Approximation error vs minimum within-groups Euclidean distance microaggregation

Minimum within-groups Euclidean distance microaggregation is defined as a form of microaggregation where the optimality criterion is to minimize the sum of Euclidean distances from records to the group centroid. Although this type of microaggregation is less frequent than minimum within-groups sum of squares microaggregation addressed in Section 5 below, its approximation is simpler and useful to introduce the approximation for the latter, more common microaggregation.

Let P be the k-partition resulting from Step 2 of Algorithm μ-Approx. We will relate the cost of k-partition P (sum of within-groups Euclidean distances to centroids) and the cost of the optimal Euclidean k-partition P_{optE} (that is, the cost of optimal Euclidean microaggregation).

Let $P = \{T_1, T_2, \ldots, T_r\}$ be the trees or groups in P. Let each tree be $T_i = (V(T_i), E(T_i))$, where $V(T_i)$ is the set of vertices of T_i and $E(T_i)$ is the set of edges of T_i. Let $W(T_i) = \sum_{e \in E(T_i)} w(e)$ be the cost of tree T_i computed as the sum of the costs of its edges (we have defined above the cost of an edge between vertices u and v as the Euclidean distance $d(u, v)$ between those vertices). Let $c(T_i)$ be the centroid of vertices in $V(T_i)$ and $\mu_E(T_i)$ the cost of microaggregating those vertices, that is, the sum of within-group Euclidean distances from vertices to $c(T_i)$, that is, $\mu_E(T_i) = \sum_{u \in T_i} d(u, c(T_i))$.

We first give two lemmata and then a theorem with the desired bound. The lemmata are proven in the Appendix and the theorem follows from the lemmata.

Lemma 1. *The cost* $\mu_E(P) = \sum_{T_i \in P} \mu_E(T_i)$ *of microaggregating* V *using* k*-partition* P *can be bounded as*

$$\mu_E(P) \leq \max(2k-1, 3k-5) W(P) \tag{1}$$

where $W(P)$ *is the cost of all trees in* P.

Let us now compare $W(P)$ with the cost $\mu_E(P_{\text{optE}})$ of optimal microaggregation. The following result holds:

Lemma 2. *The cost* $W(P)$ *of all trees in* P *can be bounded as*

$$W(P) \leq (2k-1)\mu_E(P_{\text{optE}}) \tag{2}$$

where $\mu_E(P_{\text{optE}})$ *is the cost of microaggregating* V *using the optimal* k*-partition* P_{optE}.

The combination of Bounds (1) and (2) yields the following approximation bound.

Theorem 1 (*Bound for Euclidean Microaggregation*). *The cost* $\mu_E(P)$ *(sum of within-group Euclidean distances to the centroid) of microaggregation using a* k*-partition* P *obtained in Step 2 of Algorithm* μ*-Approx can be bounded as*

$$\mu_E(P) \leq (2k-1)\max(2k-1, 3k-5)\mu_E(P_{\text{optE}}) \tag{3}$$

where $\mu_E(P_{\text{optE}})$ *is the cost of minimum Euclidean distance microaggregation (using the optimal* k*-partition* P_{optE}*).*

It should be noted that the bound in Theorem 1 is worst-case. As shown in Section 6, μ-Approx typically achieves a closeness to optimality similar to the best microaggregation heuristics in the literature. The novelty of μ-Approx is that a worst-case bound can be found for it.

5. Approximation error vs minimum within-groups sums of squares microaggregation

Minimum within-groups sum of squares microaggregation is the original form of microaggregation, defined in [36,25]. Here the optimality criterion is to minimize the within-groups sums of squares SSE.

Let P be the k-partition resulting from Step 2 of Algorithm μ-Approx. We use the same notation as in Section 4. If T_i is a tree or group in P, its cost is now $\mu_{\text{SSE}}(T_i) = \sum_{u \in T_i} (d(u, c(T_i)))^2$. We first give two lemmata and then a theorem with the desired bound. The lemmata are proven in the Appendix and the theorem follows from the lemmata.

Lemma 3. *Given* $P = \{T_1, \ldots, T_r\}$, *the cost* $\mu_{\text{SSE}}(P) = \sum_{T_i \in P} \mu_{\text{SSE}}(T_i)$ *of microaggregating* V *using* k*-partition* P *can be bounded as*

$$\mu_{\text{SSE}}(P) \leq \max(2k-1, 3k-5) \sum_{T_i \in P} (W(T_i))^2 \tag{4}$$

where $W(T_i)$ *is the cost of* T_i.

Lemma 4. *If* $P = \{T_1, \ldots, T_r\}$ *is the* k*-partition resulting from Step* 2 *of Algorithm* μ*-Approx, it holds that*

$$\sum_{T_i \in P} (W(T_i))^2 \leq 2(2k-1)\max(2k-1, 3k-5)\mu_{\text{SSE}}(P_{\text{optSSE}}) \tag{5}$$

where $\mu_{\text{SSE}}(P_{\text{optSSE}})$ *is the cost of microaggregating* V *using the optimal* k*-partition* P_{optSSE} *minimizing SSE.*

The combination of Bounds (4) and (5) yields the following approximation bound.

Theorem 2 (*Bound for Minimum SSE Microaggregation*). *The cost* $\mu_{\text{SSE}}(P)$ *(sum of within-group squared Euclidean distances to the centroid) of microaggregation using a* k*-partition* P *obtained in Step* 2 *of Algorithm* μ*-Approx can be bounded as*

$$\mu_{\text{SSE}}(P) \leq 2(2k-1)[\max(2k-1, 3k-5)]^2 \mu_{\text{SSE}}(P_{\text{optSSE}}) \tag{6}$$

where $\mu_{\text{SSE}}(P_{\text{optSSE}})$ *is the cost of minimum SSE microaggregation (using the optimal* k*-partition* P_{optSSE}*).*

Like Theorem 1, Theorem 2 gives a worst-case bound. Empirical results of μ-Approx are comparable to the best in the literature, as shown in the next section.

Table 5
Information loss measures for several datasets under various microaggregation heuristics

Dataset	Measure	Method	$k = 3$	$k = 4$	$k = 5$	$k = 10$
"Tarragona"	L_E	MDAV	34.32	38.66	41.20	49.64
		μ-Approx	34.42	38.82	42.23	54.09
	L_{SSE}	MDAV	16.93	19.55	22.46	33.19
		M-d	16.63	19.66	24.50	38.58
		μ-Approx	17.10	20.51	26.04	38.80
"Census"	L_E	MDAV	22.97	26.53	29.22	36.62
		μ-Approx	25.35	27.09	29.96	38.11
	L_{SSE}	MDAV	5.69	7.51	9.09	14.22
		M-d	6.11	8.24	10.30	17.17
		μ-Approx	6.25	8.47	10.78	17.01
"EIA"	L_E	MDAV	4.56	5.60	8.13	12.87
		μ-Approx	4.30	5.18	6.16	10.06
	L_{SSE}	MDAV	0.48	0.67	1.67	3.84
		μ-Approx	0.43	0.59	0.83	2.26

6. Empirical results

In addition to its being the only known approximation to optimal multivariate microaggregation, Algorithm μ-Approx is as good as the best heuristics in the literature for that problem. To illustrate this, we offer in this section an empirical comparison.

We have used three reference datasets [44] used in the European project CASC:

(1) The "Tarragona" dataset contains 834 records with 13 numerical attributes. This dataset was used in CASC and in [25,39].
(2) The "Census" dataset contains 1080 records with 13 numerical attributes. This dataset was used in CASC and in [45–47,39,27].
(3) The "EIA" dataset contains 4092 records with 11 numerical attributes (plus two additional categorical attributes not used here). This dataset was used in CASC, in [46] and partially in [39] (an undocumented subset of 1080 records from "EIA", called "Creta" dataset, was used in the latter paper).

Table 5 gives the information loss under various methods for the three above datasets and different values of k. For each dataset, multivariate microaggregation embraces all attributes, that is, p equal to the number of attributes is used. Two information loss measures taking values in the interval [0, 100] are considered:

- $L_E = 100 \times \text{SDE/SDT}$, where SDE is the sum of within-groups Euclidean distances to centroids (called μ_E in Theorem 1) and SDT is the total sum of Euclidean distances from all records to the dataset centroid.
- $L_{\text{SSE}} = 100 \times \text{SSE/SST}$, where SSE is the sum of within-groups squared Euclidean distances to the centroids (called μ_{SSE} in Theorem 2) and SST is the total sum of squares (sum of squared Euclidean distances from all records to the dataset centroid).

Algorithm μ-Approx is compared with the following two methods:

(1) An improved version of the heuristic in [25] called MDAV (Maximum Distance to Average Vector, [27]). MDAV is the microaggregation method implemented in the μ-Argus package [38] resulting from the CASC project. It computes a k-partition with groups of fixed size k by iterating the bipartition described in the third step of Algorithm μ-Approx: the dataset centroid is computed and a group with k records is created consisting of the record farthest from the centroid and its $k - 1$ nearest neighbors; if there are less than $2k$ records left, they are taken as the last group of the k-partition and MDAV is finished; otherwise the remaining records are fed to the next iteration.
(2) The minimum spanning tree-based heuristic M-d of [39], which is the best performer in that paper. This algorithm constructs the minimum spanning tree (MST) over the dataset records; then edges from the MST are removed to get a forest where no tree has less than k records; finally, trees in the forest that contain $2k$ or more records are

Table 6
Optimal SSE vs SSE obtained with μ-Approx and MDAV for several values of k

k	Optimal SSE (1)	SSE μ-Approx (2)	SSE MDAV (3)	((2) − (1))/(1)	((3) − (1))/(1)
2	59.52	81.46	68.48	0.3687	0.1506
3	95.64	120.39	112.92	0.2587	0.1806
4	113.28	145.01	125.77	0.2802	0.1103
5	130.24	157.65	145.54	0.2105	0.1175
6	153.17	163.28	166.06	0.0659	0.0841

Average for five simulated datasets with $n = 16$ and $p = 13$, where attribute values have been drawn from a $N(0, 1)$ distribution.

partitioned using the MDAV-like microaggregation heuristic from [25]. Thus, groups of size between k and $2k - 1$ can appear at the forest stage or as a result of the MDAV-like partition stage (the last group in the partition can be of size greater than k). Thus, M-d provides variable-size groups. Comparable results in [39] are only available for L_{SSE} with the "Tarragona" and the "Census" datasets.

It can be observed from Table 5 that, for the smaller datasets "Tarragona" and "Census", MDAV is the best performer (yielding lowest information loss), followed by M-d and μ-Approx. However, the differences between the three algorithms are quite small; in fact, for $k = 10$ and the "Tarragona" dataset, μ-Approx performs better than M-d.

For the "EIA" dataset, μ-Approx clearly outperforms MDAV. This dataset is larger and more skewed than the previous two: records in "EIA" are more clustered (the skewness of attributes is higher) than for "Tarragona" and "Census", which might explain why variable-sized groups formed by μ-Approx perform better. The information losses given in [39] for M-d on the "Creta" subset of "EIA" used there look poorer than those of the D method – equivalent to MDAV – on that subset, so M-d is unlikely to beat μ-Approx for the full "EIA" dataset.

As mentioned in Section 3.1 above, the complexity of μ-Approx is $O(n^2)$. Using a Pentium4 processor running under Debian GNU/Linux at 3.06 GHz with a 2.256 GB RAM, the 834 records in the "Tarragona" dataset were microaggregated in about 1 s, the 1080 records in "Census" took about 1 s and the 4092 records in "EIA" took 25 s. For large datasets (with more than 10 000 records) it may be wise to use blocking attributes to split the dataset into several smaller datasets and microaggregate these separately. This blocking strategy is necessary even with the fastest heuristics, such as MDAV, because their time and storage complexity is no less than $O(n^2)$.

Finally, we give some experimental results on how close to optimality are the partitions obtained using MDAV and μ-Approx. In order to be able to find the optimal k-partition by exhaustive search, we are constrained to use very small datasets. We have simulated two kinds of datasets:

Non-clustered datasets. Five simulated datasets with $n = 16$ records and $p = 13$ attributes were generated, where attribute values were drawn from a $N(0, 1)$ distribution. For several values of k, Table 6 gives the average optimal SSE, the average SSE of the k-partitions found by μ-Approx and the average SSE of the k-partitions found by MDAV. Averages are over the five datasets. The two rightmost columns of the table show the relative optimality gaps of μ-Approx and MDAV.

Clustered datasets. Two datasets were generated:

- A simulated dataset with $n = 16$ records and $p = 13$ attributes, with five clusters of records of sizes 2, 2, 3, 4 and 5, where attribute values of records in each cluster were drawn from distributions $N(0, 1)$, $N(3, 1)$, $N(6, 1)$, $N(9, 1)$ and $N(12, 1)$, respectively. For several values of k, Table 7 gives the optimal SSE, the SSE of the k-partition found by μ-Approx and the SSE of the k-partition found by MDAV. The two rightmost columns of the table show the relative optimality gaps of μ-Approx and MDAV.
- A simulated dataset with $n = 16$ records and $p = 13$ attributes, with four clusters of records of sizes 3, 4, 4 and 5, where attribute values of records in each cluster were drawn from distributions $N(0, 1)$, $N(4, 1)$, $N(8, 1)$ and $N(12, 1)$, respectively. Table 8 is analogous to Table 7 for this dataset.

In all trials, microaggregation embraced all 13 attributes. For the non-clustered datasets, it can be seen that the SSE obtained with MDAV is closer to the optimum SSE than the SSE given by μ-Approx, except for $k = 6$. For the clustered datasets, the situation is inverted: for most values of k, μ-Approx yields a smaller SSE than MDAV; moreover, in three cases, μ-Approx yields the optimum k-partition.

Table 7
Optimal SSE vs SSE obtained with μ-Approx and MDAV for several values of k and a simulated dataset with $n = 16$ and $p = 13$, with five clusters of sizes 2, 2, 3, 4 and 5

k	Optimal SSE (1)	SSE μ-Approx (2)	SSE MDAV (3)	$((2) - (1))/(1)$	$((3) - (1))/(1)$
2	109.68	122.96	197.88	0.1211	0.8041
3	298.16	325.73	399.44	0.0925	0.3397
4	443.34	554.76	458.62	0.2513	0.0345
5	698.67	1064.18	698.67	0.5231	0.0000
6	1064.18	1064.18	1154.03	0.0000	0.0844

Attribute values of records in each cluster have been drawn from distributions $N(0, 1)$, $N(3, 1)$, $N(6, 1)$, $N(9, 1)$ and $N(12, 1)$, respectively.

Table 8
Optimal SSE vs SSE obtained with μ-Approx and MDAV for several values of k and a simulated dataset with $n = 16$ and $p = 13$, with four clusters of sizes 3, 4, 4 and 5

k	Optimal SSE (1)	SSE μ-Approx (2)	SSE MDAV (3)	$((2) - (1))/(1)$	$((3) - (1))/(1)$
2	119.06	132.51	335.84	0.1130	1.8208
3	179.97	179.97	405.69	0.0000	1.2542
4	537.08	544.38	559.72	0.0136	0.0421
5	628.73	964.24	680.27	0.5336	0.0820
6	964.24	964.24	1309.23	0.0000	0.3578

Attribute values of records in each cluster have been drawn from distributions $N(0, 1)$, $N(4, 1)$, $N(8, 1)$ and $N(12, 1)$, respectively.

Overall, μ-Approx and MDAV get comparably close to the optimum, with μ-Approx being especially good in the case of clustered or skewed datasets. Further, the SSE obtained with μ-Approx is much closer to the optimal SSE than the bound given by Theorem 2; this is because the bound is worst-case and referred to the k-partition P obtained in Step 2 of μ-Approx, while empirical results refer to the (better) k-partition P' obtained in Step 3 of μ-Approx.

7. Practical implications and conclusions

Microaggregating with minimum information loss has been known to be an important and difficult issue ever since microaggregation was invented as a masking method for statistical disclosure control. However, optimality in SDC is not just about minimum information loss but about the best tradeoff between low information loss and low disclosure risk. The recent application of microaggregation to k-anonymity has strengthened the importance of minimum information loss: once a suitable k is selected to keep the re-identification risk low enough, the only job left is to k-anonymize (that is, to microaggregate) with as little information loss as possible.

We have proposed a heuristic for multivariate microaggregation for which approximation bounds can be derived for any value of k. To the best of our knowledge, ours is the first general approximation in the literature. Bounds have been proven for two different optimality criteria: an $O(k^2)$-approximation bound for minimum within-groups sums of Euclidean distances, and an $O(k^3)$-approximation bound for minimum within-groups sums of squares. In addition to having nice theoretical properties that allow bounding the approximation error, the proposed heuristic has been shown to be comparable in performance to the best multivariate microaggregation heuristics in the literature.

The approximations presented in this paper are general because they are valid for any k; however, they may be too loose for certain purposes. A possible direction for future work is to devise equally efficient heuristics which result in $O(1)$ or $O(k)$-approximations. A way to obtain tighter approximations might be to sacrifice generality, *i.e.* to explore approximations only valid for some values of k (*e.g.* specific constant k or k power of a certain number); we have achieved this for $k = 2$ in [48], but not yet for more frequently used values, like $k = 3$. Finally, finding approximations for categorical microaggregation is also an open issue.

Disclaimer and acknowledgments

The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization. The presentation of this paper has greatly benefited from the

recommendations of two anonymous reviewers. This work was partly supported by the Government of Catalonia under grant 2005 SGR 00446, by the Spanish Ministry of Education and Science under projects SEG2004-04532-C04-01 "PROPRIETAS" and CONSOLIDER CSD2007-00004 "ARES", and by Cornell University under contract no. 47632-10043.

Appendix. Proofs

The structure of this appendix is as follows. First Lemma 5 is given for use in the subsequent proof of the above Lemma 1. The proof of the above Lemma 2 comes next and needs no auxiliary results. Then the proof of the above Lemma 3 is given, which also uses Lemma 5. After that, Lemma 6 is given that is used to prove the subsequent Lemma 7; finally, both Lemmas 6 and 7 are used in the subsequent proof of the above Lemma 4.

In the proofs below, neighbor$(u, k, \mathbf{G})$ denotes a function returning the k-th nearest neighbor of vertex u within graph $\mathbf{G}$.

Lemma 5. *If u is a vertex in a tree T and $c(T)$ is the tree centroid, then $d(u, c(T)) \leq W(T)$, where $W(T)$ is the sum of lengths of the edges in T.*

Proof. By the central position of the centroid, for any vertex u in T there is another vertex v more distant from u than the centroid. Further $d(u, v) \leq W(T)$, because taking a straight path from u to v is no longer than going from u to v along the tree edges. Thus, $d(u, c(T)) \leq d(u, v) \leq W(T)$. □

Proof (*Lemma* 1). Consider a tree T_i with L_i vertices. By Lemma 5, we have

$$\mu_E(T_i) \leq L_i W(T_i). \tag{7}$$

Taking $L_i = \max(2k-1, 3k-5)$ (maximal group size in P) and adding for all trees $T_i \in P$, we obtain Bound (1). □

Proof (*Lemma* 2). For each group G_i in the optimal k-partition P_{optE} and for each $u \in G_i$ it holds that

$$d(u,\ \text{neighbor}(u, L_i - 1, \mathbf{G}_V)) \leq d(u,\ \text{neighbor}(u, L_i - 1, G_i)) \leq \mu_E(G_i)$$

where L_i is the size of G_i, that is, $k \leq L_i \leq 2k-1$. Adding for all $u \in G_i$, we get

$$\sum_{u \in G_i} d(u,\ \text{neighbor}(u, L_i - 1, \mathbf{G}_V)) \leq L_i \cdot \mu_E(G_i).$$

Adding for all groups $G_i \in P_{\text{optE}}$ and taking $L_i = 2k-1$ (maximal group size in P_{optE}), we get the following inequality

$$\sum_{u \in V} d(u,\ \text{neighbor}(u, 2k-2, \mathbf{G}_V)) \leq (2k-1)\mu_E(P_{\text{optE}}). \tag{8}$$

On the other hand, the distance to the $k-1$-th nearest neighbor is less than the distance to the $2k-2$-th nearest neighbor, so

$$\sum_{u \in V} d(u,\ \text{neighbor}(u, k-1, \mathbf{G}_V)) \leq \sum_{u \in V} d(u,\ \text{neighbor}(u, 2k-2, \mathbf{G}_V)). \tag{9}$$

Next, we note that the contribution of a vertex u to its tree in P is at most the distance to its $k-1$-th nearest neighbor in $\mathbf{G}_V$, and thus the total cost $W(P)$ can be bounded as

$$W(P) \leq \sum_{u \in V} d(u,\ \text{neighbor}(u, k-1, \mathbf{G}_V)). \tag{10}$$

Finally, by chaining Inequalities (8)–(10), we get Bound (2). □

Proof (*Lemma* 3). Consider a tree T_i with L_i vertices. For any $u \in V(T_i)$, we have by Lemma 5

$$(d(u, c(T_i)))^2 \leq (W(T_i))^2 \tag{11}$$

where $d(\cdot,\cdot)$ is the Euclidean distance. Adding up Inequality (11) for all vertices in T_i we get

$$\mu_{\mathrm{SSE}}(T_i) \leq L_i(W(T_i))^2. \tag{12}$$

Taking $L_i = \max(2k-1, 3k-5)$ (maximal group size in P) and adding for all trees $T_i \in P$, we obtain Bound (4). □

Lemma 6. *Given real numbers $d_1, d_2, \ldots, d_L$ it holds that*

$$\sum_{i=1}^{L} d_i^2 \geq \frac{\left(\sum_{i=1}^{L} d_i\right)^2}{L}.$$

Equality holds when $d_1 = d_2 = \cdots = d_L$.

Proof. The lemma is just a special case of Jensen's inequality. □

Lemma 7. *For any two vertices u, v in a graph G, it holds that*

$$(d(u,v))^2 \leq 2\mu_{\mathrm{SSE}}(G)$$

where $\mu_{\mathrm{SSE}}(G)$ is the SSE of G.

Proof. If $u, v \in G$ and $c(G)$ is the centroid of G, clearly the SSE of G is greater than or equal to the contributions of u and v to that SSE. Formally,

$$\mu_{\mathrm{SSE}}(G) \geq (d(u,c(G)))^2 + (d(v,c(G)))^2. \tag{13}$$

We now can use Lemma 6 and the triangular inequality to get

$$(d(u,c(G)))^2 + (d(v,c(G)))^2 \geq \frac{(d(u,c(G)) + d(v,c(G)))^2}{2} \geq \frac{(d(u,v))^2}{2}. \tag{14}$$

Combining Inequalities (13) and (14), we have

$$(d(u,v))^2 \leq 2\mu_{\mathrm{SSE}}(G). \quad \square \tag{15}$$

Proof (*Lemma* 4). For each group G_i in the optimal k-partition, we can use Lemma 7 to write

$$\sum_{u \in G_i} (d(u,\ \mathrm{neighbor}(u, L_i - 1, G_i)))^2 \leq 2L_i\mu_{\mathrm{SSE}}(G_i)$$

where L_i is the size of G_i. Since the $L_i - 1$-th nearest neighbor in $\mathbf{G}_V$ is no farther than the $L_i - 1$-th nearest neighbor in G_i, we can transform the previous inequality into

$$\sum_{u \in G_i} (d(u,\ \mathrm{neighbor}(u, L_i - 1, \mathbf{G}_V)))^2 \leq 2L_i\mu_{\mathrm{SSE}}(G_i).$$

Now adding for all groups G_i in the optimal k-partition P_{optSSE} and taking $L_i = 2k - 1$ (the largest possible group size in P_{optSSE}), we get

$$\sum_{u \in \mathbf{G}_V} (d(u,\ \mathrm{neighbor}(u, 2k-2, \mathbf{G}_V)))^2 \leq 2(2k-1)\mu_{\mathrm{SSE}}(P_{\mathrm{optSSE}}).$$

We now group the vertices of $\mathbf{G}_V$ following the trees in P and rewrite the previous inequality as

$$\sum_{i=1}^{r} \sum_{u \in T_i} (d(u,\ \mathrm{neighbor}(u, 2k-2, \mathbf{G}_V)))^2 \leq 2(2k-1)\mu_{\mathrm{SSE}}(P_{\mathrm{optSSE}}).$$

Since the $k-1$-th nearest neighbor is no farther than the $2k-2$-th nearest neighbor, we transform the previous inequality into

$$\sum_{i=1}^{r}\sum_{u\in T_i}(d(u,\ \text{neighbor}(u,k-1,\mathbf{G}_V)))^2 \leq 2(2k-1)\mu_{\text{SSE}}(P_{\text{optSSE}}).$$

We now use Lemma 6 on the left-hand side of the above inequality and the fact that the size of a tree T_i is at most $\max(2k-1, 3k-5)$ to get

$$\sum_{i=1}^{r}\frac{\left(\sum_{u\in T_i} d(u,\ \text{neighbor}(u,k-1,\mathbf{G}_V))\right)^2}{\max(2k-1,3k-5)} \leq 2(2k-1)\mu_{\text{SSE}}(P_{\text{optSSE}}). \tag{16}$$

Next, we note that the contribution of a vertex u to its tree T_i is at most the distance to its $k-1$-th nearest neighbor in $\mathbf{G}_V$, and thus $W(T_i)$ can be bounded as

$$W(T_i) \leq \sum_{u\in T_i} d(u, \text{neighbor}(u,k-1,\mathbf{G}_V)). \tag{17}$$

By combining Inequalities (16) and (17), we get Bound (5). □

References

[1] CASC, Computational Aspects of Statistical Confidentiality, European Project IST-2000-25069 CASC. http://neon.vb.cbs.nl/casc, 2001–2004.
[2] HIPAA, Health Insurance Portability and Accountability Act. http://www.hhs.gov/ocr/hipaa/, 2004.
[3] C. Boyens, R. Krishnan, R. Padman, On privacy-preserving access to distributed heterogeneous healthcare information, in: Proceedings of the 37th Hawaii International Conference on System Sciences HICSS-2004, IEEE Computer Society, Big Island, HI, 2004.
[4] EuroPrivacy, European privacy regulations. http://europa.eu.int/comm/internal_market/privacy/law_en.htm, 2005.
[5] USPrivacy, US privacy regulations. http://www.media-awareness.ca/english/issues/privacy/us_legislation_privacy.cfm, 2005.
[6] L. Willenborg, T. DeWaal, Elements of Statistical Disclosure Control, Springer-Verlag, New York, 2001.
[7] S. Giessing, Survey on methods for tabular data protection in ARGUS, in: Privacy in Statistical Databases, in: J. Domingo-Ferrer, V. Torra (Eds.), Lecture Notes in Computer Science, vol. 3050, Springer, Berlin, Heidelberg, 2004, pp. 1–13.
[8] J. Schlörer, Disclosure from statistical databases: Quantitative aspects of trackers, ACM Transactions on Database Systems 5 (1980) 467–492.
[9] F.Y. Chin, G. Ozsoyoglu, Auditing and inference control in statistical databases, IEEE Transactions on Software Engineering 8 (1982) 574–582.
[10] G.T. Duncan, S. Mukherjee, Optimal disclosure limitation strategy in statistical databases: Deterring tracker attacks through additive noise, Journal of the American Statistical Association 95 (2000) 720–729.
[11] R. Garfinkel, R. Gopal, D. Rice, New approaches to disclosure limitation while answering queries to a database: Protecting numerical confidential data against insider threat based on data and algorithms, in: Proc. of the 39th Annual Hawaii International Conference on System Sciences-HICSS 2006, IEEE Computer Society, Kauai, HI, 2006.
[12] D. Defays, P. Nanopoulos, Panels of enterprises and confidentiality: The small aggregates method, in: Proc. of 92 Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada, Ottawa, 1993, pp. 195–204.
[13] D. Pagliuca, G. Seri, Some results of the individual ranking method on the system of enterprise accounts annual survey, ESPRIT SDC Project, Deliverable MI-3/D2.11, 1999.
[14] M. Rosemann, Erste Ergebnisse von vergleichenden Untersuchungen mit anonymisierten und nicht anonymisierten Einzeldaten am Beispiel der Kostenstrukturerhebung und der Umsatzsteuerstatistik, in: G. Ronning, R. Gnoss (Eds.), Anonymisierung wirtschaftsstatistischer Einzeldaten, Statistisches Bundesamt, Wiesbaden, 2003, pp. 154–183.
[15] R. Lenz, D. Vorgrimler, Matching German turnover tax statistics, FDZ-Arbeitspapier Nr. 4, Statistische Ämter des Bundes und der Länder-Forschungsdatenzentren, 2005.
[16] 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.
[17] T. Dalenius, Finding a needle in a haystack — or identifying anonymous census records, Journal of Official Statistics 2 (3) (1986) 329–336.
[18] P. Samarati, Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027.
[19] J. Domingo-Ferrer, V. Torra, Disclosure protection methods and information loss for microdata, in: P. Doyle, J.I. Lane, J.J.M. Theeuwes, L. Zayatz (Eds.), Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, North-Holland, Amsterdam, 2001, pp. 91–110.
[20] R. Sarathy, K. Muralidhar, R. Parsa, Perturbing non-normal confidential: The Copula approach, Management Science 48 (12) (2002) 1613–1627.
[21] R. Sarathy, K. Muralidhar, The security of confidential numerical data in databases, Information Systems Research 13 (4) (2002) 389–403.

[22] J. Domingo-Ferrer, V. Torra, A quantitative comparison of disclosure control methods for microdata, in: P. Doyle, J.I. Lane, J.J.M. Theeuwes, L. Zayatz (Eds.), Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, North-Holland, Amsterdam, 2001, pp. 111–134.
[23] R. Moore, Controlled data swapping techniques for masking public use microdata sets, US Bureau of the Census, Washington DC, 1996 (unpublished manuscript).
[24] R. Brand, Microdata protection through noise addition, in: Inference Control in Statistical Databases, in: J. Domingo-Ferrer (Ed.), Lecture Notes in Computer Science, vol. 2316, Springer, Berlin, Heidelberg, 2002, pp. 97–116.
[25] J. Domingo-Ferrer, J.M. Mateo-Sanz, Practical data-oriented microaggregation for statistical disclosure control, IEEE Transactions on Knowledge and Data Engineering 14 (1) (2002) 189–201.
[26] V. Torra, Microaggregation for categorical variables: A median based approach, in: Privacy in Statistical Databases, in: J. Domingo-Ferrer, V. Torra (Eds.), Lecture Notes in Computer Science, vol. 3050, Springer, Berlin, Heidelberg, 2004, pp. 162–174.
[27] J. Domingo-Ferrer, V. Torra, Ordinal, continuous and heterogenerous k-anonymity through microaggregation, Data Mining and Knowledge Discovery 11 (2) (2005) 195–212.
[28] J.H. Ward, Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association 58 (1963) 236–244.
[29] A.W.F. Edwards, L.L. Cavalli-Sforza, A method for cluster analysis, Biometrics 21 (1965) 362–375.
[30] R.C. Jancey, Multidimensional group analysis, Australian Journal of Botany 14 (1966) 127–130.
[31] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–297.
[32] A.D. Gordon, J.T. Henderson, An algorithm for Euclidean sum of squares classification, Biometrics 33 (1977) 355–362.
[33] P. Hansen, B. Jaumard, N. Mladenovic, Minimum sum of squares clustering in a low dimensional space, Journal of Classification 15 (1998) 37–55.
[34] A. Oganian, J. Domingo-Ferrer, On the complexity of optimal microaggregation for statistical disclosure control, Statistical Journal of the United Nations Economic Comission for Europe 18 (4) (2001) 345–354.
[35] S.L. Hansen, S. Mukherjee, A polynomial algorithm for optimal univariate microaggregation, IEEE Transactions on Knowledge and Data Engineering 15 (4) (2003) 1043–1044.
[36] D. Defays, N. Anwar, Micro-aggregation: A generic method, in: Proceedings of the 2nd International Symposium on Statistical Confidentiality, Eurostat, Luxemburg, 1995, pp. 69–78.
[37] G. Sande, Exact and approximate methods for data directed microaggregation in one or more dimensions, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (5) (2002) 459–476.
[38] A. Hundepool, A. Van de Wetering, R. Ramaswamy, L. Franconi, A. Capobianchi, P.-P. DeWolf, J. Domingo-Ferrer, V. Torra, R. Brand, S. Giessing, μ-ARGUS Version 4.0.2 Software and User's Manual, Statistics Netherlands, Voorburg, NL. http://neon.vb.cbs.nl/casc, 2005.
[39] M. Laszlo, S. Mukherjee, Minimum spanning tree partitioning algorithm for microaggregation, IEEE Transactions on Knowledge and Data Engineering 17 (7) (2005) 902–911.
[40] P. Samarati, L. Sweeney, Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, Tech. Rep., SRI International, 1998.
[41] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression, International Journal of Uncertainty, Fuzziness and Knowledge Based Systems 10 (5) (2002) 571–588.
[42] L. Sweeney, k-anonimity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge Based Systems 10 (5) (2002) 557–570.
[43] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, A. Zhu, Approximation algorithms for k-anonymity, Journal of Privacy Technology, Paper No. 20051120001. http://www.jopt.org/papers.html, 2005.
[44] R. Brand, J. Domingo-Ferrer, J.M. Mateo-Sanz, 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.
[45] J. Domingo-Ferrer, J.M. Mateo-Sanz, V. Torra, Comparing SDC methods for microdata on the basis of information loss and disclosure risk, in: Pre-proceedings of ETK-NTTS'2001, vol. 2, Eurostat, Luxemburg, 2001, pp. 807–826.
[46] R. Dandekar, J. Domingo-Ferrer, F. Sebé, LHS-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection, in: Inference Control in Statistical Databases, in: J. Domingo-Ferrer (Ed.), Lecture Notes in Computer Science, vol. 2316, Springer, Berlin, Heidelberg, 2002, pp. 153–162.
[47] W.E. Yancey, W.E. Winkler, R.H. Creecy, Disclosure risk assessment in perturbative microdata protection, in: J. Domingo-Ferrer (Ed.), Inference Control in Statistical Databases, in: Lecture Notes in Computer Science, vol. 2316, Springer, Berlin, Heidelberg, 2002, pp. 135–152.
[48] J. Domingo-Ferrer, F. Sebé, Optimal multivariate 2-microaggregation for microdata protection: A 2-approximation, in: J. Domingo-Ferrer, L. Franconi (Eds.), Privacy in Statistical Databases-PSD 2006, in: Lecture Notes in Computer Science, vol. 4302, Springer, Berlin, Heidelberg, 2006, pp. 129–138.