A Linear-Time Multivariate Micro-aggregation for Privacy Protection in Uniform Very Large Data Sets

Agusti Solanas and Roberto Di Pietro

UNESCO Chair in Data Privacy
Dept. Computer Engineering and Mathematics
Rovira i Virgili University
Av. Països Catalans, 26. E-43001 Tarragona. Catalonia
{agusti.solanas,roberto.dipietro}@urv.cat

Abstract. Optimally micro-aggregating a multivariate data set is known to be NP-hard, thus, heuristic approaches are used to cope with this privacy preserving problem. Unfortunately, algorithms in the literature are computationally costly, and this prevents using them on large data sets.

We propose a partitioning algorithm to micro-aggregate uniform very large data sets with cost $O(n)$. We provide the mathematical foundations proving the efficiency of our algorithm and we show that the error associated to micro-aggregation is bounded and decreases when the number of micro-aggregated records grows. The experimental results confirm the prediction of the mathematical analysis. In addition, we provide a comparison between our proposal and MDAV, a well-known micro-aggregation algorithm with cost $O(n^2)$.

1 Introduction

Information and communication technologies (ICT) foster the gathering of personal data. The old paper-based files that occupy a large amount of space are being replaced by electronic files that can be stored in tiny USB flash drives. Thus, a paradigm shift is tacking place in many of our daily activities. An exponent of this paradigm shift is the so-called *e-administration* that aims to achieve a paperless office, where all the old paper-based processes are replaced by electronic ones. The main goal of this change is to improve productivity and performance. At the same time, moving aside paper-based offices may lead to total transparency and accountability and, by extension, to better *e-governance*.

This novel way of understanding the management of data is especially reflected in very important areas such as e-commerce and health-care that must conform to strict regulations [1][5]. In addition, most countries have legislation which compels national statistical agencies to guarantee statistical confidentiality [10][11][14]. Thus, protecting individual privacy is a key issue for many institutions, namely statistical agencies, Internet companies, manufacturers, etc.

Many efforts have been devoted to develop techniques guaranteeing privacy, but there are many examples of negligence regardless. British politicians became astonished when they were told on November 20th, 2007, that two computer disks full of

V. Torra and Y. Narukawa (Eds.): MDAI 2008, LNAI 5285, pp. 203–214, 2008.
© Springer-Verlag Berlin Heidelberg 2008

personal data of 25m British individuals had gone missing. The fate of the disks is unknown and the privacy of the individuals, whose personal data are lost, is in danger. This is the most recent of a series of similar unfortunate cases. In October, 2007, Her Majesty's Revenue and Customs (HMRC) lost another disk containing pension records of 15.000 people. Data on 26.5m people were stolen from the home of an employee of the Department of Veterans Affairs in America in 2006, and 658000 queries were disclosed by the AOL search engine in August of the same year. These disclosures of personal information are not new; however, due to the great advances in the Information and Communication Technologies (ICTs), it is very easy to gather large amounts of personal data, and mistakes such as the formerly explained are magnified.

Due to this dramatic increase in the amount of stored personal data, there is a real need for efficient methods to protect privacy. Micro-aggregating data is a common solution to protect the privacy of the users, whose data is stored, however, micro-aggregating very large data sets is a very costly task when current of-the-shelf methods are used.

With the aim of overcoming the limitations of the current micro-aggregation methods, we propose an efficient micro-aggregation algorithm that allows large amounts of electronic data to be micro-aggregated in linear time, *i.e.* with a cost $O(n)$. Our proposal have been designed to work with uniformly distributed data, however, it can be extended to work with other data distributions. Experiments confirm the results of the mathematical analysis and proof the usefulness of our proposal.

The rest of the paper is organised as follows. Section 2 is a summary of the main concepts of micro-aggregation and its methods. In Section 3 our proposal is explained in detail. Section 4 contains the experimental results that proof the usefulness of our method. Finally, the article concludes in Section 5.

2 Background

Personal privacy is a foundational principle stated by the Universal Declaration of Human Rights[1]. With the aim of protecting this foundational right, Statistical Disclosure Control (SDC) was proposed as a discipline that sought to transform data in such a way that they could be publicly released whilst preserving data utility and statistical confidentiality. The point was to avoid disclosure of information that could be linked to specific individual or corporate respondent entities.

It is necessary to keep the balance between the right of the individuals to privacy and the right of the society to knowledge. The solution to this problem is somewhere between two extremes: (i) **No modification**, *i.e.* maximal data utility but no privacy, and (ii) **total encryption**, *i.e.* absolute privacy but no data utility.

One of the youngest techniques proposed to keep this balance is micro-aggregation. It is an SDC technique consisting in the aggregation of individual data. It can be considered as an SDC sub-discipline devoted to the protection of individual data, also called *micro-data*. Micro-aggregation can be understood as a clustering problem with constraints on the size

[1] "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour or reputation ..." ***Universal Declaration of Human Rights.***

of the clusters. It is related to other clustering problems (*e.g.* dimension reduction or minimum squares design of clusters). However, the main difference is that micro-aggregation does not consider the number of clusters to generate or the number of dimensions to reduce, but the minimum number of records that must be grouped in the same cluster.

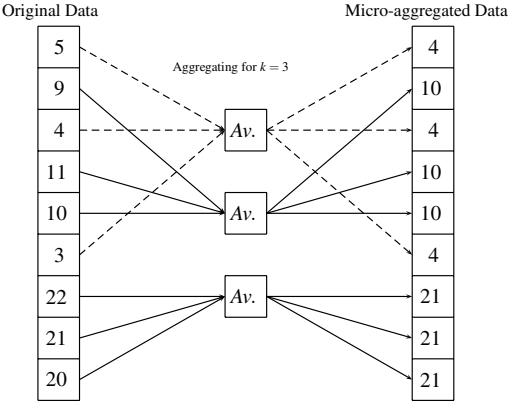


Fig. 1. Example of univariate micro-aggregation where the minimum cardinality of the groups is $k = 3$

When micro-aggregation is used, two main goals have to be kept in mind:

- *Preserving data utility.* Similar records should be micro-aggregated instead of different ones. In the example given in Figure 1, groups of three records are built and aggregated. Note that records in the same group are similar.
- *Protecting respondents' privacy.* Data have to be sufficiently modified to prevent re-identification or disclosure. In the example given in Figure 1, after micro-aggregating the original data, it is impossible to distinguish records from the same group. Thus, the probability of linking a respondent with his/her micro-data is inversely proportional to the number of aggregated records.

In order to determine the information loss produced by micro-aggregation the Sum of Squared Errors (SSE) is used (*cf.* Expression 1).

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

where g is the number of groups, n_i is the number of records in the i-th group, $\mathbf{x_{ij}}$ is the j-th record in the i-th group and $\bar{\mathbf{x}}_\mathbf{i}$ is the average record of the i-th group.

This SSE measure is generally compared with the total error (SST) defined in Equation 2.

$$\mathbf{SST} = \sum_{i=1}^{n} (\mathbf{x_i} - \bar{\mathbf{x}})'(\mathbf{x_i} - \bar{\mathbf{x}}) \tag{2}$$

where n is the number of records in the data set, x_i is a record of the data set and $\bar{x}$ is the average record of the data set.

Given a homogeneity measure such as the SSE and a security parameter k, which determines the minimum cardinality of the groups, the micro-aggregation problem can be enunciated as follows:

> *Given a data set* **D** *with* **n** *records in a characteristic space* $\mathbb{R}^d$*, the problem consists in obtaining a k-partition*[2] $\mathcal{P}$ *of* **D***, so that the SSE of* $\mathcal{P}$ *is minimised. Once* $\mathcal{P}$ *is obtained, each record of every part of* $\mathcal{P}$ *is replaced by the average record of the part.*

The micro-aggregation problem is known to be NP-hard [9] for multivariate data sets, thus, heuristic methods must be used to solve it. There is a plethora of this kind of methods to address the multivariate micro-aggregation problem [15]. Some of them are based on building a tree structure connecting the records in the data set and partition the tree to generate a k-partition. Examples of this kind are the Minimum Spanning Tree Partitioning (MSTP) proposed in [7] and the more recently proposed μ-approximation [3]. The main limitation of the MSTP remains in its foundation, *i.e.* in the minimum spanning tree. Although it could be very useful when the data are distributed in clusters, it fails to properly adapt to the data when they are distributed in a scattered way. In addition, both algorithms have a high computational cost because they must compute the distances between all records in the data set to build the tree structure.

Instead of structuring the data in trees or graphs, an alternative way to tackle the problem is greedily building groups of similar records. Examples of this kind are the Maximum Distance (MD) method [2] and the Maximum Distance to Average Vector (MDAV) method [4,6]. A slight modification of the same construction was described in [7] under the name Centroid-Based Fixed-Size Micro-aggregation and, in [12] Solanas et al. proposed an improvement on MDAV named Variable-MDAV, V-MDAV for short. The main advantage of these methods is their simplicity but their computational complexity (*i.e.* MD has a cubic cost and the others have a quadratic cost) prevents their use with very large data sets.

In [13] genetic algorithms are used to micro-aggregate small data sets. Although this method performs very well with small amounts of data (*i.e.* with less than 100 records), it cannot be applied to large data sets. An improvement of this method that mixes MDAV and genetic algorithms was proposed in [8].

3 Our Proposal

In this section we define our proposal. Firstly, we explain its foundational ideas and a high-level algorithm. Afterwards, we provide some details of the mathematical model and, we analyse the behaviour of our method for several values of the parameters.

3.1 Rationale

Current micro-aggregation algorithms are very costly (*i.e.* at least $O(n^2)$) because they compute the similarity (*e.g.* the squared Euclidean distance) between all possible pairs

[2] A k-partition of **D** is a partition where its parts have, at least, k records of **D**.

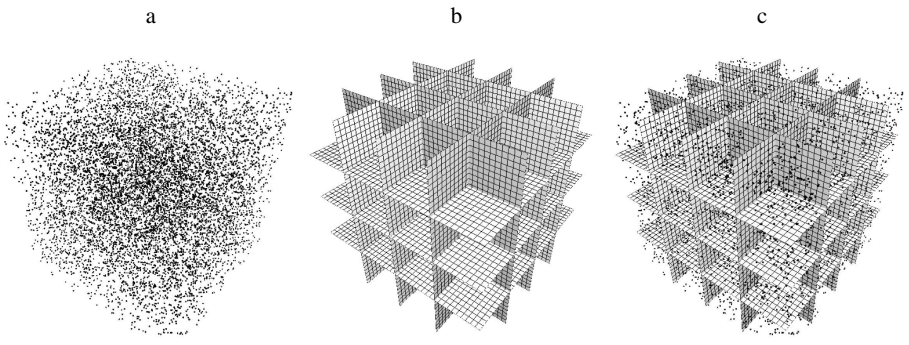


Fig. 2. Graphical example of data partitioning. a) Set of points uniformly distributed in a three dimensional space. b) Hypercubes of edge $\ell = 0.25$ generated by our algorithm. c) Wrapping of points by the generated hypercubes.

of records in the original data set. Once these similarities are known, the records are grouped.

Our proposal is absolutely different. Instead of using the similarity between records to group them, we partition the hyper-space to which the records belong. By partitioning the hyper-space we are implicitly grouping the records that are distributed inside it. By construction, the records that are in the same hypercube are very likely to be similar, thus, micro-aggregating them in the same group leads to the reduction of the SSE.

The most important contribution of our proposal is that our method dramatically reduces the time needed to micro-aggregate a given data set whilst the information loss is not significantly different from the one obtained by other costlier solutions.

Take as an example the set of records depicted in Figure 2(a). It is apparent that computing the distance between all possible pairs of records would be extremely costly. On the contrary, computing the hypercubes depicted in Figure 2(b) is easy and computationally cheap. Finally, assigning each record to a group (*i.e.* to a hypercube) is very easy thanks to the regularity of the hypercubes. Figure 2(c) depicts how the records are wrapped by their corresponding hypercubes.

3.2 Algorithm

Our method receives as input a data set **D** containing n records, and a security parameter k. Firstly, the records in **D** are normalised (step 1 in Algorithm 1) to assure that the hyper-space $\mathcal{HS}$ has a unitary edge (*i.e.* the length of the hypercube that wraps $\mathcal{HS}$ is 1). Next, the optimal length ℓ of the hypercubes is computed according to the number of records in $\mathcal{HS}$ and the privacy parameter k (step 2 in Algorithm 1) *cf.* to Section 3.5 for further details on the computation of ℓ. Once ℓ is determined, the set of hypercubes (HC) is created (step 3 in Algorithm 1). A hypercube is defined by its bounds in each dimension. For example, the bounds $\{(0,0.25),(0,0.25)\}$ define a two-dimensional hypercube (*i.e.* a square) that wraps all the points (x,y) such as $0 < x \leq 0.25$ and $0 < y \leq 0.25$. Each generated hypercube (HC_i) is identified by a unique index (i).

Algorithm 1. Efficient Hyper-Cubes-based Micro-aggregation

```
   Data: Data set D, Integer k, Integer n.
   Result: k-partition.
 1 D' ⟵Normalise(D);
 2 ℓ ⟵ ComputeHypercubeEdgeLength(n,k);
 3 HC ⟵ GenerateSetOfHypercubes(ℓ);
 4 for Each record r_i in D' do
 5 |   < HC_j, r_i > ⟵ AssingRecordToHypercube(HC,r_i);
 6 |   Card(HC_j) := Card(HC_j) + 1;
 7 end
 8 for Each Hypercube in HC do
 9 |   if Card(HC_j)< k then
10 |   |   Neigh = DetermineHypercubeNeighbour(HC_j);
11 |   |   FuseHypercubes(HC_j,Neigh);
12 |   |   HC ⟵ RemoveHypercubeFromSetOfHypercubes(HC_j);
13 |   |   Card(Neigh) := Card(Neigh)+Card(HC_j);
14 |   end
15 end
16 return set of tuples < HC_j, r_i >
```

After generating HC, each record in $\mathcal{HS}$ has to be assigned to a hypercube HC_i (step 5 in Algorithm 1). This assignment can be understood as a tuple of two indexes: the first index refers to the hypercube HC_j that contains the record and, the second index identifies the record r_i. Once a record is assigned to a hypercube, the cardinality of the hypercube (*e.g.* a counter) is increased (step 6 in Algorithm 1).

At this stage of the process, all records are already assigned to a hypercube. However, due to the fact that ℓ is an approximation (*cf.* Section 3.5), a little fraction of hypercubes can be smaller and, thus, they can contain less than k records. The aim of this algorithm is to obtain a k-partition. Hence, all hypercubes must contain, at least, k records. In order to cope with this constraint, low-cardinality hypercubes are fused with their closest neighbours (steps 9-14 in Algorithm 1). Note that, by construction, every low-cardinality hypercube has a complete hypercube (*i.e.* one having at least k records) adjacent to it, and the fusion of these hypercubes is straightforward and computationally cheap. Once all low-cardinality hypercubes have been fused, the algorithm finishes and returns the set of tuples $< HC_j, r_i >$ that represent the obtained k-partition.

3.3 Analytical Model

We assume that records in a data set **D** can be represented as points randomly scattered on a d-dimensional hyper-space $(\mathcal{HS})$.

Our proposal is based on the partition of $\mathcal{HS}$ in d-dimensional hypercubes (HCs) of edge ℓ. We indicate each of the $(1/\ell)^d$ hypercubes with a unique index (HC_i). Further, in a similar way, it is straightforward to find a bijection that binds a point in $\mathcal{HS}$ to a specific hypercube. Hence, a point q belonging to $\mathcal{HS}$ can be identified in two ways: first, using its d co-ordinates; second, via a tuple $q_{<i,j>}$, where the index i points to the

hypercube of index i (HC_i), and the index j refers to a unique index that identifies a unique point within HC_i.

Let $X_{i,j}$ be the random variable that takes on the value 1 if the point $q_{<i,j>}$ lies within HC_i, and 0 otherwise. Let C_i be the random variable that counts the number of points lying within HC_i. If we note that $E[X_{i,j}] = \ell^d$ since the points are uniformly distributed, we have that:

$$\mu_i = E[C_i] = E\left[\sum_{i=1}^{n} X_{i,j}\right] = \sum_{i=1}^{n} E[X_{i,j}] = \sum_{i=1}^{n} \ell^d = n\ell^d$$

Now, since the points in HC_i are *iid*, we have that for $0 < \varepsilon < 1$ the following equation holds:

$$Pr[|C_i - \mu_i| > \varepsilon\mu_i] \leq 2 \times exp\left(-\frac{\varepsilon^2}{3}\mu_i\right) \tag{3}$$

Note that Equation 3 provides an upper bound on the probability that a single HC_i has a number of points diverging from its mean by more (or less) than $\varepsilon\mu_i$. We are interested in identifying a condition that makes *all* of the HC_i to satisfy Equation 3. We can derive such condition as follows: let us define the event *Bad*="at least one of the hypercubes does not satisfy Equation 3", we have that this event happens with the probability given in Equation 4.

$$\begin{aligned} Pr[Bad] = \quad & Pr[|C_1 - \mu_i| > \varepsilon\mu_i \vee \ldots \vee |C_{1/\ell^d} - \mu_i| > \varepsilon\mu_i] \\ & \leq (1/\ell^d) Pr[|C_i - \mu_i| < \varepsilon\mu_i] \leq 2 \times exp\left(-\tfrac{\varepsilon^2}{3}\mu_i - \ln \ell^d\right) \end{aligned} \tag{4}$$

Note that the above equation fully characterises our model. For instance, if we set $k = \mu_i$ and $\varepsilon = 1/2$, from Equation 4 we have that:

$$Pr[Bad] \leq 2 \times exp\left(-\frac{k}{12} - \ln \ell^d\right) = 2 \times exp\left(-\frac{k}{12} - \ln \frac{k}{n}\right) = 2 \times exp\left(-\frac{k}{12} - \ln k + \ln n\right)$$

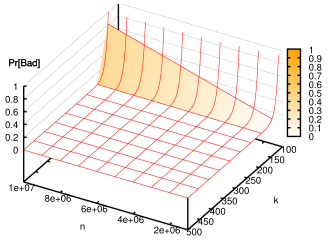
Hence, for $k = 12 \ln n$, we have:

$$Pr[Bad] \leq \frac{1}{6 \ln n}$$

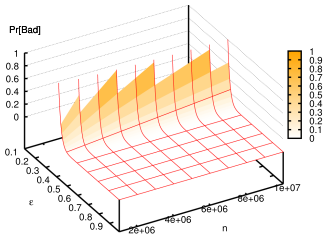
that is, all the hypercubes will have a number of points that is between $k/2$ and $2k$ with probability $\frac{1}{6 \ln n}$.

Finally, note that once set the parameters that provide the needed upper bound on the probability for the event *Bad* to happen, the corresponding value ℓ can be computed as:

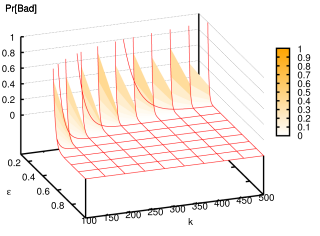
$$\ell = \sqrt[d]{\frac{k}{n}} \tag{5}$$



(a) Error graph as a function of k and n for $\varepsilon = 0.5$.



(b) Error graph as a function of ε and n for $k = 160$.



(c) Error graph as a function of ε and k for $n = 1,000,000$.

Fig. 3. 3D representation of the parameters ε, k and n

3.4 Model Validation

It is worth noticing that Equation 4, whilst providing a complete characterisation for our model, is dependant on three parameters only: n, k, and ε. To assess how these parameters affect the value of Pr[*Bad*], we have varied the value k in the range $[100,\ldots,500]$ using an incremental step of 100; the value n is varied in the range $[10^6,\ldots,10^7]$, with an incremental step of 10^6, whilst ε is varied in the range $[0.1,\ldots,0.9]$ with an incremental step of 0.1. To obtain a 3D representation, we fixed one of these three parameters, whilst allowing the other two to range in the described intervals. In particular, to obtain Figure 3(a) we set $\varepsilon = 0.5$, for Figure 3(b) we chose $k = 160$, whilst $n = 10^6$ for Figure 3(c). In each of the three figures, it can be noticed that the error probability (Pr[*Bad*]) decreases exponentially fast as free parameters increase. In particular, note from Figure 3(a) ($\varepsilon = 0.5$) that it is enough to select $k > 150$ to have Pr[*Bad*] reduced to an almost negligible value. From Figure 3(b) it turns out that for any $\varepsilon > 0.4$ the error probability Pr[*Bad*] is close to zero. Finally, from Figure 3(c) ($n = 10^6$) is apparent that for any $k \geq 100$, if one chooses $\varepsilon > 0.5$, the probability Pr[*Bad*] is very close to zero.

3.5 Tuning the ℓ Value

We normalise the points in $\mathcal{HS}$, thus, $\mathcal{HS}$ has an edge equal to one. From this result, it follows that the edge of each HC_i will be between zero and one. However, note that it could be the case that $\ell \in I$, that is ℓ belongs to the set of the irrational numbers; this

situation could not be treated on a finite state machine. Hence, we add the constraint that ℓ has to be expressed as $\ell = 1/p$, where p is an integer. Therefore we introduce the value $(\overline{\ell})$ that is, an approximation of the ideal edge for the HC_i, needed due to the finiteness of the computing machines. The simplest strategy to obtain $(\overline{\ell})$, is to set $(\overline{\ell})^{-1}$ to the smallest integer larger than $(\ell)^{-1}$, that is: $(\overline{\ell})^{-1} = \lceil (\ell)^{-1} \rceil$.

Note that in this way, we have that all but a fraction ($\hat{HC}$) of the hypercubes have an edge larger than the original ℓ. Hence, for those HC_i with edge larger than ℓ, the probability that the number of points within each HC_i is less than k further decreases. However, note that there could be a a fraction of hypercubes ($\hat{HC}$) that have an edge smaller than $(\overline{\ell})$. Indeed, this fraction will always exist whenever $\ell^{-1} \neq \lceil \ell^{-1} \rceil$. Moreover, it could be possible that the resulting edge for hypercubes in $\hat{HC}$ is even smaller than ℓ. Hence, these hypercubes could host less than k points. To cope with this issue, we extend the proposed partitioning method as follows: each of the hypercube in $\hat{HC}$ (if the edge is shorter than $(\overline{\ell})$) is merged with a neighbouring hypercube that has edge $(\overline{\ell})$. It can be shown that, by construction, such a neighbour always exists and that none of such neighbours is assigned to more than one hypercube in $\hat{HC}$. Further, from the practical point of view, as intuition suggests and experimental result will confirm, the impact of this approximation is negligible.

In an extended version of this paper we plan to show that a finer, but more complex, size for ℓ exists; however due to space limitations we consider this issue out of the scope of this paper.

3.6 Dealing with Non-uniform Data Distributions

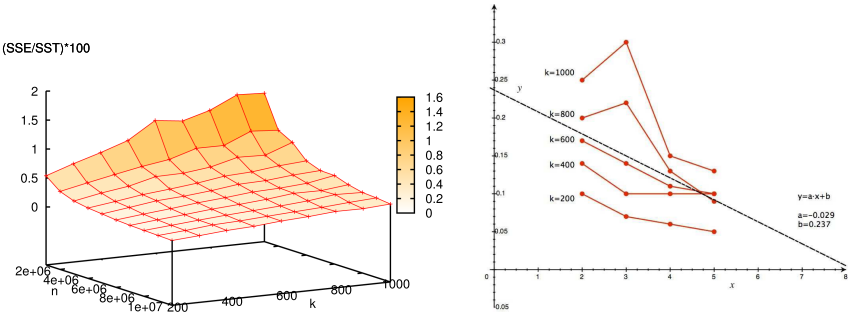
We have assumed that points in $\mathcal{HS}$ are *iid*. We are aware that, in many applications, data do not follow such a distribution pattern. However, we have detailed the mathematical foundations of the proposed method, and

we have shown that it provides very interesting results. In particular, the relative error is very low, whilst the computational cost incurred by the proposed method to partition $\mathcal{HS}$ is linear.

Furthermore, we believe that the proposed method could be used when data show a different statistical distribution, such as the Gaussian one. The key underlying idea to deal with this problem is to relax the assumption that ℓ is a constant. Indeed, we can tune ℓ to leverage the different data density in $\mathcal{HS}$ provided by a known data distribution function, so that each of the resulting HC_i still satisfy the cardinality constraint imposed by micro-aggregation. Our current investigations focus on this direction and we will tackle this problem in a future extension of this paper.

4 Experimental Results

In the following we describe the experiments that eventually show the high quality results achievable with our proposal. The experimental scenario is the following: we generated several sets of records uniformly at random in a 3-dimensional space. The size of the generated sets varied in the range $[10^6, \ldots, 10^7]$. For each of the generated sets we computed the SST (*cf.* Equation 2) and the SSE (*cf.* Equation 1) of the k-partition obtained by our method.



(a) Ratio between the SSE and SST in %.

(b) Evolution of the information loss difference between MDAV and our method

Fig. 4. Results of our method

Table 1. Brief comparison between the information loss of MDAV and our method

n	k	MDAV $\frac{SSE}{SST} \times 100$ %	Our $\frac{SSE}{SST} \times 100$ %	Diff.	n	k	MDAV $\frac{SSE}{SST} \times 100$ %	Our $\frac{SSE}{SST} \times 100$ %	Diff.
2×10^6	1000	0.75	1	0.25	4×10^6	1000	0.47	0.62	0.15
2×10^6	800	0.64	0.84	0.2	4×10^6	800	0.41	0.54	0.13
2×10^6	600	0.53	0.70	0.17	4×10^6	600	0.33	0.44	0.11
2×10^6	400	0.40	0.54	0.14	4×10^6	400	0.25	0.35	0.10
2×10^6	200	0.25	0.35	0.10	4×10^6	200	0.16	0.22	0.06
3×10^6	1000	0.57	0.87	0.3	5×10^6	1000	0.41	0.54	0.13
3×10^6	800	0.49	0.71	0.22	5×10^6	800	0.35	0.46	0.09
3×10^6	600	0.40	0.54	0.14	5×10^6	600	0.29	0.39	0.1
3×10^6	400	0.31	0.41	0.1	5×10^6	400	0.22	0.32	0.1
3×10^6	200	0.19	0.26	0.07	5×10^6	200	0.14	0.19	0.05

In Figure 4(a), on the z-axis we report the value of the ratio $(SSE/SST) * 100$, that is the percentage of the total error introduced by our micro-aggregation proposal. Note that for any value of k and n in the considered interval, the ratio does not exceed 1.6%, and that for a wide range of parameters, the same ratio is well below 0.2%. Further, the behaviour of the plot conveys another useful information: the quality of the implementation that accommodates the need of merging (few) hypercubes is highly satisfactory (*i.e.* less than 1.6% of the total error is inherited). Moreover, refining the value assigned to $(\overline{\ell})$ will further improve these results.

The fact that our algorithm has a linear cost $O(n)$ is a clear advance in the field of privacy protection by means of micro-aggregation. However, we have also compared the results of our method with a well-known $O(n^2)$ micro-aggregation algorithm *i.e.* MDAV [4,6] to proof that the dramatic computational cost reduction does not imply a significant increase in the error.

In Table 1, MDAV is compared with our method in terms of information loss. Note that we have used data sets with less than 5×10^6 records. Although our method obtains the results in few minutes, MDAV needs days to obtain them when the data sets are very large *e.g.* 10×10^6. The results in Table 1 indicate that MDAV is better than our method in terms of information loss. However the difference between both methods is not significant *i.e.* less than 0.3%. In addition, as can be clearly seen in Figure 4(b), the difference between both methods tends to decrease when the number of records increases.

These results proof that our method is useful and the only practical choice to micro-aggregate very large data sets in a reasonable time.

5 Conclusions

We have proposed an efficient multivariate micro-aggregation method that is able to micro-aggregate very large data sets in linear time $O(n)$. This is the first micro-aggregation method that achieves this goal.

We have applied our method to very large data sets of up to 10^7 records, where classical micro-aggregation algorithms cannot be applied due to their high computational cost. Thus, we have chosen values for k which are higher than the common ones. If it is necessary to obtain groups of very small cardinality, it is possible to use our method to partition the data in groups of, let's say a thousand records, and then apply a classical micro-aggregation algorithm to refine the k-partition.

In this paper we have established the foundations of a novel and efficient micro-aggregation algorithm. In the near future we plan to extend our proposal by investigating the following lines:

- Extend the algorithm to cope with non-uniform data
- Analyse the use of our algorithm as a pre-process, whose results could be used by a classical micro-aggregation algorithm to obtain a better k-partition in a reasonable time.
- Improve the SSE of our algorithm by recursively divide the hypercubes with more than $2k - 1$ records.

Acknowledgements

The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organisation. This work was partly supported by the Spanish Ministry of Education through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES", and by the Government of Catalonia under grant 2005 SGR 00446.

References

1. 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, Los Alamitos (2004)
2. Domingo-Ferrer, J., Mateo-Sanz, J.M.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and Data Engineering 14(1), 189–201 (2002)

3. Domingo-Ferrer, J., Sebé, F., Solanas, A.: A polynomial-time approximation to optimal multivariate microaggregation. Comput. Math. Appl. 55(4), 714–732 (2008)
4. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogenerous k-anonymity through microaggregation. Data Mining and Knowledge Discovery 11(2), 195–212 (2005)
5. HIPAA. Health insurance portability and accountability act (2004), http://www.hhs.gov/ocr/hipaa/
6. 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 (May 2005)
7. Laszlo, M., Mukherjee, S.: Minimum spanning tree partitioning algorithm for microaggregation. IEEE Transactions on Knowledge and Data Engineering 17(7), 902–911 (2005)
8. Martinez-Balleste, A., Solanas, A., Domingo-Ferrer, J., Mateo-Sanz, J.M.: A genetic approach to multivariate microaggregation for database privacy. In: IEEE 23rd International Conference on Data Engineering ICDE, April 17-20, 2007, pp. 180–185 (2007)
9. Oganian, A., Domingo-Ferrer, J.: On the complexity of optimal microaggregation for statistical disclosure control. Statistical Journal of the United Nations Economic Comission for Europe 18(4), 345–354 (2001)
10. European Parliament. DIRECTIVE 2002/58/EC of the European Parliament and Council of 12 july 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)(2002), http://europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf
11. Canadian Privacy. Canadian privacy regulations (2005), http://www.media-awareness.ca/english/issues/privacy/canadian_legislation_privacy.cfm
12. Solanas, A., Martínez-Ballesté, A.: V-MDAV: Variable group size multivariate microaggregation. In: COMPSTAT 2006, Rome, pp. 917–925 (2006)
13. Solanas, A., Martinez-Balleste, A., Mateo-Sanz, J.M., Domingo-Ferrer, J.: Multivariate microaggregation based on genetic algorithms. In: 3rd International IEEE Conference on Intelligent Systems IS, pp. 65–70 (2006)
14. USPrivacy. U.S. privacy regulations (2005), http://www.media-awareness.ca/english/issues/privacy/us_legislation_privacy.cfm
15. Willenborg, L., DeWaal, T.: Elements of Statistical Disclosure Control. Springer, New York (2001)