# Efficient Near-Optimal Variable-Size Microaggregation

Jordi Soria-Comas, Josep Domingo-Ferrer(✉), and Rafael Mulero

Department of Computer Science and Mathematics, CYBERCAT-Center for Cybersecurity Research of Catalonia, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Catalonia
{jordi.soria,josep.domingo,rafael.mulero}@urv.cat

**Abstract.** Microaggregation is a well-known family of statistical disclosure control methods, that can also be used to achieve the $k$-anonymity privacy model and some of its extensions. Microaggregation can be viewed as a clustering problem where clusters must include at least $k$ elements. In this paper, we present a new microaggregation heuristic based on Lloyd's clustering algorithm that causes much less information loss than the other microaggregation heuristics in the literature. Our empirical work consistently observes this superior performance for all minimum cluster sizes $k$ and data sets tried.

**Keywords:** Anonymization · Statistical disclosure control · Microaggregation · Lloyd's algorithm

## 1 Introduction

Collecting data and sharing them for secondary analysis is increasingly widespread and brings undoubted social and economic benefits. Yet, when data are personally identifiable information (PII), sharing them may be a threat to people's privacy. As a consequence, administrations have strengthened privacy regulation to protect the citizens. In a nutshell, these new privacy regulations, epitomized by the EU General Data Protection Regulation, require consent from data subjects for any PII collection, sharing or analysis. In the many situations in which obtaining consent is not feasible, anonymization is the only way to go. After anonymization, data no longer qualify as PII and, thus, are no longer subject to data protection regulations.

Anonymizing data involves not only suppressing any identifiers, but altering other attributes as well. The original data are first stripped from identifiers and then a statistical disclosure control method is used to mask the remaining attributes so that they no longer reveal information about original data subjects. Masking is not straightforward because, to keep the masked data statistically valid, the information loss must be minimized. Among the available statistical disclosure control techniques, in this paper we focus on microaggregation. Microaggregation replaces records in the original data set by (aggregated)

V. Torra et al. (Eds.): MDAI 2019, LNAI 11676, pp. 333–345, 2019.
https://doi.org/10.1007/978-3-030-26773-5_29

records that refer to groups of data subjects. The greater the groups, the stronger the protection. To guarantee at least a certain level of protection, microaggregation algorithms take a parameter $k$ that determines the minimum required group size.

In recent years, the research on data anonymization performed by the computer science community has focused on privacy models. A privacy model describes the condition that data must satisfy for disclosure risk to be at an acceptable level, but it does not describe how this condition should be attained. $k$-Anonymity [15] is among the most popular privacy models. It seeks to limit the probability of successful record re-identification by altering the value of quasi-identifier attributes. Quasi-identifiers are attributes that are not re-identifying when separately considered (e.g. in general Age, Profession and Zipcode do not identify anyone separately), but such that their combination may identify the subject to whom a record corresponds (there may be a single 95-year old doctor in a certain zipcode, and it may be easy to find her name in an electoral roll). Interestingly, running microaggregation on the quasi-identifiers yields $k$-anonymity [8]. Microaggregation is also useful to enforce $l$-diversity and $t$-closeness, two extensions of $k$-anonymity [7,19], as well as a building block of $\varepsilon$-differentially private algorithms [17,18].

To minimize the information loss incurred by microaggregation, we need to carefully choose the groups of records to be aggregated. A common approach in numerical microaggregation is to attempt to minimize the sum of squared distances between original records and their corresponding aggregated records, which will be called $SSE$. Unfortunately, finding a microaggregation that minimizes $SSE$ is an NP-hard problem. For this reason, existent approaches are heuristic. Most current microaggregation algorithms generate clusters with a fixed size (the minimum required cluster size). This cardinality constraint reduces the complexity of the microaggregation algorithm but it may result in large information loss. To reduce information loss, heuristic variable-size microaggregation algorithms have been proposed, but their computational complexity is greater than that of their fixed-size counterparts. Also, in some cases they need additional parameters whose optimal values are hard to determine.

**Contribution and Plan of this Work**

Microaggregation is closely related to clustering: in fact, it is clustering with a minimum cardinality constraint on clusters. In this work, we take advantage of the information loss minimization capabilities of Lloyd's clustering algorithm [12] to achieve near-optimal variable-size microaggregation. First, we embed a minimum cluster size constraint in the algorithm. Second, given that Lloyd's algorithm requires the number of clusters to be fixed beforehand, we modify it to allow a variable number of clusters. We call the resulting heuristic ONA (Near-Optimal microaggregation Algorithm). We then present empirical results on the information loss and the computing time of variable-size microaggregation with ONA.

In Sect. 2, we give some background on microaggregation and Lloyd's algorithm. In Sect. 3, we describe some limitations of current microaggregation algorithms. In Sect. 4 we present the ONA algorithm to deal with these limitations. In Sect. 5, we experimentally compare ONA with existing methods. We finalize with conclusions and future work directions in Sect. 6.

## 2 Background

### 2.1 Microaggregation

Microaggregation is a perturbative method for statistical disclosure control of microdata releases. It is based on the following two steps:

- *Partition:* The records in the original data set are partitioned into several clusters, each of them containing at least $k$ records (the minimum cluster size). To minimize information loss in the following step, records in each cluster should be as close to one another as possible.
- *Aggregation:* An aggregation operator is used to compute the centroid of all the records in the cluster. If all attributes are numerical, the centroid record is the mean record. Finally, every record in the cluster is replaced with the cluster centroid record.

When replacing records by cluster centroids in the aggregation step of microaggregation, some information is lost. The ensuing loss of variability is a measure of information loss. A microaggregation algorithm is optimal if it minimizes information loss.

Let $SST$ be the total sum of squares, that is, the sum of squared distances between each record $r$ in an original data set $D$ and the centroid record $c(D)$ of the entire data set:

$$SST = \sum_{r \in D} \|r - c(D)\|^2 .$$

Clearly, $SST$ represents the total variability of $D$. Then compute the sum of squared records errors $SSE$, that is, the sum of squared distances between each record $r$ and the centroid $c(r)$ of the cluster $r$ belongs to:

$$SSE = \sum_{r \in D} \|r - c(r)\|^2 .$$

$SSE$ represents the loss of variability incurred when replacing records with centroids. We can normalize $SSE$ by dividing it by $SST$, so that $SSE/SST$ accounts for the proportion of the total variability lost due to the microaggregation. With numerical attributes, the mean is a sensible choice as the aggregation operator, because for any given cluster partition it minimizes $SSE$ in the aggregation step; the challenge thus is to come up with a partition that minimizes the overall $SSE$.

Finding an optimal algorithm is feasible for univariate microaggregation of a numerical attribute. There are two well-known necessary optimality conditions in this case [4]: clusters must contain consecutive records and the size of the clusters must be between $k$ and $2k-1$. Given these two conditions, a shortest-path algorithm can find the optimal univariate microaggregation with cost $O(n \log n)$ for $n$ records [9].

Since realistic data sets contain multiple attributes, univariate microaggregation is not enough. Multivariate microaggregation is more complex: the first optimality condition above does not apply for want of a total order in the data domain. As a result, the search space for the optimal multivariate microaggregation remains too large and finding the optimal solution is NP-hard [14]. Therefore, heuristics are employed to obtain an approximation with reasonable cost. An example heuristic for the partition step of microaggregation is MDAV [8], which generates fixed-size clusters. Alternatively, VMDAV [16] is an adaptation of the MDAV heuristic that allows variable-size clusters.

### 2.2 MDAV

The MDAV algorithm aims at satisfying the optimality conditions of numerical univariate microaggregation:

1. *Optimal clusters must contain consecutive elements.* Since a total order is lacking in a multivariate domain, the meaning of consecutive elements is not well-defined. However, the intuition remains valid: it makes no sense to include a record $r'$ in a cluster if a record $r$ closer to the records of the cluster is not in the cluster.
2. *The size of optimal clusters ranges between* $k$ and $2k-1$. This condition remains valid in the multivariate case.

Thus, rather than minimizing the overall information loss, the MDAV heuristic proceeds by selecting specific records at the boundary of the set of records not yet assigned to any cluster and generating clusters of $k$ elements around them: given a record $r$, a cluster is formed with $r$ and the $k-1$ records closest to $r$ among those not clustered yet. See Algorithm 1.

### 2.3 VMDAV

VMDAV is an adaptation of MDAV that can yield variable-size clusters. The underlying idea is that variable-size clusters can be more adapted to the distribution of the records and, thus, reduce the information loss.

Essentially, VMDAV takes two steps: (i) generate a cluster of size $k$ that contains the record that is farthest from the average record and its closest $k-1$ records, and (ii) expand the cluster with neighboring records. These steps are repeated until all the records have been assigned to a cluster.

The first step is similar to MDAV. So we only describe the second step. Once we have a cluster with $k$ records, we look for $r_u$, the unclustered record that minimizes the distance to the records in the cluster. Let $d_{in}$ be such minimum distance. The we compute $d_{out}$, the minimum distance between $r_u$ and

**Algorithm 1.** MDAV microaggregation algorithm with minimal cluster size $k$

```
Let $D$ be a data set
Let $k$ be the minimum cluster size
$Clusters = \emptyset$
While $|D| \geq 3k$
    $x_a$=average record of $D$
    $x_r$=record of $D$ that is most distant from $x_a$
    $C$=cluster containing $x_r$ and the $k-1$ records of $D$ closest to $x_r$
    $Clusters = Clusters \cup C$
    $D = D \setminus C$
    $x_s$=record of $D$ that is most distant from $x_r$
    $C$=cluster containing $x_s$ and the $k-1$ records of $D$ closest to $x_s$
    $Clusters = Clusters \cup C$
    $D = D \setminus C$
End while
If $2k \leq |D| \leq 3k-1$ Then
    $x_a$=average record of $D$
    $x_r$=record of $D$ that is most distant from $x_a$
    $C$=cluster containing $x_r$ and the $k-1$ records of $D$ closest to $x_r$
    $Clusters = Clusters \cup C$
    $D = D \setminus C$
End if
$Clusters = Clusters \cup D$
Return $Clusters$
```

the remaining unclustered records. The cluster expansion procedure is based on these two distances. If $d_{in}$ is smaller than $d_{out}$, then $r_u$ is closer to the records in the cluster than to the other unclustered records. In that case, adding $r_u$ to the current cluster is a sensible choice. To allow tuning cluster expansion, VMDAV introduces a threshold parameter $\gamma$, so that the current cluster is expanded with $r_u$ if $d_{in} < \gamma d_{out}$.

### 2.4 Clustering and Lloyd's Algorithm

There are several approaches to generate clusters. In this work, we are interested in centroid-based clustering (a.k.a. $c$-means clustering). The purpose of $c$-means is to split the records in a fixed set of $c$ clusters in a way that $SSE$ is minimized.

Lloyd's algorithm is designed for $c$-means clustering. Starting from an arbitrary set of $c$ centroids, the algorithm proceeds by iteratively assigning each record to the closest centroid and recomputing the centroids, until a convergence criterion is met. See Algorithm 2.

The runtime of Algorithm 2 is $O(ncdi)$, where $n$ is the number of records, $c$ is the number of clusters, $d$ is the number of attributes per record and $i$ the number of iterations needed until convergence. Lloyd's algorithm is thus often considered of linear complexity in practice, although in the worst case it can be superpolynomial.

**Algorithm 2.** Lloyd's online clustering of a data set $D$ into $c$ clusters

```
Let $D$ be a data set
Let $Centroids = \{c_1, \ldots, c_c\}$ be the initial set of centroids
Let $C_i = \emptyset$ be the cluster associated with $c_i$ for $i = 1, \ldots, c$
Repeat
    For each $r \in D$
        If $r$ was assigned to a cluster $C_j$ Then extract $r$ from $C_j$
        Compute the distance between $r$ and $c_1, \ldots, c_c$
        Assign $r$ to the cluster around the closest centroid
    End for
Until convergence condition
Return $\{C_1, \ldots, C_c\}$
```

## 3 Limitations of MDAV and VMDAV

MDAV is quite effective at generating clusters that are as compact as possible: it looks for the record that is farthest from the average record and then generates a cluster that contains it and the $k - 1$ records closest to it. In this way MDAV creates compact clusters and avoids the presence of intersecting clusters, which are undesirable because their records could be rearranged in non-intersecting clusters, thereby reducing information loss. The greatest limitation of MDAV is that all clusters (except perhaps the last one) have fixed size $k$. This is much more restrictive than the optimality condition according to which cluster cardinality must be between $k$ and $2k - 1$, and it may have a significant negative impact on information loss. This limitation not only affects MDAV but all microaggregation methods that use fixed-size clusters.

VMDAV improves over MDAV by being more flexible about cluster sizes. However, the cluster expansion criterion is difficult to adjust. VMDAV uses an extra threshold parameter $\gamma$ to decide between expanding the current cluster with an additional element (up to a maximum $2k - 1$ elements) or creating a new cluster. The difficulty comes from the fact that it is not known how to fix $\gamma$ appropriately.

In [16], we find some vague recommendations, which suggest the use of large thresholds (*e.g.* $\gamma = 1.1$) when records are concentrated around specific areas of the data domain, whereas smaller thresholds (*e.g.* $\gamma = 0.2$) are preferable when records are scattered. The rationale for the rule that recommends the use of small $\gamma$ for scattered records is clear: in this case, small clusters are preferable to avoid large $SSE$. However, we should keep in mind that by using small $\gamma$ the cluster expansion mechanism is hampered, and VMDAV becomes closer to MDAV. The rationale for using large $\gamma$ when records are concentrated around specific points is unclear to us. After all, regardless of the distribution of records, we should prefer smaller clusters to larger clusters. This is illustrated in Fig. 1, where two microaggregation partitions with minimum size $k = 3$ are displayed that could be obtained using VMDAV. On the left, all clusters have size 3, which is a result compatible with VMDAV for small $\gamma$ (and also with MDAV). On the right, the size of the clusters is greater than 3, which is compatible with VMDAV for large $\gamma$. By looking at the distribution of the records, we observe that they are concentrated around two points; thus, according to the rules suggested in [16] we would select a large threshold, which would make the right-hand side partition likelier. However, $SSE$ and hence the information loss is larger for this partition than for the left-hand side partition.

The issues of VMDAV that we have hinted are confirmed in the experimental section, where VMDAV and MDAV achieve comparable levels of information loss. That is, the cluster expansion procedure of VMDAV is not capable of offering noticeable reductions in the information loss.

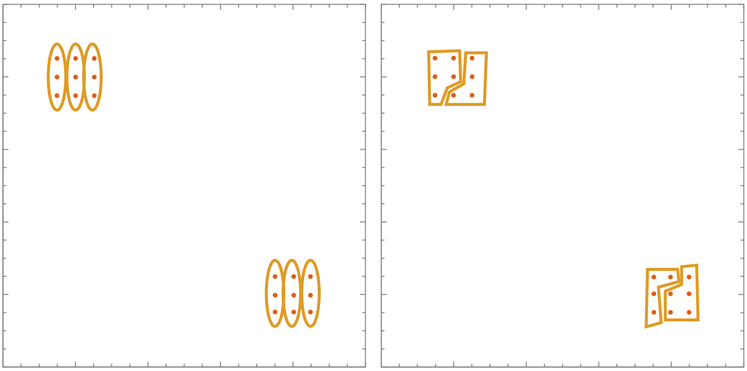

**Fig. 1.** Two microaggregation partitions with minimum size $k = 3$. Left, partition where all clusters have size 3. Right, partition where clusters have size greater than 3.

One justification for suggesting large $\gamma$ when records are concentrated in different regions is to avoid obtaining clusters that expand across more than one region. On the left-hand side of Fig. 2, we show an example of this undesirable situation. This partition, where all clusters except one have size 3, could be the result of taking $k = 3$ in MDAV or in VMDAV with small $\gamma$. Taking a large threshold in VMDAV is expected to facilitate variable-size clusters, which might solve the problem. However, as shown on the right-hand side of Fig. 2, it is not guaranteed that variable-size clusters achieve the required result: there is still a cluster spread among two regions.

Even if the previous VMDAV threshold rules were effective for data sets that are clearly concentrated or scattered, we would still be at a loss for data sets that do not qualify as any of those two types. For example, consider a data set that has several small regions with concentrated records and a big region with scattered records.

Furthermore, in general it cannot be assumed that the data controller choosing anonymization parameters knows whether her data set is scattered, concentrated, etc. In fact, for large and high-dimensional data sets, it may be quite difficult to grasp how records are distributed in the domain of attributes.

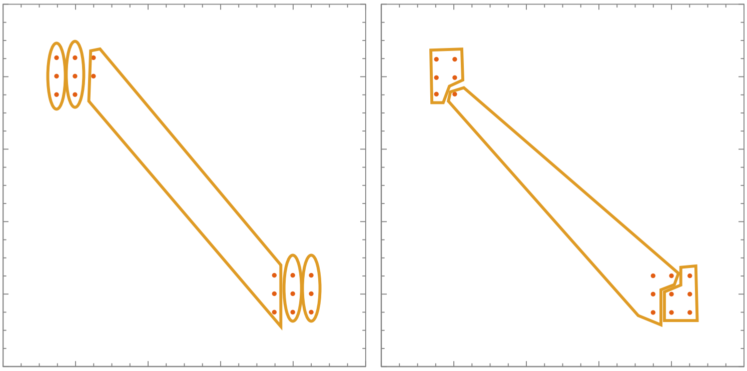

**Fig. 2.** Clusters than expand across regions. Left, partition output by MDAV with $k = 3$ or by VMDAV with $k = 3$ and small $\gamma$. On the right, partition output by VMDAV with large $\gamma$, where cluster size can vary between $k = 3$ and $2k - 1 = 5$.

In summary, fixed-size microaggregation incurs a large information loss and cluster expansion strategies such as those used in VMDAV are difficult to adjust.

## 4 ONA: Near-Optimal MicroAggregation

In this section we propose ONA (Near-Optimal microAggregation), a novel variable-size microaggregation method that is based on standard clustering algorithms. On the one hand, clustering algorithms adjust the size of each cluster automatically. We plan to take advantage of this property in ONA, while making sure that the size of the clusters stays within the known optimal bounds, that is, between $k$ and $2k - 1$. On the other side, clustering algorithms usually take the number of clusters as a parameter. In microaggregation, we do not care about the number of clusters; we simply want a valid clustering that minimizes the information loss. Thus, the need to tell the microaggregation algorithm the number of clusters we want would be an artificial restriction that we prefer to avoid, both for the sake of algorithm clarity and to avoid unnecessary information loss.

ONA follows Lloyd's online algorithm (see Algorithm 2) but it makes several adjustments to guarantee that an appropriate number of clusters with an appropriate size is generated. Algorithm 3 formalizes ONA and its steps are explained next:

- We start (at line 3) by generating a random set of clusters whose cardinality is $k$ or more. The minimum cardinality constraint of microaggregation is enforced by starting with a set of clusters that conforms to it and by making sure that any modification of the clusters does not violate it.
- The proposed algorithm is iterative. Each iteration (lines 4–29) is designed to reduce the $SSE$ of the clustering, until convergence is reached. The convergence condition is not specified in the algorithm. To be strict, we should

require a completely stable set of clusters. However, as most of the reduction in $SSE$ is attained in the first few iterations, it is usually safe to use less strict conditions to speed up the execution. We will describe alternative convergence conditions when reporting experiments in Sect. 5.

- Following Lloyd's online algorithm, loop through the records (lines 5–28) in the data set and reassign them (if needed) to the closest cluster so that $SSE$ decreases.
- It is only possible to reassign a record if its current cluster contains more than $k$ records (lines 7–11). Otherwise, there would remain less than $k$ records in the cluster and the clustering would not satisfy the minimum cardinality constraint. If the cluster of the current record has more than $k$ records, remove the record from the cluster (line 9) and assign it to the closest cluster (line 11).
- When the cluster of the current record has $k$ records, the only way to reassign the current record to another cluster is to dissolve the cluster and reassign all its records to other clusters (lines 12–20). This is only done if it reduces $SSE$. In line 15 all reassignments are computed: $C_{j(s)}$ is the cluster to which record $s$ is reassigned. The contribution to $SSE$ of the original clusters ($SSE_1$, line 16) and the $SSE$ of the reassigned clusters ($SSE_2$, line 17) are computed. If $SSE_2 < SSE_1$, the reassignments are applied; otherwise, the current clustering is kept unmodified.
- Finally, the algorithm checks that all clusters have at most $2k - 1$ records (as one of the optimality conditions requires). This condition must be checked because the reassignments can make clusters grow beyond $2k - 1$ records. If a cluster with $2k$ or more records is found, we apply the same Algorithm 3 to the cluster, which will split it into two clusters of size between $k$ and $2k - 1$ thereby reducing $SSE$.

In spite of the distinction between the current cluster having more than $k$ records or $k$ records, the complexity of Algorithm 3 remains essentially the same as the one of Lloyd's algorithm (see Sect. 2.4).

# 5 Experimental Evaluation

## 5.1 Evaluated Methods

The motivation of our algorithm has been based on the limitations of MDAV and VMDAV. However, for completeness, the experimental section will not be limited to comparing with those two methods. We will compare the information loss using $SEE$ and $100 \times SSE/SST$ (as described in Sect. 2.1) for the following methods: MDAV [4], VMDAV [16], MD-MHM [3], MDAV-MHM [3], CBFS-MHM [3], NPN-MHM [3], $\mu$-Approx [6], M-d [10], TFRP-1 [2], TFRP-2 [2], DBA-1 [11], DBA-2 [11] and IMHM [13].

**Algorithm 3.** ONA algorithm for a data set $D$ and minimal cluster size $k$.

```
Let $D$ be a data set
Let $k$ be the minimal cluster size
Randomly generate a set of clusters $\mathcal{C} = \{C_1, \ldots, C_{\lfloor |D|/k \rfloor}\}$ such that each cluster contains at least $k$ records
Repeat
    For each $r \in D$
        Let $C_{i(r)} \in \mathcal{C}$ be the cluster that contains $r$
        If $|C_{i(r)}| > k$ Then
            // Should $r$ be reassigned to another cluster?
            Extract $r$ from $C_{i(r)}$
            Compute the distance between $r$ and the centroids of the clusters in $\mathcal{C}$
            Add $r$ to the cluster whose centroid is closest to $r$
        Else If $|C_{i(r)}| = k$ Then
            // Should cluster $C_{i(r)}$ be dissolved?
            Let $C_{j(s)}$ be the cluster with the closest centroid to $s \in C_{i(r)}$ among those in $\mathcal{C} \setminus C_{i(r)}$
            Let $C'_k = C_k \cup \{s \in C_{i(r)} : j(s) = k\}$, for each $k \neq i(r)$
            Let $SSE_1 = SSE(C_{i(r)}) + \sum_{k \in \{j(s) : s \in C_{i(r)}\}} SSE(C_k)$
            Let $SSE_2 = \sum_{k \in \{j(s) : s \in C_{i(r)}\}} SSE(C'_k)$
            If $SSE_1 > SSE_2$ Then
                $\mathcal{C} = \{C'_k : k \neq i(r)\}$
            End if
        End if
        // Split clusters that have become too large
        For each $C \in \mathcal{C}$
            If $|C| \geq 2k$ Then
                Run Algorithm 3 on $C$ with minimal cluster size $k$
            End if
        End for
    End for
Until convergence_condition
```

### 5.2 Data Sets

The evaluation was performed on data sets [1] that have been used in the literature to evaluate microaggregation algorithms:

- *Census.* Data set with 1080 records and 13 numerical attributes.
- *Tarragona.* Data set with 834 records and 13 numerical attributes.
- *EIA.* Data set with 4092 records and 11 numerical attributes.

### 5.3 Evaluation Results

The evaluation results are shown in Table 1. We observe that, while there are only small differences in the information loss reported by other methods, our proposal achieves a significantly smaller information loss. This behavior is consistent across cluster sizes and data sets.

**Table 1.** Information loss $100 \times SSE/SST$ for several values of $k$ and several data sets

| Data set | Method | $k = 3$ | $k = 5$ | $k = 10$ |
|---|---|---|---|---|
| Census | **ONA** | **1.59** | **2.33** | **3.88** |
| | MDAV | 5.69 | 9.09 | 14.16 |
| | VMDAV | 5.69 | 8.98 | 14.07 |
| | MD-MHM | 5.69 | 8.99 | 14.40 |
| | MDAV-MHM | 5.65 | 9.08 | 14.22 |
| | CBFS-MHM | 5.67 | 8.89 | 13.89 |
| | NPN-MHM | 6.34 | 11.34 | 18.73 |
| | $\mu$-Approx | 6.25 | 10.78 | 17.01 |
| | M-d | 6.11 | 10.30 | 17.17 |
| | TFRP-1 | 5.93 | 9.36 | 14.44 |
| | TFRP-2 | 5.80 | 8.98 | 13.96 |
| | DBA-1 | 6.15 | 10.84 | 15.79 |
| | DBA-2 | 5.58 | 9.04 | 13.52 |
| | IMHM | 5.37 | 8.42 | 12.23 |
| Tarragona | **ONA** | **5.75** | **9.54** | **14.40** |
| | MDAV | 16.93 | 22.46 | 33.19 |
| | VMDAV | 16.96 | 22.88 | 33.26 |
| | MD-MHM | 16.98 | 22.53 | 33.18 |
| | MDAV-MHM | 16.93 | 22.46 | 33.19 |
| | CBFS-MHM | 16.97 | 22.53 | 33.18 |
| | NPN-MHM | 17.39 | 27.02 | 40.18 |
| | $\mu$-Approx | 17.10 | 26.04 | 38.80 |
| | M-d | 16.63 | 24.50 | 38.58 |
| | TFRP-1 | 17.23 | 22.11 | 33.19 |
| | TFRP-2 | 16.88 | 21.85 | 33.09 |
| | DBA-1 | 20.70 | 26.00 | 35.39 |
| | DBA-2 | 16.15 | 25.45 | 34.81 |
| | IMHM | 16.93 | 22.19 | 30.78 |
| EIA | **ONA** | **0.23** | **0.41** | **1.02** |
| | MDAV | 0.48 | 1.67 | 3.84 |
| | VMDAV | 0.53 | 1.30 | 2.88 |
| | MD-MHM | 0.44 | 1.26 | 3.64 |
| | MDAV-MHM | 0.41 | 1.26 | 3.77 |
| | NPN-MHM | 0.55 | 0.96 | 2.32 |
| | $\mu$-Approx | 0.43 | 0.83 | 2.26 |
| | TFRP-1 | 0.53 | 1.65 | 3.24 |
| | TFRP-2 | 0.42 | 0.91 | 2.59 |
| | DBA-1 | 1.09 | 1.89 | 4.26 |
| | DBA-2 | 0.42 | 0.82 | 2.08 |
| | IMHM | 0.37 | 0.76 | 2.18 |

The algorithm has been implemented in Java and the experiments have been run on a AMD Ryzen 1700X machine under Ubuntu 17.04 x64. Table 2 shows the runtimes of ONA for the various test data sets and cluster sizes. To compute these runtimes, we have used the strictest convergence criterion: we keep iterating until no more record reassignments take place. We should remark that the steepest $SSE$ decrease takes place during the first few iterations. Thus, a less strict convergence condition could offer significantly shorter runtimes without a substantial difference in the SSE. Indeed, we have observed that the SSE reaches a stationary value long before the number of reassignments reaches 0.

**Table 2.** ONA runtimes in seconds for the test data sets and the tested cluster sizes.

| Time (s) | $k = 3$ | $k = 5$ | $k = 10$ |
|---|---|---|---|
| Census | 0.295 | 0.376 | 0.196 |
| Tarragona | 0.254 | 0.485 | 0.212 |
| EIA | 1.751 | 1.430 | 1.607 |

## 6 Conclusions and Future Research

We have proposed ONA, a novel microaggregation algorithm that significantly reduces the information loss with respect to existent algorithms. ONA operates iteratively and is based on Lloyd's clustering algorithm. Each iteration of ONA decreases the information loss until it converges to a (possibly local) minimum.

In the design of ONA, we have tried to match the two necessary conditions for optimal microaggregation as closely as possible. First, we make sure that each cluster contains only adjacent records. This is achieved by reassigning records to the cluster with the closest centroid. Second, we make sure that the size of clusters ranges between $k$ and $2k - 1$. In record reassignments, we take care that a source cluster is never left with less than $k$ records (otherwise we disband it) and that a destination cluster never increases to more than $2k - 1$ records (otherwise we split it into two clusters).

In the experimental section, we have presented an exhaustive comparison of the information loss with existent microaggregation algorithms. The results show that ONA offers a very significant reduction of the information loss. It is also important to remark that such a reduction is effected without resorting to complex procedures. Indeed, the internal operation of ONA is simpler than that of most of the microaggregation algorithms included in the comparison.

As future work, we plan to conduct a detailed analysis of the convergence conditions for ONA and also to extend it to categorical data. Currently, the range of microaggregation algorithms available for dealing with this kind of data is rather limited. The work in [5] provides a good starting point.

**Acknowledgments and disclaimer.** The following funding sources are gratefully acknowledged: European Commission (project H2020-700540 "CANVAS"), Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2017 SGR 705) and Spanish Government (project RTI2018-095094-B-C21 “Consent”). The views in this paper are the authors’ own and do not necessarily reflect the views of UNESCO or any of the funders.

## References

1. Brand, R., Domingo-Ferrer, J., Mateo-Sanz, J.M.: Reference data sets to test and compare SDC methods for protection of numerical microdata. European Project IST-2000-25069 CASC (2002). http://neon.vb.cbs.nl/casc/CASCtestsets.htm
2. Chang, C.C., Li, Y.C., Huang, W.H.: TFRP: an efficient microaggregation algorithm for statistical disclosure control. J. Syst. Softw. **80**(11), 1866–1878 (2007)
3. Domingo-Ferrer, J., Martínez-Ballesté, A., Mateo-Sanz, J.M., Sebé, F.: Efficient multivariate data-oriented microaggregation. VLDB J. **15**(4), 355–369 (2006)
4. Domingo-Ferrer, J., Mateo-Sanz, J.M.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Trans. Knowl. Data Eng. **14**, 189–201 (2002)
5. Domingo-Ferrer, J., Sánchez, D., Rufian-Torrell, G.: Anonymization of nominal data based on semantic marginality. Inf. Sci. (Ny) **242**, 35–48 (2013)
6. Domingo-Ferrer, J., Sebé, F., Solanas, A.: A polynomial-time approximation to optimal multivariate microaggregation. Comput. Math. Appl. **55**, 714–732 (2008)
7. Domingo-Ferrer, J., Soria-Comas, J.: Steered microaggregation: a unified primitive for anonymization of data sets and data streams. In: IEEE International Conference on Data Mining Workshops, ICDMW, pp. 995–1002. New Orleans (2017)
8. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Min. Knowl. Discov. **11**, 195–212 (2005)
9. Hansen, S.L., Mukherjee, S.: A polynomial algorithm for optimal univariate microaggregation. IEEE Trans. Knowl. Data Eng. **15**(4), 1043–1044 (2003)
10. Laszlo, M., Mukherjee, S.: Minimum spanning tree partitioning algorithm for microaggregation. IEEE Trans. Knowl. Data Eng. **17**(7), 902–911 (2005)
11. Lin, J.L., Wen, T.H., Hsieh, J.C., Chang, P.C.: Density-based microaggregation for statistical disclosure control. Expert Syst. Appl. **37**(4), 3256–3263 (2010)
12. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
13. Mortazavi, R., Jalili, S., Gohargazi, H.: Multivariate microaggregation by iterative optimization. Appl. Intell. **39**, 529–544 (2013)
14. Oganian, A., Domingo-Ferrer, J.: On the complexity of optimal microaggregation for statistical disclosure control. Stat. J. UN Econ. Comm. Eur. **18**, 345–354 (2001)
15. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International (1998)
16. Solanas, A., Martínez-Ballesté, A.: V-MDAV: a multivariate microaggregation with variable group size. In: Proceedings in Computational Statistics, pp. 917–926 (2006)
17. Soria-Comas, J., Domingo-Ferrer, J.: Differentially private data publishing via optimal univariate microaggregation and record perturbation. Knowl.-Based Syst. **153**, 78–90 (2018)
18. Soria-Comas, J., Domingo-Ferrer, J., Sánchez, D., Martínez, S.: Enhancing data utility in differential privacy via microaggregation-based k-anonymity. VLDB J. **23**, 771–794 (2014)
19. Soria-Comas, J., Domingo-Ferrer, J., Sánchez, D., Martínez, S.: t-Closeness through microaggregation: strict privacy with enhanced utility preservation. In: 2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016. pp. 1464–1465 (2016)