A Genetic Approach to Multivariate Microaggregation for Database Privacy

Antoni Martínez-Ballesté, Agusti Solanas, Josep Domingo-Ferrer and Josep M. Mateo-Sanz
Dept. of Computer Engineering and Maths,
Universitat Rovira i Virgili,
Av. Països Catalans 26,
E-43007 Tarragona, Catalonia
e-mail {antoni.martinez,agusti.solanas}@urv.cat
e-mail {josep.domingo,josepmaria.mateo}@urv.cat

Abstract

*Microaggregation is a technique used to protect privacy in databases and location-based services. We propose a new hybrid technique for multivariate microaggregation. Our technique combines a heuristic yielding fixed-size groups and a genetic algorithm yielding variable-sized groups. Fixed-size heuristics are fast and able to deal with large data sets, but they sometimes are far from optimal in terms of the information loss inflicted. On the other hand, the genetic algorithm obtains very good results (*i.e. *optimal or near optimal), but it can only cope with very small data sets.*

Our technique leverages the advantages of both types of heuristics and avoids their shortcomings. First, it partitions the data set into a number of groups by using a fixed-size heuristic. Then, it optimizes the partitions by means of the genetic algorithm. As an outcome of this mixture of heuristics, we obtain a technique that improves the results of the fixed-size heuristic in large data sets.

1 Introduction

Microdata, *i.e.* sets of records containing information on individual respondents, are important for planning or analysis purposes in a number of human activities: healthcare, market analysis, public policies, etc. As a consequence, statistical agencies and large enterprises routinely gather such microdata. However, this information cannot be freely released because the privacy of the respondents would be jeopardized. Against common belief, suppressing the direct identifiers (names, passport numbers, etc.) is not enough to ensure respondent privacy: *e.g.* if a data set contains civil status, age and sensitive information (salary, diseases, etc.) for a number of people, a 17-year old widow is easy to re-identify even if names have been suppressed in the data set.

A plethora of methods for statistical database privacy collectively known as statistical disclosure control (SDC) methods have been developed, which include on-line database query control [19] and data perturbation. In the latter case the aim is to anonymize the collected information so that the privacy of respondents is preserved and the anonymized data are still useful. Microaggregation is a family of SDC methods achieving anonymization through data perturbation. Quite recently, microaggregation has been shown to be also applicable to privacy problems different from statistical databases, namely location privacy [3].

The microaggregation problem can be stated as a clustering problem with restrictions on the size of clusters. Specifically, given a security parameter k and a data set $\mathbf{X}$ consisting of n records with p attributes each, the microaggregation problem consists in obtaining a partition of $\mathbf{X}$, such that each group in the partition has at least k records and the within-groups homogeneity is maximized. The resulting partition is called k-partition [5]. Once the k-partition is built, a microaggregated data set $\mathbf{X}'$ can be obtained by replacing each record in $\mathbf{X}$ by the centroid (*i.e.* the average vector) of the group to which it belongs. This clearly anonymizes original records, because all microaggregated records within a group are the same. The requirement of maximum within-groups homogeneity in the original data set is intended to minimize the information loss caused by microaggregation, that is, by the replacement of the original records in $\mathbf{X}$ by the centroids of their groups.

There are several group homogeneity measures based on different distance definitions (*e.g.* Euclidean distance, Chebyshev distance, Minkowski distance, etc.). The most common homogeneity measure for clustering is the within-groups sum of square errors (SSE) [8, 22, 10, 11, 16]. This

1-4244-0832-6/07/$20.00 ©2007 IEEE.

is the sum of squared Euclidean distances from the centroid of each group to every element in the group. For a given k-partition, the SSE is computed as:

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

where s is the number of groups in the k-partition, n_i is the number of elements in the i-th group, $\mathbf{x}_{ij}$ is the j-th element in the i-th group and $\bar{\mathbf{x}}_i$ is the centroid of the i-th group.

Given a data set, the optimal k-partition is the one with minimum SSE. The optimal k-partition of a univariate data set can be found in polynomial time [12]. However, finding the optimal k-partition for a multivariate data set is NP-hard [17]. Thus, multivariate microaggregation —*i.e.* aiming at minimizing the SSE of multivariate data— must be heuristically tackled. However, it is known that groups in an optimal k-partition contain between k and $2k-1$ records [5] and heuristics make use of this important knowledge. Multivariate microaggregation heuristics in the literature fall into two categories:

Fixed-size All groups in the resulting k-partition are of size k except perhaps one, which is of size between k and $2k-1$. Heuristics in this class can be found in [2, 5, 14, 7, 9].

Variable-size All groups can have sizes varying between k and $2k-1$, depending on the natural grouping of the data. Heuristics in this class include [5, 18, 15, 9, 4, 20, 21]. They are normally slower than fixed-size heuristics.

1.1 Contribution and plan of this paper

In this paper we show how to construct a hybrid heuristic for multivariate microaggregation offering the lower information loss caused by variable-size microaggregation and taking advantage of the speed of fixed-size microaggregation. The idea is to combine a fixed-size heuristic with the genetic microaggregation algorithm in [21] by these authors.

Our proposal can be divided in two main steps, namely group generation and group refinement. During the first step we apply the fixed-size heuristic to the original data set in order to obtain a number of groups with a cardinality not less than k. In the second step, we use the genetic algorithm on each group to refine it and we finally obtain a k-partition of the original data set.

Thanks to the genetic algorithm, our technique improves the results of the fixed-size heuristic in terms of within-groups homogeneity. Although our construction works with any fixed-size heuristic, empirical results in this paper have been computed for the specific case of MDAV [14, 7], arguably the most broadly used heuristic of this class.

The rest of the article is organized as follows. Section 2 recalls the MDAV heuristic and the main idea of our approach to solving microaggregation with a genetic algorithm. Section 3 describes how to combine the genetic algorithm with a fixed-size heuristic to obtain better results. Section 4 reports some experimental results for the specific case of MDAV. Finally Section 5 contains some concluding remarks and lines for future work.

2 Background

In this section we recall the MDAV heuristic and our approach to solving microaggregation using a genetic algorithm.

2.1 The MDAV heuristic

MDAV is a multivariate microaggregation method that was implemented as part of the μ-Argus [14] package for statistical disclosure control; its description can be found in [7]. Algorithm 1 is a short description of MDAV.

The goal of any microaggregation method is to generate a microaggregated data set protecting the privacy of the respondents. To do so, it is necessary to find a k-partition which maximizes within-groups homogeneity. MDAV builds a k-partition as follows. First, a square matrix of distances between all records is computed (line 1 of the algorithm). Two main approaches can be adopted to perform these distance calculations. The first approach is to compute and store the distances at the beginning of the microaggregation process. The second approach consists in computing the distances on the fly when they are needed. The first approach is computationally cheaper but it requires too much memory when the number of records in the data set is large. In this paper, we have used the first approach. However, the second one could be used as well without affecting the proposed technique nor the reported results.

After computing the matrix of distances, MDAV iterates (lines 4-13) and builds two groups at each iteration. In order to build these groups, the centroid c —*i.e.* the average vector— of the remaining records —those which have not yet been assigned to any group— is computed at the beginning of each iteration (line 5). Then the most distant record r from c is taken (line 6) and a group of k records is built around r (line 8). The group of k records around r is formed by r and the $k-1$ closest records to r. Next, the most distant record s from r is taken (line 7) and a group of k records is built around s (line 10). The generation of groups continues until the number of remaining records (RR) is less than $2k$. When this condition is met, two cases are possible, namely $RR < k$ or $RR \geq k$. In the first

Algorithm 1: Maximum Distance to Average Vector (MDAV) algorithm

Data: Dataset $\mathbf{X}$, Integer k, Integer n.
Result: k-partition.
ComputeDistanceMatrix($\mathbf{X}$);
$RR = n$; //Remaining Records
$i = 0$;
while $RR < 2k - 1$ **do**
  c = ComputeNewCentroid($\mathbf{X}$);
  r = GetFarthestRecordFromCentroid($\mathbf{X}$,c);
  s = GetFarthestRecordFromRecord ($\mathbf{X}$,r);
  g_i = BuildGroupAroundRecord(r,$\mathbf{X}$,k);
  $i = i + 1$;
  g_i = BuildGroupAroundRecord(s,$\mathbf{X}$,k);
  $i = i + 1$;
  $RR = RR - 2k$;
end
$g_1 \ldots g_s$ = AssignRemainingRec($\mathbf{X}$, $g_1 \ldots g_s$);
return $g_1 \ldots g_s$

case, the remaining records are assigned to their closest group. In the second case, a new group is built with all the remaining records. Note that all groups have k elements except perhaps the last one. Thus, the generated k-partition is a candidate to optimality because it fulfills the necessary condition that the cardinality of all its groups is between k and $2k - 1$.

Finally, given the k-partition obtained by MDAV, a microaggregated data set is computed by replacing each record in the original data set by the centroid of the group to which it belongs.

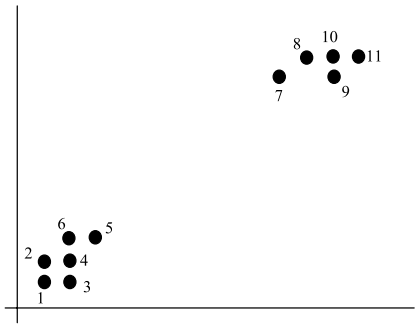

Figure 1. Example bivariate data set to illustrate the problem of MDAV.

Although the k-partition output by MDAV meets the necessary optimality condition on group cardinalities, it can be sometimes quite far from the actual optimum. In Figure 1 an example illustrating this problem is shown. For the given example, MDAV returns the 3-partition $\{\{1,2,3\},\{4,5,6,7,8\},\{9,10,11\}\}$. Clearly, the resulting 3-partition is not optimal in terms of within-groups homogeneity: an optimal k-partition would be $\{\{7,8,9,10,11\},\{4,5,6\},\{1,2,3\}\}$. It becomes apparent that the lack of flexibility in the size of the groups splits "natural" groups. As a result, the obtained k-partition is far from optimal.

2.2 A genetic algorithm for multivariate microaggregation

Our approach to microaggregation corresponds to a common Genetic Algorithm (GA) scheme [13] with some modifications to adapt it to the microaggregation task on multivariate data sets. Further details on those adaptations can be found in [21]. Specifically, our fitness function is

$$F = \frac{1}{SSE + 1} \tag{2}$$

where SSE is computed using Expression 1. Our GA uses the roulette wheel selection algorithm, which is based on weighing the probability of selecting a chromosome proportionally to its fitness. We have used the most common genetic operators (*i.e.* simple one-point crossover and mutation).

Following the recommendations of [21], we use the values of Table 1 for the main parameters of the GA.

Table 1. Values for the main GA parameters

Chromosomes	Mut.Rate	Cross.Rate	Iterations
10	0.1	0.5	10^4

3 Our method

As shown in [21], the GA outperforms fixed-size heuristics in terms of SSE when applied to very small data sets. Thus, our method consists of: i) partitioning the original data set using a fixed-size heuristic; ii) applying the GA on the groups output by the fixed-size heuristic. It is not straightforward to determine how to partition the original data set in order to obtain groups that are suitable for GA-based optimization. Next, we analyze two different partition strategies.

3.1 One-step partitioning

A simple approach for partitioning a data set with n elements is as follows:

1. Let k be a small value, (*e.g.* $k = 3$).
2. Let K be larger than k and small enough to be suitable for our GA with a reasonable number of iterations (*e.g.* $K = 20$).
3. Use a fixed-size heuristic to build a K-partition.

Finally, the GA is applied to each group of the K-partition in order to obtain an optimal or near optimal k-partition.

One-step partitioning has an important shortcoming, though. Running a fixed-size heuristic to get groups of cardinality K may split natural groups which would be better preserved if a lower cardinality parameter k was used. Figure 2 shows, on the left, the groups obtained using MDAV with $K = 24$. It can be observed that some nearby records are assigned to different groups. Hence, using one-step partitioning prevents the GA from finding solutions with a better SSE.

3.2 Two-step partitioning

With the aim of averting the shortcomings of the one-step partitioning method, we propose an alternative method that uses the fixed-size heuristic twice rather than once. With the proposed two-step partition method, the number of split natural groups is significantly reduced.

The proposed method can be summarized as follows:

1. Let k be a small value (*e.g.* $k = 3$).
2. Let K be larger than k and divisible by k, small enough to be suitable for our GA (*e.g.* $K = 21$).
3. Use a fixed-size heuristic to build a k-partition.
4. Taking as input records the average vectors obtained in the previous step, apply the fixed-size heuristic to build *macrogroups* (*i.e.* sets of average vectors) of size K/k.
5. For each given *macrogroup*, replace the average vectors by the k original records to obtain a K-partition.

Finally, apply the GA to each macrogroup in the K-partition in order to generate an optimal or near optimal k-partition of the macrogroup. The composition of the k-partitions of all macrogroups yields a k-partition for the entire data set. An important issue when using the GA on a macrogroup is to include the fixed-size k-partition of the macrogroup induced by the k-partition of the entire data set computed in Step 3 above as one of the chromosomes of the initial population fed to the GA.

Figure 2 shows, on the right, the groups built using the two-step partitioning method with $K = 24$ and $k = 3$. It can be observed that the resulting groups are better than those obtained with the one-step partitioning method. The following result holds:

Lemma 1 *The SSE of the k-partition obtained using a two-step partition based on a fixed-size heuristic followed by a GA is not greater than the SSE of the k-partition output by the fixed-size heuristic alone.*

Proof: By assumption, the fixed-size k-partition of each macrogroup induced by the fixed-size k-partition of the data set obtained at Step 3 is one of the chromosomes of the initial population fed to the GA. Therefore, for each macrogroup, the k-partition output by the GA is at least as good as the fixed-size k-partition in terms of the fitness function. This implies that the SSE of the k-partition of the entire data set obtained by composition of the macrogroup k-partitions output by the GA is not greater than the SSE of the k-partition of Step 3. □.

4 Experimental results

In this section we summarize the empirical results obtained with our hybrid technique. MDAV has been used as a fixed-size heuristic. The two partitioning methods described above and the GA have been tested on four different data sets, each of them presenting particular features in terms of data dispersion and natural grouping.

The data sets used are next described:

Synthetic data sets "Scattered" and "Clustered" are data sets that contain $n = 1000$ records with $p = 2$ attributes. The former is *scattered* in the sense that no natural clusters are apparent. Attribute values were drawn from the $[-10000, 10000]$ range by simple random sampling. The second data set is considered to be *clustered* because its records are grouped in natural clusters. The attribute values were uniformly drawn from $[-10000, 10000]$ as in the previous case, but records were then groupwise shifted by a random amount in order to form clusters of size between 3 and 5 records each; more details on the generation of this data set can be found in [4].

Real data sets "Census" is a data set that contains 1080 records with 13 numerical attributes. "EIA" contains 4092 records with 11 numerical attributes. These data sets were proposed as reference microdata sets during the CASC project [1] and have been widely used [5, 6, 7, 15].

Table 2 shows the results of running the one-step and two-step partitioning methods on the aforementioned data sets. Microaggregation has been performed with $k = 3$,

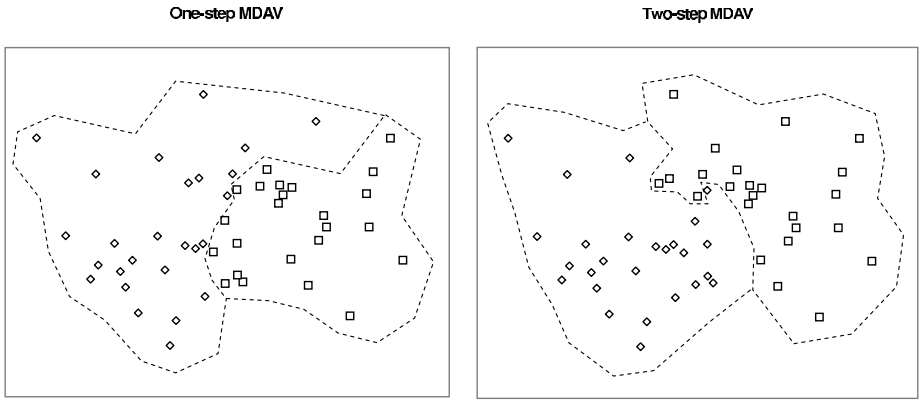

Figure 2. Left, one-step MDAV partitioning of the data ($K = 24$). Right, two-step MDAV partitioning of the data ($K = 24$, $k = 3$).

which is a common choice. Different values of K have been tested. Results comprise the number of macrogroups generated as well as the SSE of the microaggregated data sets using both MDAV on the whole set and GA on the macrogroups. We have used the GA parameters given in Table 1. Using these parameters, we have run our C++ GA implementation on a macrogroup of 27 elements, and it takes 2 seconds on a Pentium 4 processor running at 3 GHz under a Linux operating system.

It can be observed in Table 2 that, in general, the results of the one-step partitioning method are not better than the results yielded by the MDAV heuristic alone. Except for the "Scattered" data set with $K = 27$ and the "Clustered" data set, the breaking of several natural clusters by the one-step partitioning approach decreases the within-groups homogeneity of the microaggregated data set (increasing SSE). On the other hand, it can be noticed that in all cases the SSE of the data sets microaggregated using our GA after partitioning the data with the two-step partitioning method are better than the SSE obtained using MDAV only. The improvements achieved range from 1% to 41%.

5 Conclusions and future work

Microaggregation is a technique used to protect privacy in databases and location-based services. We have proposed a new hybrid technique for multivariate microaggregation that combines a heuristic yielding fixed-size groups and a genetic algorithm yielding variable-sized groups. This mixture blends the low information loss (high within-groups homogeneity) of the genetic approach with the ability to cope with data sets of realistic size (typical of fixed-size heuristics).

The empirical work presented shows that, in the case of the MDAV heuristic, the hybrid approach described can reduce information loss by as much as 41%.

Future work will include tuning the parameters of the hybrid heuristic described to make it fit for applications other than statistical database privacy. Specifically, we plan to adapt it to location privacy. Finally, a comparison of the proposed heuristic with genuine variable-size algorithms mentioned in Section 1 is also under way.

Acknowledgments

This work was partly supported by the Spanish Ministry of Education through project SEG2004-04352-C04-01 "PROPRIETAS" and by the Government of Catalonia under grant 2005 SGR 00446.

References

[1] R. Brand, J. Domingo-Ferrer, and J. M. Mateo-Sanz. Reference data sets to test and compare SDC methods for protection of numerical microdata, 2002. European Project IST-2000-25069 CASC, http://neon.vb.cbs.nl/casc.

[2] D. Defays and N. Anwar. Micro-aggregation: a generic method. In *Proceedings of the 2nd International Symposium on Statistical Confidentiality*, pages 69–78, Luxemburg, 1995. Eurostat.

[3] J. Domingo-Ferrer. Microaggregation for database and location privacy. In *Next Generation Information*

Table 2. Summary of experimental results ($k = 3$)

Data set	K	SSE_{MDAV}	One-step MDAV partitioning # clus.	SSE	% impr.	Two-step MDAV partitioning # clus.	SSE	% impr.
"Scattered"	12	4.71	83	5.11	-8%	83	4.28	**9%**
	18	4.71	55	4.91	-4%	56	4.33	**8%**
	27	4.71	37	4.71	0%	38	4.58	**3%**
"Clustered"	12	3.57	83	2.26	37%	84	2.84	**20%**
	18	3.57	55	1.83	49%	56	2.14	**40%**
	27	3.57	37	1.18	67%	41	2.09	**41%**
"Census"	12	799	90	865.04	-8%	91	768	**4%**
	18	799	60	879.94	-10%	61	767	**4%**
	27	799	40	919.88	-15%	41	789	**1%**
"EIA"	12	217	341	414.56	-91%	342	189	**13%**
	18	217	227	295.78	-36%	228	186	**14%**
	27	217	151	333.26	-54%	152	197	**9%**

Technologies and Systems-NGITS'2006, volume 4032 of *Lecture Notes in Computer Science*, pages 106–116, Berlin, 2006.

[4] J. Domingo-Ferrer, A. Martínez-Ballesté, J. M. Mateo-Sanz, and F. Sebé. Efficient multivariate data-oriented microaggregation. *The VLDB Journal*, 15(4):355–369, 2006.

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

[6] J. Domingo-Ferrer, F. Sebé, and A. Solanas. A polynomial-time approximation to optimal multivariate microaggregation. *Manuscript*, 2005.

[7] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogenerous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[8] A. W. F. Edwards and L. L. Cavalli-Sforza. A method for cluster analysis. *Biometrics*, 21:362–375, 1965.

[9] E. Fayyoumi and B. J. Oommen. A fixed structure learning automaton micro-aggregation technique. In J. Domingo-Ferrer and L. Franconi, editors, *Privacy in Statistical Databases-PSD 2006*, volume 4302 of *Lecture Notes in Computer Science*, pages 114–128, Berlin, 2006.

[10] A. D. Gordon and J. T. Henderson. An algorithm for euclidean sum of squares classification. *Biometrics*, 33:355–362, 1977.

[11] P. Hansen, B. Jaumard, and N. Mladenovic. Minimum sum of squares clustering in a low dimensional space. *Journal of Classification*, 15:37–55, 1998.

[12] S. L. Hansen and S. Mukherjee. A polynomial algorithm for optimal univariate microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1043–1044, July-August 2003.

[13] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[14] A. Hundepool, A. V. de Wetering, R. Ramaswamy, L. Franconi, A. Capobianchi, P.-P. DeWolf, J. Domingo-Ferrer, V. Torra, R. Brand, and S. Giessing. *μ-ARGUS version 4.0 Software and User's Manual*. Statistics Netherlands, Voorburg NL, may 2005. http://neon.vb.cbs.nl/casc.

[15] M. Laszlo and S. Mukherjee. Minimum spanning tree partitioning algorithm for microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, 17(7):902–911, 2005.

[16] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[17] A. Oganian and 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):345–354, 2001.

[18] 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):459–476, 2002.

[19] J. Schlörer. Disclosure from statistical databases: quantitative aspects of trackers. *ACM Transactions on Database Systems*, 5:467–492, 1980.

[20] A. Solanas and A. Martínez-Ballesté. V-MDAV: Variable group size multivariate microaggregation. In *COMPSTAT'2006*, pages 917–925, Rome, 2006.

[21] A. Solanas, A. Martínez-Ballesté, J. M. Mateo-Sanz, and J. Domingo-Ferrer. Towards microaggregation with genetic algorithms. In *3rd IEEE Conference On Intelligent Systems*, pages 65–70, Westminster, 2006. IEEE Computer Society Press.

[22] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.