Differentially Private Data Publishing via Optimal Univariate Microaggregation and Record Perturbation

Jordi Soria-Comas, Josep Domingo-Ferrer

Universitat Rovira i Virgili, Department of Computer Science and Mathematics, UNESCO Chair in Data Privacy, CYBERCAT-Center for Cybersecurity Research of Catalonia, Av. Països Catalans 26, 43007 Tarragona, Catalonia

Abstract

We present an approach to generate differentially private data sets that consists in adding noise to a microaggregated version of the original data set. While this idea has already been pursued in the literature to reduce the sensitivity of attributes and hence the noise required to reach differential privacy, the novelty of our approach is that we focus on the microaggregated data set as our protection target (rather than aiming at protecting the original data set and viewing the microaggregated data set as a mere intermediate step). Interestingly, by starting from the microaggregated data set rather than the original data set, we achieve differential privacy for the individuals having contributed the original records while preserving substantially more utility. Compared with previous contributions using microaggregation as a prior step to reach differential privacy, the utility improvement comes from avoiding the need to use insensitive microaggregation. This claim is supported by theoretical and empirical utility comparisons between our approach and existing approaches. We analyze several microaggregation strategies: multivariate MDAV, individual-ranking MDAV, and optimal microaggregation. In particular, we reformulate optimal microaggregation to fit it to the generation of differentially private data sets.

Keywords: Differential privacy, Microaggregation, Anonymization, Statistical disclosure control, Privacy

1. Introduction

Microdata (that is, information at the individual level) are usually the most convenient type of data for secondary use. However, the risk of disclosure inherent to releasing such detailed information is significant. Traditionally, data were mostly handled by a reduced number of controllers (e.g. national statistical

Email address: {jordi.soria, josep.domingo}@urv.cat (Jordi Soria-Comas, Josep Domingo-Ferrer)

offices), who had collected them under strong pledges of privacy. In that scenario, reasonable assumptions about the knowledge available to intruders could be made and the methodology for disclosure risk limitation could be adjusted accordingly. Nowadays, the developments in information technology greatly facilitate the collection of all kinds of personal data by a variety of controllers. This bounty of information increasingly complicates making well-grounded assumptions about the background knowledge available to potential intruders [24].

Differential privacy [10] (DP) is a well-known privacy model that gives privacy guarantees without assuming anything on the intruder's background knowledge. In this sense, DP is well adapted to the current scenario with many data controllers. Unlike privacy models designed to protect sets of microdata (e.g. [20, 15, 13]), DP was originally introduced to limit the disclosure risk incurred when returning answers to queries on a database. However, DP mechanisms to generate entire data sets were proposed soon after the inception of DP [14]; we will use the term DP data set to denote a data set output by a DP mechanism.

The dominant approach to generate DP microdata is based on the computation of DP histograms. However, histogram-based approaches have severe limitations when the number of attributes grows: for fixed attribute granularities, the number of histogram bins grows exponentially with the number of attributes, which has a severe impact on both computational cost and accuracy. Alternatively, the DP data set can be generated using a record perturbation approach. The simplest way to do this is to collect DP answers to a set of queries that ask for each individual record in the original data set. However, the amount of noise required to attain DP for such queries is too large for the DP data set to stay useful. In [25], microaggregation is used as an intermediate step to reduce the sensitivity of the queries: rather than asking for the record of each individual, we arrange the records into clusters and query for a representative of each cluster (for example, the centroid record). The sensitivity of the latter queries is smaller because they depend on several individuals (those in the cluster). Related approaches were followed in [26, 21, 22]. However, these contributions are not without limitations: [25] and [26] need special microaggregation algorithms that are less utility-preserving than the standard ones, whereas [21] and [22] can only deal with data sets containing a single attribute.

1.1. Contribution and plan of this paper

In this work we present a novel record-level perturbation-based methodology to generate DP data sets. Unlike existing perturbation-based approaches, we can use standard microaggregation algorithms and deal with multiple attributes, which leads to a significant improvement in the utility of the resulting DP data set. Our approach can work with any microaggregation algorithm, but we will choose a few well-known algorithms for the sake of evaluation. The use of standard microaggregation algorithms becomes possible because we switch the focus of DP from the original data set to the microaggregated data set. To make this change compatible with DP, we extend the definition of DP to data sets in which the usual assumption of a one-to-one mapping between records and

individuals does not hold. This extension makes sense because the aim of DP is to protect individuals (rather than records).

In Section 2 we briefly introduce basic concepts about DP and recall the state of the art in DP data set generation via record masking. In Section 3 we extend the notion of DP to data sets where there is no one-to-one mapping between records and individuals. In Section 4 we describe our approach to generating DP data sets. In Section 5 we propose several microaggregation methods based on the MDAV heuristic for generating DP data sets. In Section 6 we introduce a new optimal microaggregation method for DP data set generation. In Section 7 we present theoretical analyses on the utility of the generated DP data sets. In Section 8 we experimentally evaluate the proposed microaggregation methods by comparing them with each other and with previous approaches. Finally, in Section 9 we summarize the conclusions and outline future research avenues.

2. Preliminaries

2.1. Background on differential privacy

Differential privacy [10] is popular among academics due to the strong privacy guarantees it offers. DP does not rely on assumptions about the background knowledge available to the intruders. Rather, disclosure risk limitation is tackled in a relative manner: the result of any analysis should be similar between data sets that differ in one record. As stated in [9], such a guarantee should encourage individuals to participate in a data set because the disclosure risk they incur when contributing their data is limited:

> Any given disclosure will be, within a multiplicative factor, just as likely whether or not the individual participates in the database. As a consequence, there is a nominally higher risk to the individual in participating, and only nominal gain to be had by concealing or misrepresenting one's data.

Differential privacy assumes the presence of a trusted party that: (i) holds the data set, (ii) receives the queries submitted by the data users, and (iii) responds to them in a privacy-aware manner. The notion of differential privacy is formalized according to the following definition:

Definition 1 (ϵ-differential privacy)**.** *A randomized function κ gives ϵ-differential privacy (ϵ-DP) if, for all data sets D_1 and D_2 that differ in one record (a.k.a. neighbor data sets), and all $S \subset Range(\kappa)$, one has*

$$\Pr(\kappa(D_1) \in S) \leq \exp(\epsilon) \Pr(\kappa(D_2) \in S). \tag{1}$$

Given a query function f, the goal in differential privacy is to find a randomized function κ_f that satisfies ϵ-DP and approximates f as closely as possible. For the case of numerical queries, κ_f can be obtained via noise addition; that is, $\kappa_f(\cdot) = f(\cdot) + N$, where N is a random noise that has been properly adjusted to attain ϵ-DP. Adding Laplace-distributed noise whose scale has been adjusted

to the global sensitivity of the query f is probably the most common method to derive κ_f (although other methods have been proposed [23, 16, 17]).

Definition 2 (L_1-sensitivity)**.** *The L_1-sensitivity, Δ_f, of a function $f : \mathcal{D}^n \to \mathbb{R}^d$ is the maximum variation of f between data sets that differ in one record:*

$$\Delta_f = \max_{d(D,D')=1} \|f(D) - f(D')\|_1 .$$

Proposition 1. *Let $f : \mathcal{D}^n \to \mathbb{R}^d$ be a query function. The mechanism $\kappa_f(D) = f(D) + (N_1, \ldots, N_d)$, where N_i are drawn i.i.d. from a Laplace$(0, \Delta_f/\epsilon)$ distribution, is ϵ-DP.*

2.2. State of the art

In [25, 26, 21, 22], DP data sets are generated via record masking. In these works, microaggregation is employed to reduce the sensitivity of the queries used to generate the DP data sets: rather than querying for each original record, representatives of the microaggregation clusters are queried. Since a cluster representative is an aggregation of the records in the cluster, it is less sensitive to changes than any single record. The amount of sensitivity reduction depends on how such representative values are computed.

Specifically, in [25, 26], multivariate microaggregation is used to partition the original data set into clusters of k or more records. The DP data set is derived by querying the centroid (arithmetic average) of each cluster. Since multivariate microaggregation with minimal cluster size k over all the attributes ensures k-anonymity, one can regard this approach as combining k-anonymity and DP. However, a special type of microaggregation called *insensitive microaggregation* is required. The main reason is that, in standard microaggregation algorithms, changing one value in one record may yield a completely unrelated set of clusters, which would not reduce sensitivity; although it is certainly unlikely that changing one record value changes all clusters, to guarantee DP one needs to consider the worst case. In insensitive microaggregation, one requires changes in a single record to produce a set of clusters that differ (at most) in one record. In this way, the sensitivity of a centroid divides the sensitivity of the original records by the corresponding cluster size. The downside of insensitive microaggregation is that it yields worse within-cluster homogeneity than standard microaggregation and, hence, higher information loss. Furthermore, the minimum cluster size k grows with the data set size, as we show next. Let n be the number of records in a data set. To obtain a DP version of it, n/k centroids are released, each one computed on a cluster of cardinality k and having sensitivity Δ/k, where Δ is the sensitivity of the original records (strictly speaking, Δ is the sensitivity of a query Q_i that returns the i-th original record, for $i = 1$ to the number of records). Hence, the sensitivity of the whole data set to be released is $n/k \times \Delta/k$. Thus, for numerical data sets, Laplace noise with scale parameter $(n/k \times \Delta/k)/\epsilon$ must be added to each centroid to obtain an ϵ-DP output. For this approach to reduce the amount of noise required to attain ϵ-DP, the inequality $n/k \times \Delta/k \leqslant \Delta$ must hold. Equivalently, one needs $k \geqslant \sqrt{n}$.

To circumvent the previous problems of [25, 26], an alternative approach was proposed in [21, 22] based on individual-ranking (IR) microaggregation. Since IR microaggregates only one attribute at a time, it achieves insensitivity while avoiding the information loss penalty of multivariate insensitive microaggregation. On the other hand, there is no lower bound on k: when using IR microaggregation, we can bound the sensitivity of the set of centroids by Δ/k (to be compared with $n/k \times \Delta/k$ in the case multivariate insensitive microaggregation); hence, no matter the value of k, the sensitivity is reduced.

The seeming superiority of the IR microaggregation approach is, however, severely limited by the fact that it can only deal with data sets that contain a single attribute; combining the DP versions of several attributes via sequential composition does not yield a DP data set.

3. Extending DP to protect individuals rather than records

DP implicitly assumes a one-to-one mapping between records in the data set and individuals to whom the records correspond: each record corresponds to a different individual and each individual is represented by only one record. Under this assumption, protecting the privacy of individuals (the goal of DP) can be achieved by ensuring the indistinguishability of the responses to queries computed on data sets that differ in one record (see Definition 1). However, as noted in [12], even in the one-to-one mapping case, DP does not offer the expected protection (to hide any evidence about any single individual) *if there are dependences between individuals*. For individuals that are not independent, information about one individual can be deduced from the set of dependent records. Unlike [12], we make the mainstream assumption that individuals are independent, but *we do not require a one-to-one mapping between records and individuals*. In this respect, our approach is more similar to [2], but we avoid their strong assumptions, such as computing the sensitivity for each particular data set by measuring the variability in a query as the result of removing one individual. Indeed, this sensitivity computation offers significantly less protection than the standard global sensitivity; in particular, it may not provide indistinguishability with respect to individuals that are not in the data set.

We assume that there may be several records that correspond to a single individual, there may be a single record that corresponds to a group of individuals, or, more generally, there may be several records that correspond to a group of individuals. In these cases, DP as formalized in Definition 1 is not applicable. We want to extend DP to the previous cases while keeping its intuition: the presence or absence of an individual should not alter the outcome of queries significantly. Before proposing such an extension of DP, let us present two illustrative examples.

Example 1. Let D be a data set such that each record in D corresponds to a different individual, and let f be a query to be computed on D. If the data set D is protected by a DP access mechanism, when we query for $f(D)$ we get an

ϵ-DP response $\kappa_f(D)$. To be specific, assume that the Laplace noise addition mechanism is used: $\kappa_f(D) = f(D) + Lap(0, \Delta_f/\epsilon)$.

Now, let D_k be a data set that contains k repetitions of each of the records in D. We will show that protecting D_k by DP as stated in Definition 1 fails to deliver the expected privacy protection to individuals. Let us consider the query $f' = (f_1 + \ldots + f_k)/k$, where f_i is the query f restricted to the i-th occurrence of D in D_k. We have $f'(D_k) = f(D)$; that is, both queries yield the same output. Since $f_1, \ldots, f_k$ are computed on disjoint sets of records, the L_1-sensitivity of f' is Δ_f/k (a change in one record may alter the result of only one of the f_i by Δ_f). The Laplace noise addition mechanism for f' is $\kappa_{f'}(D_r) = f'(D_r) + Lap(0, \Delta_f/(k\epsilon))$.

It becomes clear from the amount of noise added in $\kappa_f(D)$ and $\kappa_{f'}(D_k)$ that $\kappa_{f'}(D_k)$ fails to deliver the expected protection to individuals. In fact, the ϵ-DP mechanism for f' is equivalent to a $k\epsilon$-DP mechanism for f.

To avoid underprotection when querying D_k, what is relevant is that a change *in one individual* results in changes in k *records* of D_k. Thus, the L_1-sensitivity of f' to a change in one individual is Δ_f rather than Δ_f/k.

Example 2. Let D be a data set such that each record in D corresponds to a different individual, and let $\{C_1, \ldots, C_s\}$ be a partition of D into clusters of records. Suppose that we are interested in computing the average record of cluster C_1, say $mean_{C_1}$.

If Δ_r is the maximum possible change in a record, the L_1-sensitivity of $mean_{C_1}$ is $\Delta_r/|C_1|$, where $|C_1|$ is the cardinality of C_1. Thus, the Laplace mechanism is $\kappa_{mean_{C_1}}(D) = mean_{C_1}(D) + Lap(0, \Delta_r/(\epsilon|C_1|))$.

Let us assume now that, rather than D, we are given the microaggregated data set $\bar{D} = \{r_{mean_{C_1}(D)}, \ldots, r_{mean_{C_s}(D)}\}$, where each cluster of records has been replaced by the average record of the cluster. The domain of the original and the microaggregated data sets is the same: for any record r in the domain of the original data set, a cluster formed by repetitions of r may exist, and in this case the average of the cluster is record r. To obtain the average record of cluster C_1 in the microaggregated data set, we simply need to query for record $r_{mean_{C_1}(D)}$. If we use the Laplace mechanism to achieve DP as per Definition 1, we need to add a noise proportional to the maximum possible change in a record, which remains Δ_r. Thus, the Laplace mechanism is $\kappa_{r_{mean_{C_1}(D)}}(\bar{D}) = r_{mean_{C_1}(D)} + Laplace(0, \Delta_r/\epsilon)$.

By observing the amount of noise added in each case (original and microaggregated data set), we realize that, if we apply DP as per Definition 1 to the microaggregated data set, we are overprotecting the individuals.

To avoid the overprotection that results from $\kappa_{r_{mean_{C_1}(D)}}(\bar{D})$, we should notice that a change *in one individual* can only alter $r_{mean_{C_1}(D)}$ by $\Delta_r/|C_1|$.

From the previous examples, it becomes clear that, when there is no one-to-one mapping between individuals and records, the sensitivity must be computed with respect to data sets that differ *in one individual*, rather than one record, in order to avoid over and underprotection.

Definition 3. *A mechanism κ is ϵ-DP if, for all $S \subset Range(\kappa)$ and for all data sets D and D' that differ* in one individual*, one has*

$$\Pr(\kappa(D) \in S) \leq \exp(\epsilon) \Pr(\kappa(D') \in S).$$

The difference between Definitions 1 and 3 is the kind of neighborhood relationship of data sets D and D': whereas in the former definition neighbor data sets differ in one record, in the latter definition they differ in one individual.

Adapting the Laplace mechanism to our definition is simple. We only need to compute the sensitivity as the maximum change that the query answer can undergo between data sets that differ in one individual.

Definition 4. *The L_1-sensitivity of a query function f is*

$$\Delta_f = \max\{\|f(D) - f(D')\|_1 : D, D' \textit{differ in one individual}\}$$

The following proposition can be proven in the same way Proposition 1 was proven in [9].

Proposition 2. *When using DP as per Definition 3 and the L_1-sensitivity as per Definition 4, the mechanism $\kappa_f = f + Lap(0, \Delta_f/\epsilon)$ is ϵ-DP.*

4. DP data sets via microaggregation

Let D be the original data set. Assume that we want to generate D^ϵ —an anonymized version of D— that satisfies ϵ-DP. Let $I_r(D)$ be the query that returns r. We can think of the data set D as the collected answers to the queries $I_r(D)$ for $r \in D$, and we can generate D^ϵ by collecting ϵ-DP responses to $I_r(D)$ for $r \in D$. Such a naive procedure to generate a DP data set is, however, likely to produce a large information loss. In the end, the purpose of DP is to make sure that individual records do not have any significant effect on query responses, which implies that the accuracy of the responses to $I_r(D)$ is necessarily low.

The accuracy shortcoming described in the previous paragraph is inherent to DP making the presence or absence of individuals unnoticeable. To put it otherwise, if a DP mechanism must make individuals unnoticeable, it cannot provide accurate responses to queries $I_r(D)$ on individual records, for $r \in D$. Even if ϵ-differential privacy is relaxed to (ϵ, δ)-differential privacy (by adding δ to the right-hand side of Inequality (1)), we can show that the accuracy shortcoming subsists. Certainly, (ϵ, δ)-DP can be attained with mechanisms calibrated to the smooth sensitivity [17] rather than the global sensitivity. However, the smooth sensitivity is based on the local sensitivity and the local sensitivity of the list of queries $I_r(D)$ for $r \in D$ is equal to the maximum possible change in a record of D. In the best case (when all the records take values at the center of the data domain), the maximum change is $\Delta_D/2$, where Δ_D is the maximum distance between two records. Hence, neither calibrating to the smooth sensitivity nor calibrating to the local sensitivity is very helpful to improve the accuracy of the generated data.

To make perturbative masking viable for generating DP data sets, we have to reduce the sensitivity of the queries used. This requires a shift from individual queries to queries that ask for aggregate or statistical information. Along the lines of [25, 26, 21, 22], our proposal is based on microaggregation. In spite of microaggregation being itself a well-known technique in disclosure risk limitation, we use it here with the sole purpose of reducing the sensitivity of the queries. The disclosure risk limitation comes from the enforcement of DP. This change of purpose carries along a change in the traditional way of thinking about microaggregation.

In standard microaggregation, one splits the data set into clusters of at least k records and then replaces the records in each cluster by the cluster centroid, where the minimum value k prevents the cluster from being too representative of any individual in it. In our case, we are also interested in having not too small clusters (in order to limit the impact of individual contributions and hence the sensitivity), but *we can drop the requirement of a minimum cluster size*. In our case, the total error is the combination of the error introduced by microaggregation and the error due to noise addition; thus, if adding one more record to a cluster produces a large increase in the microaggregation error, it may be preferable to use the smaller cluster. In this work, we think of microaggregation as an algorithm that proceeds in the following two steps:

1. Split the data set into clusters of records;
2. Compute a representative record of each cluster and replace the records in the cluster by it.

To reduce the error introduced by microaggregation, we usually want to generate clusters that are as homogeneous as possible. For the sake of generality, in this section, we do not favor any particular strategy to generate the microaggregation clusters: they can all have the same cardinality or not, they can be optimal (maximally homogeneous) or not, be generated in a randomized or deterministic way, etc. However, to be able to analyze the effect of microaggregation on the sensitivity, we need to fix the particular way in which the records in a cluster are combined to generate a record that is representative of the cluster. In this work, we use the mean as aggregation operation (that is, we compute the centroid of the cluster).

The approach we propose differs from those of [25, 26, 21, 22] in that *here we consider that the data set to be protected is the microaggregated one*, rather than the original one. In other words, given an original data set D, we generate $\bar{D}$ by microaggregating the records in D. From this point on, we discard D and we focus on protecting $\bar{D}$. Hence, the goal is to publish $\bar{D}^\epsilon$, a DP version of $\bar{D}$. Counterintuitively, by starting from the microaggregated data set rather than the original data set, we manage to obtain a DP data set that *preserves substantially more utility* with respect to the original data. Of course, since in the microaggregated data set there is no one-to-one mapping between individuals and records, we need to use the extension of DP described in Section 3. In particular, Example 2 is close to our needs.

The data set $\bar{D}$ acts as a proxy for the original data set D. Thus, when evaluating the utility of $\bar{D}^\epsilon$ we need to account for two sources of error: (i) the error due to the microaggregation (that is, the error caused by using $\bar{D}$ as a proxy of D), and (ii) the noise introduced to attain ϵ-DP. The advantage of the proposed approach lies in the fact that the error introduced in the microaggregation step is likely to be more than compensated by the reduction in the noise required to attain DP (compared with the noise that would be required to attain DP directly from the original data set D).

Since the contribution of a record to the centroid is inversely proportional to the cardinality of the corresponding cluster, the centroid sensitivity can be obtained as the record sensitivity (the maximum change in a record) divided by the cluster cardinality. This is formalized in the following proposition.

Proposition 3. *Let $C \subset D$ be a cluster of records and let c be the mean of the records in C. Let Δ_D be the L_1-sensitivity of a record in D. The L_1-sensitivity of the centroid c is $\Delta_c = \Delta_D/|C|$.*

Proof. Δ_c represents the maximum change in c due to an arbitrary change in a single record. Since the maximum change in a single record is Δ_D and each record contributes to c, at most, in a proportion of $1/|C|$, the maximum change in c is $\Delta_D/|C|$. □

Notice that the sensitivities may differ for centroids of different clusters, because the sensitivity depends on the cluster cardinality. Once the sensitivity of a centroid c is computed, ϵ-DP can be attained by adding a Laplace noise with zero mean and scale Δ_c/ϵ. Since each cluster contains disjoint records, parallel composition applies; thus, by adding Laplace noise independently to each cluster, we obtain the list of ϵ-DP centroids (see Figure 1).

Since each record is replaced by the corresponding centroid, each centroid is repeated as many times as there are records in the corresponding cluster. We now explain why in Figure 1 all repetitions of a centroid value are added exactly the same noise. If we added a different random noise to each repetition of the centroid, we would have $|C|$ non-independent DP outcomes each of which has sensitivity $\Delta_D/|C|$; hence, by sequential composition, the sensitivity of the list of centroid repetitions in the cluster would be Δ_D, which would cancel the benefits of microaggregation. To keep the sensitivity of the centroid repetitions at $\Delta_D/|C|$, we must have a single DP centroid value, that is, we must add exactly the same noise to all the repetitions of a given centroid. In other words, for each cluster C_i, we take a single draw, n_i, from the $Lap(0, \Delta_D/(|C_i|\epsilon))$ distribution and use it to mask the $|C_i|$ occurrences of c_i.

The procedure to generate an ϵ-DP data set based on record-level microaggregation is formally described in Algorithm 1. The algorithm takes as input parameters the microaggregated data set $\bar{D}$ (whose records consist in the corresponding cluster centroids), the mapping between records in $\bar{D}$ and clusters, and the desired level ϵ of DP. Next, we fix the noise n_i that will be added to all records mapped to each cluster C_i. Finally, we loop through the records in $\bar{D}$

	$\bar{D}$		$\bar{D}^\epsilon$	
C_1	c_1	$\longrightarrow$	$c_1 + n_1$	$n_1 = Lap(0, \frac{\Delta_D}{\lvert C_1 \rvert \epsilon})$
	$\dots$	$\dots$	$\dots$	
	c_1	$\longrightarrow$	$c_1 + n_1$	
C_2	c_2	$\longrightarrow$	$c_2 + n_2$	$n_2 = Lap(0, \frac{\Delta_D}{\lvert C_2 \rvert \epsilon})$
	$\dots$	$\dots$	$\dots$	
	c_2	$\longrightarrow$	$c_2 + n_2$	
	$\vdots$	$\vdots$	$\vdots$	
C_l	c_l	$\longrightarrow$	$c_l + n_l$	$n_l = Lap(0, \frac{\Delta_D}{\lvert C_l \rvert \epsilon})$
	$\dots$	$\dots$	$\dots$	
	c_l	$\longrightarrow$	$c_l + n_l$	

Figure 1: Generation of an ϵ-DP data set using record-level microaggregation to reduce the amount of noise required

and add to each record the noise that corresponds to the cluster it is mapped to.

The procedure depicted in Figure 1 assumes that microaggregation is performed over whole records (either because the data set contains a single attribute or because multivariate microaggregation over all the attributes is used). In the remainder of this section, we generalize the previous procedure to work independently with several individual attributes or subsets of attributes. Essentially, we split the attributes into disjoint subsets, apply the previous procedure independently to each subset, and use sequential composition to determine the overall level of DP.

Let us assume that the microaggregation has been performed independently over the disjoint subsets of attributes $AS_1, \dots, AS_m$. According to sequential composition, the level of differential privacy from several independent queries accumulates to determine the overall level of DP. As we aim to work independently with each of the subsets AS_i, following sequential composition we need to split the overall privacy budget, ϵ, among the previous subsets. That is, we fix values $\epsilon_1, \dots, \epsilon_m$ subject to the constraints $\epsilon_i \geq 0$ and $\epsilon_1 + \dots + \epsilon_m = \epsilon$. For each subset AS_i, we apply the procedure in Algorithm 1 to attain ϵ_i-DP. Sequential composition tells that the result is ϵ-DP. This is illustrated in Figure 2 and formalized in Algorithm 2.

We have presented two algorithms to generate DP data sets based on microaggregation and record perturbation: in Algorithm 1 microaggregation is performed over entire records, while in Algorithm 2 it is performed independently for several groups of disjoint attributes. Which of both algorithms yields more accurate results depends on two factors: the error introduced by microaggregation and the amount of noise required to attain DP. Let us discuss both factors in the next paragraphs.

From the statistical disclosure control literature, we know that the utility of microaggregation degrades quickly with the dimensionality (number of at-

Algorithm 1 Procedure to generate an ϵ-DP data set using record-level microaggregation to reduce the amount of noise required

Require:

$\bar{D} = \{r_1, \ldots, r_L\}$: microaggregated data set (each record r_j is the corresponding cluster centroid)

Mapping τ between records of $\bar{D}$ and the clusters $C_1, \ldots, C_l$ formed in the microaggregation

ϵ: desired level of DP

Output

$\bar{D}^\epsilon$: an ϵ-DP data set

for $i \in \{1, \ldots, l\}$ **do**

set n_i= random draw from the $Lap(0, \frac{\Delta_D}{|C_i|\epsilon})$ distribution

end for

for $j \in \{1, \ldots, L\}$ **do**

let $C_i := \tau(r_j)$

set $r_j^\epsilon = r_j + n_i$

end for

return $\bar{D}^\epsilon = \{r_1^\epsilon, \ldots, r_L^\epsilon\}$

Algorithm 2 Procedure to generate an ϵ-DP data set by independently microaggregating the groups of attributes $AS_1, \ldots, AS_n$ and reaching ϵ_i-DP for group AS_i

Require:

$AS_1, \ldots, AS_m$: list of disjoint subsets of attributes

$\bar{D}$: microaggregated data set, where microaggregation has been independently computed for the projections on each subset of attributes (each record has been replaced by the centroids of the clusters that contain it in each projection)

$(\tau_1, \ldots, \tau_m)$: τ_i is the mapping between records in $\bar{D}$ and the clusters $C_1^i, \ldots, C_{l_i}^i$ computed for the projection $\bar{D}[AS_i]$ of $\bar{D}$ on attribute subset AS_i

$\epsilon_1, \ldots, \epsilon_m$: ϵ_i is the level of DP for attributes AS_i (subject to $\sum \epsilon_i = \epsilon$)

Output

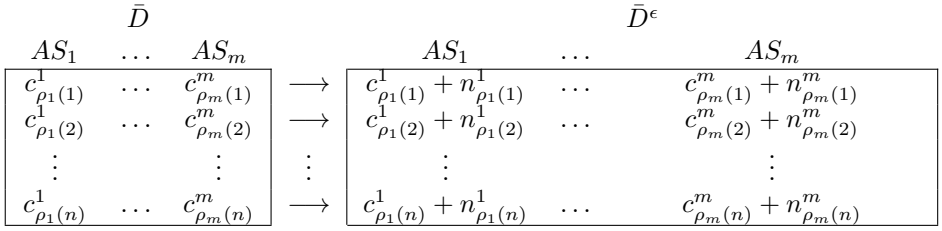
$\bar{D}^\epsilon$: an ϵ-DP data set

for $i \in \{1, \ldots, m\}$ **do**

$\bar{D}^\epsilon[AS_i]$ = Algorithm 1$(\bar{D}[AS_i], \tau_i, \epsilon_i)$

end for

return $\bar{D}^\epsilon$



where $\rho_i(r)$ =cluster number associated to record r for attribute group AS_i
$n^i_j = Lap(0, \frac{\Delta_{AS_i}}{|C^i_j|\epsilon_i})$

Figure 2: Generation of an ϵ-DP data set by independently microaggregating the subsets of attributes $AS_1, \ldots, AS_m$ and reaching ϵ_i-DP for group AS_i

tributes microaggregated together). This is why individual-ranking microaggregation (independent microaggregation of each attribute) is much more utility-preserving than multivariate microaggregation. The downside of separately microaggregating subsets of attributes is an increase of the disclosure risk. In the extreme case, we have individual-ranking microaggregation, that is known to offer very little protection [7, 4]. However, since we are not using microaggregation to obtain protection (we rely on DP for that), but rather as a way to reduce sensitivity and hence the noise required by DP, separately microaggregating subsets of attributes seems the best option.

The effect of the dimensionality on the amount of noise required to attain DP is not so obvious. When microaggregating entire records, the sensitivity of the centroids is greater but we can use the entire privacy budget ϵ to mask them. When independently microaggregating several groups of attributes, the sensitivity of the centroids decreases but we have to split the privacy budget among the groups of attributes (each attribute group AS_i is given ϵ_i-DP, in such a way that $\sum \epsilon_i = \epsilon$). The following proposition states that, if the microaggregation clusters have constant size k, we can make the amount of noise added when independently microaggregating groups of attributes equivalent to the noise added when microaggregating entire records. Since we can keep constant the amount of error introduced to attain DP, the conclusion is that we should increase as much as possible the granularity of microaggregation (because doing so decreases the error due to microaggregation). Therefore, *univariate microaggregation (that is, individual-ranking microaggregation) is the best option.*

Proposition 4. *Let M be a microaggregation algorithm that generates clusters with fixed size k. Let ϵ be the target DP level. By selecting appropriate values for ϵ_i, the amount of noise added to each attribute by Algorithm 2 can be made equivalent to the noise added by Algorithm 1.*

Proof. According to Algorithm 1, to attain ϵ-DP, we need to add to each attribute a noise drawn from $Lap(0, \Delta_D/k\epsilon)$. According to Algorithm 2, attribute group AS_i, having sensitivity Δ_i , is added noise drawn from $Lap(0, \Delta_i/k\epsilon_i)$.

Both Laplace distributions are equal when $\Delta_D/\epsilon = \Delta_i/\epsilon_i$, which may be enforced by taking $\epsilon_i = \epsilon\Delta_i/\Delta_D$. Since $\Delta_D = \sum \Delta_i$, the sum of the ϵ_i amounts to ϵ (as required by Algorithm 2). □

5. MDAV microaggregation methods for DP data set generation

In the previous section, we have described two algorithms to generate DP data sets based on microaggregation and noise addition. Algorithm 1 microaggregates entire records, while Algorithm 2 independently microaggregates over disjoint subgroups of attributes. However, so far we have not focused on any particular method to perform the microaggregation itself. In this section, we propose several methods based on MDAV microaggregation [8], which is a well-known microaggregation heuristic. Moreover, MDAV microaggregation has previously been used to improve the accuracy of DP data sets generated via record perturbation [26, 22]. Thus, instantiating our DP data set generation algorithms with MDAV microaggregation seems natural. Specifically, we consider the following approaches to generate DP data sets:

- MDAV_DP. Based on Algorithm 1 and using MDAV microaggregation to microaggregate entire records.
- MDAV_IR_DP. Based on Algorithm 2 and using MDAV microaggregation on each attribute independently.

According to Proposition 4, MDAV_IR_DP should be preferred to MDAV_DP. The reason to still consider MDAV_DP is that multivariate MDAV has previously been used to generate DP data sets, and hence it is a natural benchmark.

Normalizing the values of different attributes is important in microaggregation, in order to prevent some attributes from having more weight than others due to scale differences. Without normalization, attributes with larger scales would have more influence in cluster formation, which would result in substantial damage to the correlations among attributes with smaller scales. Of course, if the data set is normalized before microaggregating it, the microaggregated data set should be de-normalized to recover the original scales before the subsequent DP noise addition step.

6. Optimal univariate microaggregation for DP

We have argued that, being univariate, MDAV_IR_DP is better than the multivariate MDAV_DP. However, we can still do better than MDAV_IR_DP. In this section we describe the use of optimal univariate microaggregation to generate DP data sets according to Algorithm 2.

In microaggregation, optimality is measured in terms of the information loss produced by the replacement of the records within a cluster by the cluster centroid. We will use SSE (sum of squared errors) to measure the information loss, which is the usual information loss measure in microaggregation. SSE is

the sum of the squared errors (distances) between each record and its microaggregated version, that is, its corresponding centroid. The SSE of a partition P of a data set D into clusters of records is computed as

$$SSE_P = \sum_{x \in D} (x - c(C_x))^2,$$

where C_x is the cluster of P that contains x, and $c(C_x)$ is the centroid of C_x.

In our case, the goal is not to minimize the SSE due to the microaggregation step, but to minimize the total SSE that results from both microaggregation and noise addition. Since the noise addition step introduces randomness, we seek to minimize the expected SSE, that is,

$$SSE_P^\epsilon = E\left(\sum_{x \in D} (x - c_\epsilon(C_x))^2\right) = \sum_{x \in D} E\left((x - c_\epsilon(C_x))^2\right),$$

where C_x is the cluster of P that contains x, and $c_\epsilon(C_x)$ is an ϵ-DP version of the centroid of C_x.

The following proposition shows that the contribution of one record to SSE_P^ϵ can be decomposed as the contribution to SSE_P plus the variance of the DP centroid.

Proposition 5. *Let $x \in D$ be a record, C_x be the microaggregation cluster in partition P associated to x, $c(C_x)$ be the centroid of C_x, and $c_\epsilon(C_x)$ be an ϵ-DP version of c. The contribution of x to SSE_P^ϵ is the contribution of x to SSE_P plus the variance of the DP centroid $c_\epsilon(C_x)$.*

Proof. The contribution of x to SSE_P^ϵ is $SSE_{C_x,x}^\epsilon = E\left((x - c_\epsilon(C_x))^2\right)$. By adding and subtracting $c(C_x)$, we can rewrite this contribution as

$$E\left((x - c(C_x) + c(C_x) - c_\epsilon(C_x))^2\right),$$

which can be further expressed as:

$$E\left((x - c(C_x))^2 + (c(C_x) - c_\epsilon(C_x))^2 + 2(x - c(C_x))(c(C_x) - c_\epsilon(C_x))\right)$$

$$= E\left((x - c(C_x))^2\right) + E\left((c(C_x) - c_\epsilon(C_x))^2\right)$$

$$+2\left(E(xc(C_x)) - E(xc_\epsilon(C_x)) - E(c(C_x)c(C_x)) + E(c(C_x)c_\epsilon(C_x))\right).$$

In the last expression, the first addend $E\left((x - c(C_x))^2\right)$ is the contribution of x to SSE_P, the second addend $E\left((c(C_x) - c_\epsilon(C_x))^2\right)$ is the variance of $c_\epsilon(C_x)$, and the last addend is 0 (as its terms cancel each other). □

By using Proposition 5, we can express SSE_P^ϵ in terms of SSE_P and the variances of the DP centroids as

$$SSE_P^\epsilon = \sum_{C \in P} SSE_C^\epsilon = SSE_P + \sum_{C \in P} |C| Var(c_\epsilon(C)). \tag{2}$$

In general, microaggregation is an NP-hard problem [18]. However, an optimal algorithm that is quasi-linear in the number n of records has been proposed for the univariate case [11]. In the remainder of this section, we give a new optimal algorithm more suited to our scenario. Like [11], it is based on computing the shortest path in an appropriately defined graph. First, we construct a directed graph and then we show that finding the partition P_{opt} that minimizes SSE_P^ϵ can be attained by finding a shortest path in the graph.

The algorithm we present in what follows differs substantially from the one in [11]. On the one hand, since we do not restrict the cluster cardinality to multiples of k, our graph has many more edges ($O(n^2)$ edges vs $O(n)$ edges in [11]); on the other hand, the length of our edges is SSE plus the expected DP error (in [11] it was only SSE). With a quadratic number of edges, direct edge length calculation would yield $O(n^3)$ complexity; by crafting a suitable calculation procedure, we manage to keep the cost at $O(n^2)$.

6.1. Graph construction

Let $D = \{r_1, \ldots, r_n\}$ be the data set to be microaggregated, where r_i are either univariate records or the values of an attribute that is individually microaggregated. Let us assume that $r_1 \leq r_2 \leq \ldots \leq r_n$ (even non-numerical values can be ordered, e.g., by using the S-distance defined in [5]). The graph G must be constructed as follows:

- Add nodes labeled $1, 2, \ldots, n$ to G. These nodes correspond to the values $r_1, r_2, \ldots, r_n$.
- Add a node with label 0 to G.
- For each pair of nodes (i, j) with $i < j$, add a directed edge from i to j. The length of the edge (i, j) is computed as the contribution to SSE_P^ϵ of cluster $\{r_{i+1}, \ldots, r_j\}$.

Computing the length of the edges is the costliest step in the construction of the graph. This computation can be done independently for each edge. By proceeding in this way, the cost of computing the length of edge (i, j) is linear in $j - i$, and the overall cost is

$$\Theta\left(\sum_{0 \leq i < j \leq n} (j - i)\right) = \Theta\left(\frac{1}{6} n(1 + n)(2 + n)\right) = \Theta\left(n^3\right).$$

However, we can take advantage of the fact that the SSE of a cluster can be computed from the SSE of a subset of the cluster, as described in the following proposition.

Proposition 6. *Let $C(i, j) = \{r_{i+1}, \ldots, r_j\}$ be the cluster associated with edge (i, j) and let $c(i, j) = \frac{1}{j-i} \sum_{r \in C(i,j)} r$ be the average record (centroid) of $C(i, j)$. The SSE of $C(i, j + 1)$ can be computed in terms of the SSE of $C(i, j)$ as*

$$SSE_{C(i,j+1)} = SSE_{C(i,j)} + (j - i)(c(i, j) - c(i, j + 1))^2 + (r_{j+1} - c(i, j + 1))^2.$$

Proof. By definition, we have $SSE_{C(i,j+1)} = \sum_{r \in C(i,j+1)} (r - c(i, j+1))^2$, which can be rewritten as

$$SSE_{C(i,j+1)} = \sum_{r \in C(i,j)} \left(r - c(i,j) + c(i,j) - c(i,j+1)\right)^2 + (r_{j+1} - c(i,j+1))^2.$$

By developing the previous formula, we have

$$SSE_{C(i,j+1)} = \sum_{r \in C(i,j)} (r - c(i,j))^2 + \sum_{r \in C(i,j)} (c(i,j) - c(i,j+1))^2$$

$$+2 \sum_{r \in C(i,j)} (r - c(i,j))(c(i,j) - c(i,j+1)) + (r_{j+1} - c(i,j+1))^2.$$

The proof concludes by noting that the first addend in the above expression is $SSE_{C(i,j)}$, the second addend is $(j-i)(c(i,j) - c(i,j+1))^2$, and the third addend equals 0. □

Algorithm 3 makes use of Proposition 6 to reduce the cost of computing the length of the edges. In each step of the inner loop, the SSE associated to the current cluster is computed from the SSE of the previous cluster, thereby reducing the computational cost to constant time complexity. Recall that the length of the edges is not SSE but SSE^ϵ (that is, SSE plus the variance of the DP centroid times the cluster size). Since the DP centroid can be computed using different random noises [23], we do not specify the value of the variance in the algorithm. However, when using the Laplace distribution adjusted to the global sensitivity, we have

$$c'_\epsilon = c' + Lap(0, \frac{\Delta_D}{\epsilon(j-i)})$$

and the variance becomes $Var(c'_\epsilon) = 2\left(\frac{\Delta_D}{\epsilon(j-i)}\right)^2$.

Algorithm 3 consists of three main steps: (i) sort the records in ascending order, (ii) insert the nodes, and (iii) compute the length and insert the edges. By using an appropriate sorting algorithm, the cost of the first step can be $\Theta(n \log n)$. The cost of the second step is linear in the size of the data set $\Theta(n)$. The third step deals with each of the $n(n+1)/2$ edges, each of them having constant cost; thus the overall cost of the third step is $\Theta(n^2)$. By adding the cost of the three steps, the computational cost of Algorithm 3 becomes $\Theta(n^2)$.

6.2. Optimal partition

According to [3], in univariate microaggregation any partition that minimizes SSE is formed by clusters with consecutive elements. We start by showing that this result is also true when using SSE^ϵ.

Lemma 1. *Let P be a partition of D in two clusters. For P to minimize the SSE^ϵ of the partitions in two clusters, the elements within each cluster must be consecutive.*

Algorithm 3 Graph construction algorithm

Require: $D = \{r_1, \ldots, r_n\}$ data set
Output: $G = (V, E)$ graph

set D'=records in D sorted in ascending order

// Insert the nodes
for $i = 0$ **to** n
  insert a node with label i in V
end for

// Insert the edges with the appropriate length
for $i = 0$ **to** $n - 1$
  let $sum = 0$ // sum of the records in the cluster
  let $c = 0$ //centroid of the records in the cluster
  let $sse = 0$ //SSE of the records in the cluster
  for $j = i + 1$ **to** n
    set $sum = sum + D'[j]$
    set $c = sum/(j - i)$
    if $j \neq i + 1$ **then**
      set $sse = sse + (j - i - 1)(c' - c)^2 + (D'[j] - c)^2$
    end if
    let c_ϵ be an ϵ-DP version of c
    insert the edge (i, j) with length $sse + (j - i)Var(c_\epsilon)$
    set $c' = c$
  end for
end for

return G

Proof. Let us assume that the clusters in partition $P = \{C_1, C_2\}$ do not have consecutive elements. Let us consider another partition $P' = \{C'_1, C'_2\}$ such that C'_1 contains the first $|C_1|$ records and C'_2 contains the remaining $|C_2|$ records. Let us now show that $SSE^{\epsilon}_{P'} < SSE^{\epsilon}_{P}$.

From Equation (2),

$$SSE^{\epsilon}_{P} = SSE_P + \sum_{C \in P} |C| Var(c_\epsilon(C));$$

$$SSE^{\epsilon}_{P'} = SSE_{P'} + \sum_{C \in P'} |C| Var(c_\epsilon(C)).$$

On the one side, we know from [3] that $SSE_{P'} < SSE_P$. On the other side, the second addends of SSE^{ϵ}_{P} and $SSE^{\epsilon}_{P'}$ are the same: the variability in $c_\epsilon(C)$ depends only on the domain (which is constant) and on the number of records in the clusters (which are equal in P and P', by the construction of P'). Thus, we conclude that $SSE^{\epsilon}_{P'} < SSE^{\epsilon}_{P}$. □

Lemma 2. *If $P = \{C_1, C_2, \ldots, C_l\}$ is the partition of the set of records in $\bigcup_{C \in P} C$ with minimum SSE^{ϵ}, then $P' = P \setminus C_i$ is the partition of the set of records in $\bigcup_{C \in P'} C$ with minimum SSE^{ϵ}.*

Proof. Let us assume that P' does not minimize SSE^{ϵ} in the set of records $\bigcup_{C \in P'} C$ and let P'' be a partition of $\bigcup_{C \in P'} C$ with $SSE^{\epsilon}_{P''} < SSE^{\epsilon}_{P'}$. Then

$$SSE^{\epsilon}_{P'' \cup \{C_i\}} = SSE^{\epsilon}_{P''} + SSE^{\epsilon}_{C_i} < SSE^{\epsilon}_{P'} + SSE^{\epsilon}_{C_i} = SSE^{\epsilon}_{P},$$

which contradicts the assumption that P is the partition of the set $\bigcup_{C \in P} C$ with minimum SSE^{ϵ}. □

Theorem 1. *A partition P of D that minimizes SSE^{ϵ} consists of clusters with consecutive elements.*

Proof. Let us assume that P contains clusters with non-consecutive elements. Let C_1 and C_2 be clusters of P such that they have interlaced elements (that is, at least one element in one cluster is greater than one element in the other cluster and smaller than another element in the other cluster). By Lemma 2, if P minimizes SSE^{ϵ} over D, then the partition $\{C_1, C_2\}$ must minimize the SSE^{ϵ} over the records in $C_1 \cup C_2$. However, by Lemma 1, partition $\{C_1, C_2\}$ cannot minimize the SSE^{ϵ} over the records in $C_1 \cup C_2$, because there are non-consecutive elements in C_1 or C_2. The contradiction comes from the assumption that P contains clusters with non-consecutive elements. □

Let us now show that any partition of D that minimizes SSE^{ϵ} can be viewed as a shortest path from node 0 to node n in the graph G. Notice that, by construction, the length of an edge (i_0, i_1) is equal to the SSE^{ϵ} associated with the cluster $\{i_0 + 1, \ldots, i_1\}$. Thus, the length of any path $\{(0, i_1), (i_1, i_2), \ldots, (i_{l-1}, n)\}$ from node 0 to node n accounts for the SSE^{ϵ} of the partition $\{\{1, \ldots, i_1\}, \ldots, \{i_{l-1}+1, n\}\}$. Since, by Theorem 1, any partition

that minimizes SSE^ϵ has the latter form (clusters with consecutive elements), the SSE^ϵ of the optimal partition equals the shortest path from node 0 to node n.

6.3. Computational cost

The computational cost of finding a partition that minimizes the SSE^ϵ equals the cost of generating the graph plus the cost of finding a shortest path. We have seen in Section 6.1 that the cost of building the graph is $\Theta(n^2)$. The cost of finding a shortest path in a directed acyclic graph is $\Theta\left(|V| + |E|\right)$, where V is the set of vertices and E is the set of edges. In our case, we have $n+1$ vertices and $n(n+1)/2$ edges, which leads to a cost $\Theta(n^2)$. Thus, the overall cost of finding a partition that minimizes SSE^ϵ is $\Theta(n^2)$.

7. Security analysis

In Section 4, we argued that it is not practical to generate a DP data set by collecting DP answers to the list of queries that ask for each of the records and, for that reason, we propose to query a microaggregated data set. In this section we evaluate, from a theoretical perspective, the effect of prior microaggregation on the accuracy of the DP data set. Following Section 6, the theoretical evaluation is based on the expected SSE. For the sake of simplicity, we assume a data set with a single attribute.

We start by computing the expected SSE of the generated data set when microaggregation is not used.

Proposition 7. *Let $D = \{r_1, \ldots, r_n\}$ be a data set that contains information about a numerical attribute A with values within the range $[min_A, max_A]$. Let D^ϵ be an ϵ-DP data set generated by collecting ϵ-DP answers to the set of queries that ask for each of the records:*

$$D^\epsilon = \{r_1 + Lap\left(0, \frac{\Delta_A}{\epsilon}\right), \ldots, r_n + Lap\left(0, \frac{\Delta_A}{\epsilon}\right)\},$$

where $\Delta_A = max_A - min_A$. Then it holds that the expected SSE of D^ϵ is $2n(\Delta_A/\epsilon)^2$.

Proof. Since we use the Laplace mechanism to mask the original records, the expected SSE can be computed as a sum of variances of Laplace distributions:

$$SSE = E\left(\sum_{i=1,\ldots,n} Lap\left(0, \frac{\Delta_A}{\epsilon}\right)^2\right)$$

$$= \sum_{i=1,\ldots,n} Var\left(Lap\left(0, \frac{\Delta_A}{\epsilon}\right)\right) = 2n\left(\frac{\Delta_A}{\epsilon}\right)^2.$$

□

Next, we compute the expected SSE of the DP data set when microaggregation is used. As shown in Section 6, in this case the expected SSE can be split into two parts: the SSE due to microaggregation and the sum of variances of the random noise. The former depends heavily on the actual data set. As the purpose is to show that microaggregation is useful at reducing the error, we take the worst-case scenario: a data set that maximizes the microaggregation error.

Proposition 8. *Let $D = \{r_1, \ldots, r_n\}$ be a data set that contains information about a numerical attribute A with values within the range $[min_A, max_A]$. Let $\bar{D} = \{c_{\rho(1)}, \ldots, c_{\rho(n)}\}$ be a microaggregated version of D generated by making clusters of k records and replacing each of the records in a cluster by the cluster centroid, where the function $\rho(i)$ return the cluster index that corresponds to record i. Let $\bar{D}^\epsilon$ be an ϵ-DP data set generated by collecting ϵ-DP answers to the queries that ask for each of the records (centroids) in $\bar{D}$:*

$$\bar{D}^\epsilon = \{c_{\rho(1)} + n_{\rho(1)}, \ldots, c_{\rho(n)} + n_{\rho(n)}\},$$

where n_i is a random number generated from a $Lap(0, \frac{\Delta_A}{k\epsilon})$. The expected SSE of $\bar{D}^\epsilon$ is upper-bounded by $k(\frac{\Delta_A}{2})^2 + 2n(\frac{\Delta_A}{k\epsilon})^2$.

Proof. According to Proposition 5, the SSE can be expressed as the the SSE due to the microaggregation plus the variances of the DP centroids.

The maximum microaggregation SSE is reached when the error is concentrated in a single cluster and, in particular, when the cluster has $k/2$ records at each of the two most distant ends of the attribute domain. The SSE is, then, $k(\Delta_A/2)^2$.

The sum of variances of the DP centroids is $\sum_{i=1,\ldots,n} Var(Lap(0, \Delta_A/(k\epsilon))) = 2n(\frac{\Delta_A}{k\epsilon})^2$. □

From Propositions 7 and 8, we conclude that the use of microaggregation is expected to improve the results if

$$2n\left(\frac{\Delta_A}{\epsilon}\right)^2 > k\left(\frac{\Delta_A}{2}\right)^2 + 2n\left(\frac{\Delta_A}{k\epsilon}\right)^2.$$

By operating on the previous expression, we can determine, for fixed values of ϵ and n, the conditions that k must satisfy to reduce the expected SSE. In particular, we conclude that

$$\frac{k^3}{k^2 - 1} < \frac{8n}{\epsilon^2}.$$

If we accept a small loss of precision, by using that $k < k^3/(k^2 - 1)$, we can express the previous inequality as

$$k < \frac{8n}{\epsilon^2}. \tag{3}$$

Keeping in mind that k must be smaller than n, Inequality (3) tells that for $\epsilon < \sqrt{8}$ the use of microaggregation reduces the SSE, regardless of the k used. For greater values of ϵ, Inequality (3) gives an upper bound for k, such that taking k less than the upper bound ensures that SSE is reduced. However, we should keep in mind that the previous upper bound is based on a (very unlikely) worst-case scenario for microaggregation. In less ill-conditioned data sets, SSE is likely to be reduced by microaggregation for k well above the bound (3).

The expected SSE given by Proposition 8 enables several other interesting analyses. For example, for given values of n and ϵ, we can compute the worst-case optimal value of k, that is, the one that minimizes the worst-case expected SSE. However, for most data sets, the optimal value of k is likely to be larger than the worst-case optimal value.

Let us now analyze the effect of the number of records on the accuracy of the generated data. When dealing with data sets with different number of records, the expected SSE is not a good measure, as it will be larger for the larger data set. In this case, we use the mean square error $MSE = SSE/n$. The expected MSE reflects the average contribution of a record to the expected SSE. Thus, a lower expected MSE is equivalent to a (overall) more accurate DP data set. We compute the worst-case expected MSE by dividing the expected SSE of Proposition 8 by the number of records:

$$\frac{k}{n}\left(\frac{\Delta_A}{2}\right)^2 + 2\left(\frac{\Delta_A}{k\epsilon}\right)^2. \tag{4}$$

The following proposition describes the effect of increasing the records if the cluster size remains unaltered.

Proposition 9. *If we increase the size of the data set by a factor t and the cluster size remains unaltered, the expected MSE due to microaggregation is divided by t and the expected MSE due to DP remains the same.*

Proof. The proof is simply a matter of replacing n by tn in Expression (4):

$$\frac{k}{tn}\left(\frac{\Delta_A}{2}\right)^2 + 2\left(\frac{\Delta_A}{k\epsilon}\right)^2.$$

We observe that the first addend (related to microaggregation) is divided by t with respect to Expression (4), whereas the second addend (related to the noise introduced by DP) remains unaltered. □

We can analyze the effect of the size from yet another point of view if, rather than keeping the cluster size constant, we increase it proportionally to the data set size. This is described in the following proposition.

Proposition 10. *If we increase the data set size and the cluster size by a factor t, the expected MSE error due to microaggregation remains unaltered and the expected MSE due to DP is divided by t^2.*

Proof. The proof is simply a matter of replacing n by tn, and k by kt in Expression (4):

$$\frac{k}{n}\left(\frac{\Delta_A}{2}\right)^2 + 2\left(\frac{\Delta_A}{kt\epsilon}\right)^2.$$

We see that the first addend (related to microaggregation) remains unaltered with respect to Expression (4), whereas the second addend (related to the noise introduced by DP) is divided by t^2. □

In general, DP data set generation methods fail to deliver accurate results when the number of attributes is large. When dealing with each attribute separately (for example, by using the method $MDAV\_IR\_DP$ described in Section 5), the greater the number of attributes, the smaller the share of the privacy budget that can be assigned to each attribute (and hence the more distorted is the attribute). Proposition 10 suggests one possible way to counter this effect, namely to increase the size of the data set and the cluster size proportionally to the increase of the number of attributes:

- By increasing the number of attributes by a factor s, we divide also by s the privacy budget ϵ_i assigned to each attribute A_i. It follows from Expression (4) that, as a result, the addend of the expected MSE related to the random noise is multiplied by a factor s^2, while the addend related to microaggregation remains unaltered.

- If we increase the size of the data set and the cluster size by a factor s, we compensate the effect of the number of attributes: the addend of Expression (4) related to the random noise is divided by s^2, while the addend related to microaggregation remains unaltered.

8. Experimental evaluation

We evaluate the proposal in Section 4 by choosing several microaggregation strategies and comparing the new proposal with existing methods that are also based on record perturbation [26, 22]. At first sight, employing standard microaggregation algorithms rather than (the more restrictive and less utility-preserving) insensitive microaggregation [26] seems a substantial advantage. Moreover, the method in Section 4 allows adjusting the noise to the size of each cluster.

A difference between the method of Section 4 and the methods in [26, 22] is that the former considers that the data set to be protected is the microaggregated one ($\bar{D}$), whereas the latter aim at protecting the original data set (D). Nonetheless, regardless of the method used, utility must be evaluated in terms of how good is the DP data set $\bar{D}^\epsilon$ as a replacement for the original data set D.

8.1. Evaluated methods

As mentioned above, our proposal can work with any microaggregation heuristic, but the utility of the output DP data set does depend on the way microaggregation is performed.

Hence, we evaluate the utility of MDAV_DP (multivariate MDAV microaggregation), MDAV_IR_DP (individual-ranking MDAV microaggregation), and the optimal microaggregation described in Section 6. The choice of MDAV is justified in Section 5. However, an important limitation of MDAV is that the clusters it generates have fixed cardinality k (except, maybe, the last cluster, that is of size between k and $2k-1$); but, as noted in Section 4, the method to generate DP data sets described in that section does not require a fixed cluster size, not even a minimum cluster size. The optimal microaggregation algorithm described in Section 6 drops the constraints on the cluster size and returns the partition that minimizes SSE^{ϵ}.

We have evaluated the following DP methods in our comparison:

- MDAV_DP, defined as in Section 5.
- MDAV_IR_DP_1. This is MDAV_IR_DP as defined in Section 5 and with the privacy budget ϵ evenly distributed among the attributes.
- MDAV_IR_DP_2. This is MDAV_IR_DP with the privacy budget ϵ distributed among attributes as a function of the attribute sensitivity. Attributes with greater sensitivity get a greater share of the privacy budget ($\epsilon_i = \epsilon\Delta_i/\Delta_D$).
- OPT_DP_1. This is Algorithm 2 where microaggregation is performed with the optimal univariate microaggregation method described in Section 6. The privacy budget ϵ is evenly distributed among the attributes.
- OPT_DP_2. Same as the previous case, but attributes with greater sensitivity get a greater share of the privacy budget ($\epsilon_i = \epsilon\Delta_i/\Delta_D$).
- INS_DP (baseline). The microaggregation step is based on the insensitive multivariate microaggregation described in [26]. This method is a suitable comparison baseline for MDAV_DP because both methods use multivariate microaggregation of entire records.

The method described in [22] could also be considered as a comparison baseline (it would be a good baseline for MDAV_IR_DP_1 and MDAV_IR_DP_2, because it is also based on individual-ranking MDAV microaggregation). However, we skip it because the computation of the sensitivity in [22] is flawed, which leads to overly reducing the noise required to attain DP.

Even if they do not yield DP, standalone MDAV (multivariate MDAV microaggregation), MDAV_IR (individual-ranking MDAV microaggregation), and OPT (optimal individual-ranking microaggregation) have also been evaluated. The reason is that they provide upper bounds on the accuracy reachable with MDAV_DP, MDAV_IR_DP and OPT_DP , respectively.

8.2. Evaluation data

We have used two different data sets:

- *Census data set* [1]. This data set was first used in the "CASC" European project and, since then, it has become a reference data set to test and compare statistical disclosure control methods. In particular, it was used in [26]. It contains 13 numerical attributes and 1,080 records. For the sake of comparability with [26], we focus on 4 attributes: FICA (Social security retirement payroll deduction), FEDTAX (Federal income tax liability), INTVAL (Amount of interest income) and POTHVAL (Total other persons income).

- *California housing data set* [19]. This data set contains information about housing prices that was extracted from the 1990 California census. It is larger than the previous data set: it contains 20,960 records and 9 attributes. Among those attributes, we focus on: Latitude, Longitude and Price. We use this data set to evaluate the effect of the data set size on the distortion caused by DP. To that end, we generate several subsets with a smaller number of records by randomly sampling from the original data set without replacement. In particular, we generate subsets of size 1,290, 2,580, 5,160, 10,320, and 20,640 records.

The selected attributes take values above 0 but they are not naturally upper-bounded. Since the L_1-sensitivity is proportional to the sizes of the domains of attributes, we need to upper-bound the domain of each attribute. For the sake of comparability, we use the upper bounds that were employed in [26]; that is, we upper-bound the domain of an attribute by 1.5 times the maximum value of the attribute in the data set. The domain bounds of the attributes are also enforced when adding noise to attain DP: the DP masked values are truncated to lie within the fixed bounds.

8.3. Evaluation measures

The evaluation is based on the SSE between the original and the DP data set:

$$SSE = \sum_{i=1,\ldots,n} \sum_{j=1,\ldots,m} (r_{ij} - r_{ij}^{\epsilon})^2$$

where r_{ij} is the value of attribute j in original record r_i and r_{ij}^{ϵ} is value of attribute j in the record r_i^{ϵ} of the DP data set $\bar{D}^{\epsilon}$ that corresponds to r_i.

8.4. Experimental results

We start by evaluating the proposed methods on the Census data set, and by comparing our results with [26]. Figure 3 shows the evolution of SSE as a function of the cluster size k. Notice that the cluster size parameter is only significant when using MDAV microaggregation. *Taking $k = 1$ (leftmost abscissa) means plain DP with no prior microaggregation.* Since OPT and OPT_DP do

not use a specific cluster size (but rather select clusters to minimize SSE), they are not affected by k.

In both graphs of Figure 3 we can see that, as expected, SSE for MDAV and MDAV_IR increases with the minimum cluster size k. There is a steep increase for small k that flattens out progressively as k grows. In contrast, for MDAV_DP, MDAV_IR_DP_1 and MDAV_IR_DP_2, the opposite occurs: SSE decreases with k and the decrease is steeper for small k. If we look at the plots in detail, we can observe that SSE decreases only up to a given (optimal) k and then it starts a slight increase. This was predicted in the theoretical analysis, but the real effect in a normal (rather than worst-case) data set is almost imperceptible. For example, if we look at the curve of MDAV_IR_DP_2 (the best method with a fixed cluster size) for $\epsilon = 1$, we observe that the minimum SSE is reached at around $k = 140$; if we switch to $\epsilon = 2$, the minimum SSE is reached at around $k = 90$. As expected, a greater value of ϵ reduces the optimal value of k. If we go beyond the optimal k, we observe that, for large k, the SSEs of all DP methods converge to the SSE of the underlying microaggregation. This result was to be expected because, the greater the cluster size, the less noise is needed to attain DP. As it can be seen by comparing both graphs, the rate of convergence is proportional to ϵ (faster convergence for larger ϵ, that is, faster convergence for less strict privacy requirements). The comparison between MDAV_DP and MDAV_IR_DP (both variants) shows that the MDAV_IR_DP variants yield a lower SSE. This was to be expected, because individual-ranking MDAV is more utility-preserving than multivariate MDAV. The comparison between MDAV_IR_DP_1 and MDAV_IR_DP_2 shows that the latter yields a slightly lower SSE than the former. This was also to be expected because, unlike MDAV_IR_DP_1, MDAV_IR_DP_2 assigns more privacy budget to attributes with greater sensitivity. The results obtained with OPT_DP (both variants) are very satisfactory. Not only the results for OPT_DP_1 and OPT_DP_2 are significantly better than the results obtained for MDAV_IR_DP_1 and MDAV_IR_DP_2, but they also improve on the results obtained with the bare microaggregation algorithms (MDAV and MDAV_IR). In particular, for $\epsilon = 2$, OPT_DP_2 yields less SSE than MDAV with $k \geq 20$, and less SSE than MDAV_IR with $k \geq 50$.

We then compared the SSE obtained with the methods in this paper with the SSE obtained with the method in [26]. Figure 4a in [26] shows the SSE of the DP data set generated by performing a prior insensitive microaggregation to reduce the noise needed to reach DP. By comparing that figure with Figure 3, we observe that MDAV_IR_DP with $\epsilon = 1$ performs as well as the insensitive approach INS_DP in [26] with $\epsilon = 10$. This is a very significant improvement in the utility of the data.

As the second part of the evaluation, we evaluate the effect of the data set size on the error, as measured by the expected $MSE = SSE/n$. This evaluation is performed on the California housing data sets: we compute the expected MSE for different versions of the previously mentioned California housing data sets (with sizes 1,290, 2,580, 5,160, 10,320 and 20,640). Figure 4 depicts the effect of the data set size on the expected MSE of the price attribute in algorithm

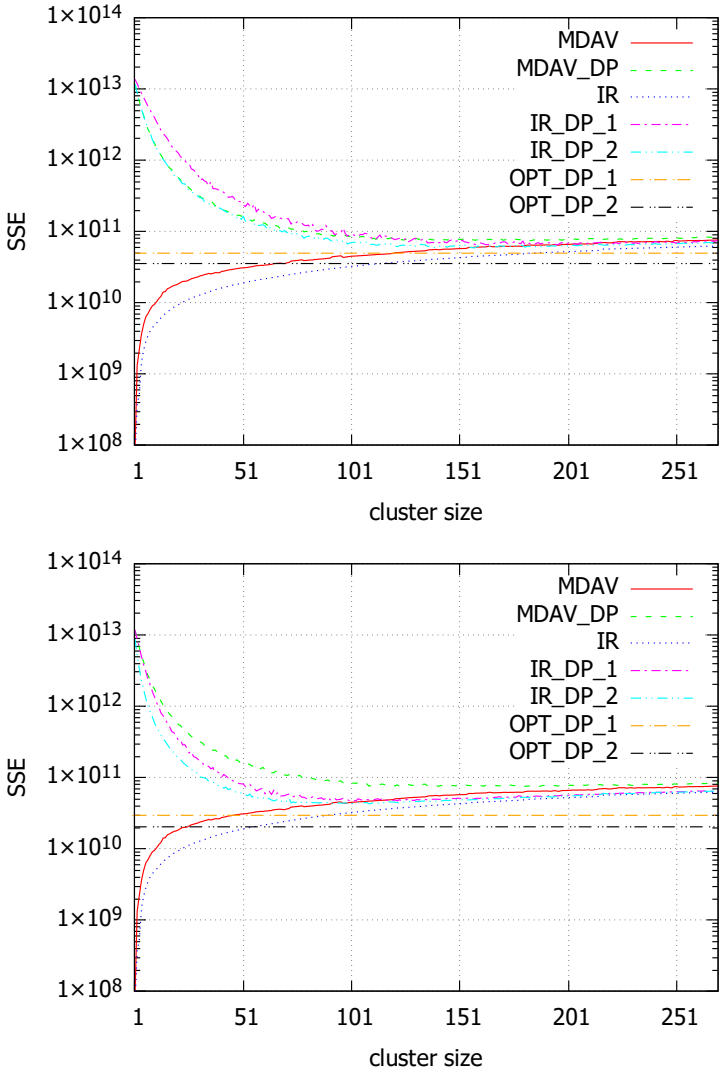


Figure 3: *Census data set.* Sum of squared errors SSE for $\epsilon = 1$ (top) and $\epsilon = 2$ (bottom).

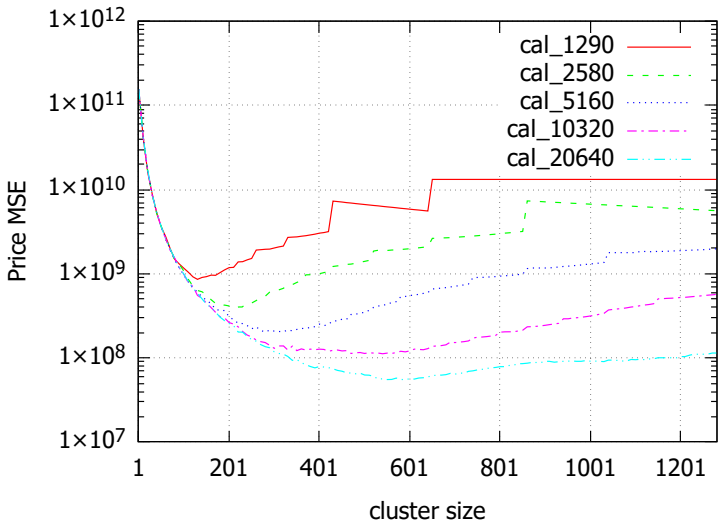


Figure 4: *California housing data sets.* Expected MSE of the house price attribute for algorithm MDAV_IR_DP_1 when $\epsilon = 1$ for different data set sizes.

MDAV_IR_DP_1 when $\epsilon = 1$. As expected, the total error decreases with the data set size, because the error due to microaggregation becomes smaller for larger data sets. Additionally, we see that the optimal cluster size depends on the data set size. This could also be expected as, when the microaggregation error is smaller, the overall effect of the error due to noise addition becomes more noticeable and, therefore, it makes sense to increase the cluster size to decrease the latter error.

In Figure 5, we evaluate the effect of the data set size on the expected MSE when the OPT_DP_1 algorithm with $\epsilon = 1$ is used. Like for MDAV_IR_DP_1, the expected MSE decreases as the data set size increases. Each line pattern corresponds to a data set of different size; the color straight lines show the MSE obtained with the OPT_DP_1 algorithm on the respective data set; the black curves show the MSE obtained with MDAV_IR_DP_1 on the respective data set. Interestingly, we observe that the minimum MSEs for MDAV_IR_DP_1 are close to the optimal values. In particular, increasing the data set size seems, in general, more effective than using the optimal algorithm.

9. Conclusions and future work

We have presented an approach to generate DP data sets that consists of adding noise to a microaggregated version of the original data set. Using microaggregation as a prior step to reduce the sensitivity of the data and hence the noise that needs to be added to reach DP had already been proposed in the literature. However, the novelty of our approach is that we focus on the

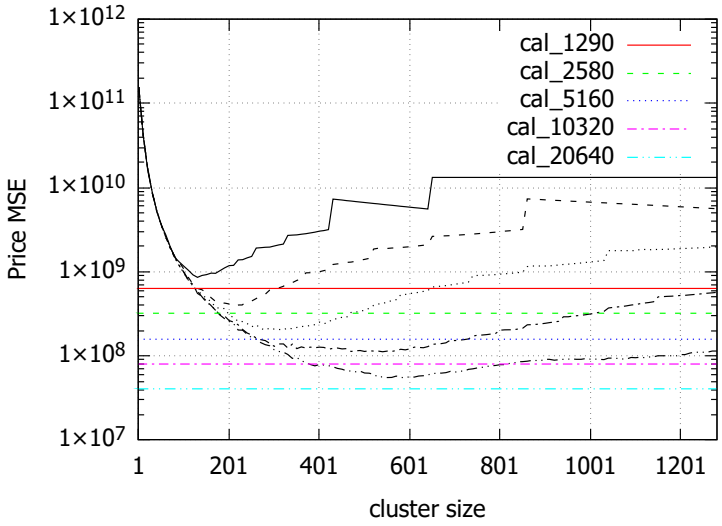


Figure 5: *California housing data sets.* Expected MSE of the house price attribute for $\epsilon = 1$ and different data set sizes. Each line pattern indicates a different data set size. The straight lines depict the optimal expected MSE obtained with OPT_DP_1 (which does not depend on the cluster size). The curves in black depict the expected MSE obtained with MDAV_IR_DP_1.

microaggregated data set as the target of protection (rather than focusing on the original data set and viewing the microaggregated data set as a mere intermediate step). As a result, we avoid the complexities inherent to insensitive microaggregation and we significantly improve the utility of the DP data. To start from the microaggregated data set, we had to extend DP to data sets without a one-to-one mapping between records and individuals. This has been done by keeping in mind that the aim of DP is to protect individuals rather than records.

The approach we have presented works with any microaggregation algorithm. For concreteness and convenience, we have analyzed five specific approaches to generate DP data sets: MDAV_DP, two variants of MDAV_IR_DP, and two variants of OPT_DP. Optimal microaggregation in the latter two variants is performed using a new algorithm that we have also presented in this paper.

The comparison (both theoretical and empirical) has shown that OPT_DP is better than MDAV_IR_DP, which in turn is better than MDAV_DP. Indeed, OPT_DP yields less SSE than the bare microaggregation algorithms MDAV and MDAV_IR (which do not ensure DP), already for moderate cluster sizes. Comparisons of MDAV_IR_DP with the method based on insensitive microaggregation of [26] have shown that MDAV_IR_DP with $\epsilon = 1$ is similar in terms of SSE to the insensitive approach with $\epsilon = 10$. This is a significant improvement in the utility with respect to prior work.

Additionally, we have shown that microaggregation reduces the expected

SSE significantly. We have computed a lower bound for the optimal cluster size, and we have analyzed the effect of the data set dimensionality in terms of number of records and number of attributes. Particularly interesting is the fact that the proposed methods can satisfactorily deal with a large number of attributes: to preserve the accuracy of the DP data when increasing the number of attributes by a factor s, we need to increase the number of records and the cluster size by the same factor.

Future work will include:

- Considering non-numerical data by using microaggregation algorithms capable of dealing with categorical data (ordinal, nominal or hierarchical);
- Trying aggregation operators different from the mean (e.g. the medoid) to compute the representative record of a cluster.

Acknowledgments and disclaimer

Partial support to this work has been received from the European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"), the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer), and from the Spanish Government (projects TIN2014-57364-C2-1-R "SmartGlacis" and TIN 2015-70054-REDC). The authors are with the UNESCO Chair in Data Privacy, but the views in this paper are their own and are not necessarily shared by UNESCO.

References

[1] R. Brand, J. Domingo-Ferrer and J. M. Mateo-Sanz. Reference data sets to test and compare SDC methods for the protection of numerical microdata. Deliverable of the EU FP5 "CASC" project, 2002. `http://neon.vb.cbs.nl/casc/CASCtestsets.htm`

[2] S. Chen and S. Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data-SIGMOD '13*, pp. 653–664, New York, NY, USA, 2013. ACM.

[3] 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.

[4] J. Domingo-Ferrer, J. M. Mateo-Sanz, A. Oganian and A. Torres. On the security of microaggregation with individual ranking: analytical attacks. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):477-492, 2002.

[5] J. Domingo-Ferrer, D. Sánchez and G. Rufián-Torrell. Anonymization of nominal data based on semantic marginality. *Information Sciences* 242:35-48, 2013.

[6] J. Domingo-Ferrer, F. Sebé and A. Solanas. A polynomial-time approximation to optimal multivariate microaggregation. *Computers & Mathematics with Applications*, 55(4):714-732, 2008.

[7] J. Domingo-Ferrer and V. Torra. A quantitative comparison of disclosure control methods for microdata. In *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, pp. 111-134. North-Holland, 2001.

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

[9] C. Dwork. Differential privacy. In *Automata, Languages and Programming-ICALP 20016*, LNCS 4052, pp. 1–12. Springer, 2006.

[10] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In *Third Theory of Cryptography Conference-TCC 2006*, LNCS 3876, pp. 265–284. Springer, 2006.

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

[12] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data-SIGMOD '11*, pp. 193–204, New York, NY, USA, 2011. ACM.

[13] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: privacy beyond k-anonymity and l-diversity. In *23th IEEE International Conference on Data Engineering-ICDE 2007*, pp. 106–115. IEEE, 2007.

[14] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: theory meets practice on the map. In *24th IEEE International Conference on Data Engineering-ICDE 2008*, pp. 277–286, 2008.

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), March 2007.

[16] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science-FOCS 2007*, pp. 94–103, Washington DC, 2007. IEEE Computer Society.

[17] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *39th Annual ACM Symposium on Theory of Computing-STOC 2007*, pp. 75–84, New York, NY, USA, 2007. ACM.

[18] A. Oganian and J. Domingo-Ferrer. On the complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal of the United Nations Economic Commission for Europe*, 18(4):345–354, 2001.

[19] R. K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[20] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[21] D. Sánchez, J. Domingo-Ferrer, and S. Martínez. Improving the utility of differential privacy via univariate microaggregation. In *Privacy in Statistical Databases-PSD 2014*, LNCS 8744, pp. 130–142. Springer, 2014.

[22] D. Sánchez, J. Domingo-Ferrer, S. Martínez, and J. Soria-Comas. Utility-preserving differentially private data releases via individual ranking microaggregation. *Information Fusion*, 30:1 – 14, 2016.

[23] J. Soria-Comas and J. Domingo-Ferrer. Optimal data-independent noise for differential privacy. *Information Sciences*, 250:200–214, 2013.

[24] J. Soria-Comas and J. Domingo-Ferrer. Big data privacy: challenges to privacy principles and models. *Data Science and Engineering*, 1(1):21–28, 2015.

[25] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez. Improving the utility of differentially private data releases via k-anonymity. In *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications-TrustCom 2013*, pp. 372–379, 2013.

[26] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *The VLDB Journal*, 23(5):771–794, 2014.

[27] Y. Xiao, L. Xiong, and C. Yuan. Differentially private data release through multidimensional partitioning. In *Secure Data Management*, LNCS 6358, pp. 150–168. Springer, 2010.

[28] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: private data release via bayesian networks. In *2014 ACM SIGMOD International Conference on Management of Data-SIGMOD '14*, pp. 1423–1434, New York, NY, USA, 2014. ACM.