TRANSACTIONS ON DATA PRIVACY 18 (2025) 29–50

Conciliating Privacy and Utility in Data Releases via Individual Differential Privacy and Microaggregation

Jordi Soria-Comas[a], **David Sánchez**[b*], **Josep Domingo-Ferrer**[b], **Sergio Martínez**[b], **Luis Del Vasto-Terrientes**[b]

[a]Universitat Oberta de Catalunya, Rambla del Poblenou 156, Barcelona, 08018, Catalonia

E-mail: jordi_sc@uoc.edu

[b]Department of Computer Engineering and Mathematics, CYBERCAT-Center for Cybersecurity Research of Catalonia, Universitat Rovira i Virgili, Av. Paisos Catalans 26, Tarragona, 43007, Catalonia

E-mail: {david.sanchez,josep.domingo,sergio.martinezl}@urv.cat, luismiguel.delvasto@fundacio.urv.cat

Received 21 December 2023; received in revised form 31 July 2024; accepted 7 August 2024

Abstract. ϵ-Differential privacy (DP) is a well-known privacy model that offers strong privacy guarantees. However, when applied to data releases, DP significantly deteriorates the analytical utility of the protected outcomes. To keep data utility at reasonable levels, practical applications of DP to data releases have used weak privacy parameters (large ϵ), which dilute the privacy guarantees of DP. In this work, we tackle this issue by using an alternative formulation of the DP privacy guarantees, named ϵ-individual differential privacy (iDP), which causes less data distortion while providing the same protection as DP to subjects. We enforce iDP in data releases by relying on attribute masking plus a pre-processing step based on data microaggregation. The goal of this step is to reduce the sensitivity to record changes, which determines the amount of noise required to enforce iDP (and DP). Specifically, we propose data microaggregation strategies designed for iDP whose sensitivities are significantly lower than those used in DP. As a result, we obtain iDP-protected data with significantly better utility than with DP. We report on experiments that show how our approach can provide strong privacy (small ϵ) while yielding protected data that do not significantly degrade the accuracy of secondary data analysis.

Keywords. Individual differential privacy, Data releases, Data microaggregation, Machine learning

1 Introduction

Data analysis has become an essential tool in today's world. Its applications range from the enhancement of customers' experience (*e.g.* via recommender systems) to the support of strategic decision making (*e.g.* using data mining), and in general may substantially

*Corresponding author

improve human life and human endeavors. However, when using data on people for secondary purposes, the privacy of the subjects in the data set must be preserved. This is increasingly important under the new stronger privacy regulations, epitomized by the European General Data Protection Regulation (GDPR).

Personally identifiable information (PII) should be protected before releasing or sharing it for analysis. Several approaches are possible: one may release a fixed set of statistics, offer interactive access to the PII via some query mechanism that provides anonymized answers, or release an anonymized data set. The latter is the most convenient way for the data analyst, and the one that we consider in this work [1]. Releasing sufficiently useful protected data sets gives freedom to the data analyst to carry out unconstrained exploratory data analyses, data mining tasks or even machine learning at will.

We use the term microdata set to refer to a data set whose records contain detailed information about a single subject. Privacy protection in microdata releases is a discipline with a long history. It was initially developed in the context of official statistics under the name of Statistical Disclosure Control (SDC) [1]. At that time, the number of data controllers was limited, and the data were collected under a strong pledge of privacy. This allowed making reasonable assumptions about the side knowledge available to intruders for them to conduct inference attacks leading to disclosure. Such assumptions were very useful to adjust the SDC methods in view of obtaining adequate disclosure protection. However, with the development of IT, the landscape changed radically. Nowadays, large amounts of heterogeneous personal data are collected by a large number of data controllers [2], public and private; in this context, making assumptions about the knowledge available to intruders is quite difficult, and using robust privacy models to protect data releases seems the best option.

ϵ-Differential privacy [3] (ϵ-DP) is a well-known privacy model whose privacy guarantees are independent of the side knowledge available to intruders. This makes DP particularly suitable in the current landscape. However, unlike other privacy models originally developed for data releases (*e.g.* k-anonymity [4], l-diversity [5] or t-closeness [6]), DP was initially designed for interactive settings, in which only the outcomes of queries submitted to a database held by a trusted party are protected.

More recently, DP has also been used in the more convenient non-interactive setting, in which the aim is to release anonymized data sets [7, 8, 9, 10, 11] Yet, in such a setting, DP introduces large distortions in the protected outcomes thereby significantly hampering their analytical utility. As a result, DP has only being deployed to a limited extent in real-world applications and, when done, the privacy requirements (ϵ value) have been severely relaxed in order to keep data reasonably useful [12]. A paradigmatic example is the recent use of DP by the U.S. Census Bureau to protect the 2020 Decennial Census release [13]. To retain some utility, they were forced to take $\epsilon = 39.9$ [14] and, even with this large value, data utility significantly degraded w.r.t. the former Census releases using non-DP data protection [15]. In this sense, it is well-known that employing ϵ values larger than 1 dilutes the DP privacy guarantees until the point that *DP delivers privacy mostly in name* [16].

In order to reconcile data utility with DP-like privacy guarantees, we proposed ϵ-individual differential privacy [17] (ϵ-iDP), a privacy model that can incur less information loss than the standard DP, while giving subjects the same privacy protection as DP. The focus of this work is to design mechanisms to use iDP in data releases and thereby benefit from its enhanced utility-privacy trade-off.

Contribution and Plan of This Paper

To take advantage of iDP in data releases, we propose several strategies based on data microaggregation [18] whose local sensitivities are significantly lower than the global sensitivity required in standard DP. In this way, we enable privacy-preserving data releases that offer the robust privacy guarantees of DP at the individual level while preserving data utility significantly better than standard DP.

Due to its definition, the microaggregation-based iDP-protected data we obtain can never offer less protection than the underlying microaggregation, which hides individuals in a group. In fact, our work shows that it does much better: *our approach allows using the small values of ϵ recommended in [19]* (the only ones that are actually meaningful in DP-like privacy models) *while maintaining analytical utility to an extent that standard DP cannot offer for such small values*.

We validate the above through a set of experiments on several standard data sets, whose utility is evaluated through general-purpose utility metrics and in machine learning tasks.

The rest of this paper is organized as follows. In Section 2 we give background on DP and iDP. In Section 3, we review our approach to DP data releases and detail our proposal to generate iDP data sets through carefully tailored microaggregation strategies. In Section 4, we report on the experiments we conducted on several data sets. Section 5 gathers conclusions and identifies future research lines.

2 From Differential Privacy to Individual Differential Privacy

Differential privacy [3] stands out because of the strong privacy guarantees it offers. DP does not make any assumptions about the side 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. Assuming that each record corresponds to an individual, the rationale of DP is explained in [20]:

> 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.

With DP, individuals should not be reluctant to participate in the data set because the risk of disclosure is only very marginally increased by such participation. Differential privacy assumes a trusted party that: (i) holds the database, (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 if, for all data sets D_1 and D_2 that differ in one record (*a.k.a.* neighbor data sets), and all $S \subset Range(\kappa)$, we have

$$\Pr(\kappa(D_1) \in S) \leq \exp(\epsilon) \Pr(\kappa(D_2) \in S).$$

Safe values for ϵ are 0.01, 0.1 [19]; for larger values, the privacy guarantees of DP tend to vanish because the DP-protected outcomes may be substantially affected by the presence or absence of each individual, which increases the disclosure risk [17].

DP has composability properties, that is, aggregating several differentially private results still satisfies DP although, sometimes, with a different ϵ.

Theorem 2 (Sequential composition)**.** *Let κ_1 be a randomized function giving ϵ_1-DP and κ_2 a randomized function giving ϵ_2-DP. Then, any deterministic function of (κ_1, κ_2) gives $(\epsilon_1 + \epsilon_2)$-DP.*

Theorem 3 (Parallel composition)**.** *Let κ_1 and κ_2 be randomized functions giving ϵ-DP. If κ_1 and κ_2 are applied to* disjoint *data sets or subsets of records, any deterministic function of (κ_1, κ_2) gives ϵ-DP.*

For a numerical query f, ϵ-DP can be attained via noise addition; that is, by adding some random noise to the actual query result: $\kappa(x) = f(x) + N$. The amount of noise that needs to be added depends on the variability of the query function between neighbor data sets, that is, on the global sensitivity of the query.

Definition 4 (Global sensitivity)**.** Let f be a function that is evaluated at data sets in $\mathcal{D}$ and returns values in $\mathbb{R}^k$. The global sensitivity of f in $\mathcal{D}$ is

$$\Delta f = \max_{\substack{D_1, D_2 \in \mathcal{D} \\ d(D_1, D_2) = 1}} \|f(D_1) - f(D_2)\|_1 \,,$$

where $d(D_1, D_2)$ means that data sets D_1 and D_2 differ in one record.

Even though several noise distributions are possible, the Laplace distribution [3] is the most commonly employed one.

As mentioned in Section 1, the deployment of DP has been limited in practice, in spite of its strong privacy guarantees. In fact, those guarantees are only meaningful for very small ϵ values, but practitioners need to use large, unsafe ϵ values to preserve sufficient utility. In an attempt to improve the accuracy of the protected data, several relaxations of DP have been proposed, such as (ϵ, δ)-DP [21], concentrated DP [22], zero-concentrated DP [23] or Rényi DP [24]. Essentially, these relaxations allow DP guarantees to be broken either by a small amount or with a small probability. An alternative relaxation of DP is iDP [17]. The latter is particularly interesting because, unlike the above, it preserves the privacy guarantees that DP gives to individual subjects for a given ϵ.

Next, we recall the rationale of iDP. When formalizing DP in Definition 1, the trusted party is not allowed to take advantage of her knowledge about the actual data set to adjust the level of noise. This leads to the formalization of DP being stricter than required by the intuition of DP (see quotation above), which results in unnecessary accuracy loss. Let us explain this in greater detail.

Consider an individual subject I who has to decide between participating in a data set or not. To neutralize any reluctance by I to disclose her private information, I is told that query answers based on the data set will not allow anyone to learn anything that was not learnable without I's presence; this is precisely the intuitive privacy guarantee DP offers. To attain such privacy guarantees, DP requires the response to be indistinguishable between any pair of neighbor data sets. While such a requirement yields the target privacy guarantees, it is an overkill because the trusted party is not allowed to take advantage of her knowledge of the data set. In other words, if D is the collected data set, the target privacy guarantees can be attained by just requiring indistinguishability of the responses between D and its neighbor data sets. Notice that, although the data set D is not known until all the individuals have made their decisions about participating/contributing to it, it is known to the trusted party at the time of query response.

According to the previous discussion, ϵ-individual differential privacy (ϵ-iDP) is defined as follows.

Definition 5 (ϵ-Individual differential privacy [17]). Given a data set D, a response mechanism $\kappa(\cdot)$ satisfies ϵ-individual differential privacy (or ϵ-iDP) if, for any neighbor data set D' of D, and any $S \subset Range(\kappa)$ we have

$$\begin{aligned}&\exp(-\epsilon)\Pr(\kappa(D') \in S) \leq \Pr(\kappa(D) \in S)\\ &\leq \exp(\epsilon)\Pr(\kappa(D') \in S).\end{aligned}$$

In line with Definition 1, iDP requires the probability of any result to differ between neighbor data sets at most by a factor $\exp(\epsilon)$. However, unlike in Definition 1, the role of the data sets D and D' is not exchangeable: D refers to the actual data set, and D' to a neighbor data set of D. The asymmetry between D and D' is relevant, because indistinguishability is achieved only between D and its neighbor data sets. As a side effect of this asymmetry, we need to explicitly enforce an upper bound ($\Pr(\kappa(D) \in S) \leq \exp(\epsilon)\Pr(\kappa(D') \in S)$) and a lower bound ($\exp(-\epsilon)\Pr(\kappa(D') \in S)) \leq \Pr(\kappa(D) \in S)$). This was not needed in Definition 1 because the upper bound could be obtained from the lower bound by exchanging the roles of D and D'.

This difference has an important and beneficial practical consequence. Unlike DP, which requires calibrating the added noise to the global sensitivity (that is, to the greatest change between any pair of neighbor data sets), iDP can be attained by calibrating noise to local sensitivity (which is normally much lower).

Definition 6 (Local sensitivity [25]). The local sensitivity of a query function f at a data set D is

$$LS_f(D) = \max_{y:d(y,D)=1} \|f(y) - f(D)\|_1 ,$$

where $d(y, D)$ means that data set y differs from D in one record.

The following result is proven in [17].

Proposition 7. *Let f be a query function that takes values in $\mathbb{R}^k$. The mechanism $\kappa(x) = f(x) + (N_1, \ldots, N_k)$, where N_i are independent identically distributed $Laplace(0, LS_f(D)/\epsilon)$ random noises, gives ϵ-iDP.*

3 iDP Data Sets via Individual Ranking Microaggregation

Even though the natural application of DP and iDP is the interactive setting, both can be used to generate protected data sets via data masking. In the following, we first discuss how this can be achieved for standard DP, and then we tailor masking to reap the utility-preserving advantages of iDP.

3.1 DP Data Sets

For years, the usual approach to generate DP data sets was based on computing DP histograms [8, 7]; that is, on approximating the data distribution by partitioning the data domain and counting the number of records in each partition set in a DP manner. However, histogram-based approaches have severe limitations when the number of attributes grows: for a fixed granularity in each attribute, the number of histogram bins grows exponentially

$$
\begin{array}{c|ccc|c|ccc}
 & & D & & & & \bar{D} & \\
\hline
 & A^1 & \dots & A^m & & A^1 & \dots & A^m \\
\hline
I_1 & x_1^1 & \dots & x_1^m & & c^1_{\rho_1(I_1)} & \dots & c^m_{\rho_m(I_1)} \\
I_2 & x_2^1 & \dots & x_2^m & \rightarrow & c^1_{\rho_1(I_2)} & \dots & c^m_{\rho_m(I_2)} \\
\vdots & \vdots & & \vdots & & \vdots & & \vdots \\
I_n & x_n^1 & \dots & x_n^m & & c^1_{\rho_1(I_n)} & \dots & c^m_{\rho_m(I_n)}
\end{array}
$$

Figure 1: Generation of $\bar{D}$ via univariate microaggregation of each attribute in D

with the number of attributes, which has a devastating effect on both computational cost and accuracy. To mitigate these issues, an alternative approach has been proposed that is based on masking attribute values of the records in the original data set [9, 10]. In order to reduce the sensitivity to record changes (which is the factor that basically determines the noise needed to attain DP and hence the utility damage incurred), this approach applies a microaggregation step before masking.

Let D be the collected data set. Assume that one wants to generate D_ϵ –an anonymized version of D– that satisfies ϵ-DP. Let $I_r(D)$ be the query that returns r. Then, one can think of the data set D as the collected answers to the queries $I_r(D)$ for $r \in D$, and one can generate D_ϵ by collecting ϵ-DP responses to the previous queries. However, since the purpose of DP is to make sure that individual records do not have any significant effect on query responses, the amount of noise that should be added to the responses of $I_r(D)$ to fulfill DP is necessarily high, which severely deteriorates the accuracy of D_ϵ.

To make masking a viable option for generating DP data sets, the sensitivity of individual records needs to be reduced. This was attained in [10, 26] for standard DP by adding a microaggregation step before the actual DP data set generation. Even though microaggregation was initially proposed as an anonymization technique in its own right [18], in this work we will use it as a means to reduce the sensitivity of the queries.

Microaggregation proceeds in two steps:

1. Split the data set into clusters of similar records of cardinality greater than or equal to k (a given parameter).

2. Compute a representative record of each cluster and replace each record in the cluster by a copy of the representative.

In Figure 1 we describe the generation of a microaggregated data set $\bar{D}$ via univariate microaggregation of each attribute in D. Let D contain data about attributes $A^1, \dots, A^m$, for individuals $I_1, \dots, I_n$. To microaggregate attribute A^a, we cluster records by their similarity w.r.t. attribute A^a, compute the centroid for A^a of each cluster, and replace the original values in A^a by the corresponding centroid. Formally, let $C^a = \{C^a_j\}_j$ be the clustering associated with attribute A^a, let c^a_j be the centroid associated with cluster C^a_j, and let $\rho_a(I_i)$ be the index of the centroid associated with individual I_i. We replace x^a_i (the value of attribute A^a for individual I_i) by the corresponding centroid $c^a_{\rho_a(I_i)}$.

Once the microaggregated data set $\bar{D}$ has been created, we generate $\bar{D}_\epsilon$, a DP version of $\bar{D}$, through the process depicted in Figure 2.

- We work independently with each attribute A^a to make it ϵ_a-DP, where ϵ_a is the share of the privacy budget assigned to attribute A^a. Afterwards, we combine all the DP

$$\begin{array}{c|ccc} & & \bar{D} & \\ & A^1 & \dots & A^m \\ \hline I_1 & c^1_{\rho_1(I_1)} & \dots & c^1_{\rho_m(I_1)} \\ I_2 & c^1_{\rho_1(I_2)} & \dots & c^1_{\rho_m(I_2)} \\ \vdots & \vdots & & \vdots \\ I_n & c^1_{\rho_1(I_n)} & \dots & c^1_{\rho_m(I_n)} \end{array} \rightarrow \begin{array}{ccc} & \bar{D}_\epsilon & \\ A^1 & \dots & A^m \\ \hline c^1_{\rho_1(I_1)} + n^1_{\rho_1(I_1)} & \dots & c^m_{\rho_m(I_1)} + n^m_{\rho_m(I_1)} \\ c^1_{\rho_1(I_2)} + n^1_{\rho_1(I_2)} & \dots & c^m_{\rho_m(I_2)} + n^m_{\rho_m(I_2)} \\ \vdots & & \vdots \\ c^1_{\rho_1(I_n)} + n^1_{\rho_1(I_n)} & \dots & c^m_{\rho_m(I_n)} + n^m_{\rho_m(I_n)} \end{array}$$

where n^a_j is drawn from $Laplace(0, \Delta c^a_j/\epsilon_a)$

Figure 2: Generation of $\bar{D}_\epsilon$ by masking the centroids with the appropriate amount of noise

attributes to generate the ϵ-DP data set. By sequential composition (see Theorem 3), the overall privacy budget must be split among each of the attributes: $\epsilon = \sum \epsilon_a$.

- In the generation of the ϵ_a-DP attributes, parallel composition applies (see Theorem 2) because each centroid depends on a disjoint set of individuals (records). Thus, we can use the entire privacy budget assigned to attribute A^a, ϵ_a, to mask each of the centroids associated with A^a. This is done by masking each centroid c^a_j (*e.g.*, via Laplace noise) according to its global sensitivity (Δc^a_j) and privacy budget ϵ_a.

The global sensitivity of a centroid is the maximum change in the centroid value that can ensue from a change in one of the records in the cluster. Since the sensitivity of the centroids is smaller than the sensitivity of the original records, making $\bar{D}$ differentially private requires less noise than making D differentially private. Loosely speaking, centroids are less sensitive than individual records because the former are an aggregation of several records. Counterintuitively, even though the prior microaggregation step distorts data to some extent, 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 for a given ϵ, as shown in [10]. This is so because the noise reduction enabled by the prior microaggregation (due to cluster centroids being less sensitive than individual records), more than compensates the information loss introduced by such microaggregation; specifically, whereas noise is random, microaggregation can exploit the underlying structure of data.

Algorithm 1 formalizes the process described in Figures 1 and 2. The algorithm receives the original data set, the privacy budget assigned to each attribute and the microaggregation algorithm associated with each attribute. For each attribute, we run the microaggregation (line 08), compute the global sensitivity of the centroids (line 10), draw the noise from the Laplace distribution (line 11), and mask each occurrence of the centroid (line 12). The algorithm receives the microaggregation algorithms as a parameter. As discussed in [10], a sensible choice is to use individual-ranking microaggregation (form clusters containing consecutive values of the attribute) and compute the centroid as the arithmetic mean of the values in the cluster:

$$c^a_j = \frac{1}{|C^a_j|} \sum_{x \in C^a_j} x.$$

In this case, the global sensitivity of a centroid equals the size of the attribute domain over the size of the associated cluster:

$$\Delta c^a_j = \frac{\max A^a - \min A^a}{|C^a_j|},$$

Algorithm 1 ϵ-DP data set generation via univariate microaggregation and Laplace noise addition

```
Require:
    D : data set with attributes A^1, ..., A^m
    ε_a : privacy budget assigned to A^a (with ε = Σ ε_a)
    M_a: univariate microaggregation algorithm over attribute A^a
Output:
    D̄_ε: ε-DP data set

for a = 1 to m
    let (C_j^a, c_j^a)_j = the clusters and centroids produced by applying M_a to A^a
    for each C_j^a
        let Δc_j^a be the global sensitivity of cluster c_j^a
        let n_j^a be a draw from a Laplace(0, Δc_j^a/ε_a) distribution
        Replace each c_j^a ∈ C_j^a by c_j^a + n_j^a
    end for
end for
return D
```

where $\max A^a$ is the maximum of the domain of A^a, and $\min A^a$ is the minimum of the domain of A^a.

3.2 iDP Data Sets

To enforce DP, the proposal described in Algorithm 1 needs to use the global sensitivity of the attributes. Unfortunately, global sensitivities can be very large, especially when attributes have domains that are much larger than their actual value ranges in the original data set. This results in much noise being added, which severely damages the accuracy of the generated data set. Also, it may be difficult to bound the domain of some attributes (what is the upper bound of an attribute such as *income*?) or it may be thoroughly impossible (if the attribute is naturally unbounded). In such cases, attribute domains should be artificially limited to reasonably large bounds, which will also significantly increase the global sensitivity. These issues are the result of mechanisms enforcing DP not being allowed to leverage the knowledge of D, due to the strict formulation of DP (see Section 2).

In the following, we face these issues with a proposal to generate iDP data sets. Under iDP, individual subjects are given the same privacy guarantees as under DP (*i.e.*, the presence or absence of any single subject's data does not have a significant effect on the protected data set), but the accuracy of the iDP data set is better because local sensitivities are used rather than global ones. One significant distinction between DP and iDP is that the latter permits the trusted party to leverage its understanding of the actual data set to calibrate the noise addition. Consequently, by evaluating the data set's characteristics, such as size and attribute distribution, trusted parties can implement privacy mechanisms that uphold individual privacy while enhancing the utility of the data.

First, we describe a naive adaptation of the approach described above for DP to iDP: the global sensitivity is merely replaced by the local sensitivity. Then, we take several steps to use the knowledge of D and to further reduce the sensitivity. This is done by introducing a pre-processing step that alters the data set before microaggregation.

3.2.1 iDP Data Sets Using Local Sensitivity

By using iDP, we can adjust the noise to the local sensitivity of the attributes rather than to their global sensitivity:

- The local sensitivity of an attribute centroid is the maximum change in the centroid value that can occur when switching from D to a neighbor data set. In other words, it is the maximum change of the centroid value that can result from a change in one of the records in D. In the worst case, we may change the smallest record in the cluster w.r.t. attribute A to $\max A$ (the maximum value of A's domain), or change the largest record in the cluster to $\min A$ (the minimum value of A's domain).
- On the other hand, the global sensitivity of a centroid is the maximum of the local sensitivity across all pairs of neighbor data sets. Thus, in general, the global sensitivity is greater than the local sensitivity. They are equal only if the original data set is one in which the distance between the smallest (largest) record to $\max A$ ($\min A$) equals the distance between $\max A$ and $\min A$.

The first mechanism we propose is an adaptation of Algorithm 1 to iDP: we merely replace the global sensitivity (Δc_j^a) by the local sensitivity at $\bar{D}$, which is computed for each cluster centroid ($LS_{c_j^a}(\bar{D})$). Notice that, since each centroid is computed on a disjoint set of records, by parallel composition we can work with each centroid independently. More specifically, the changes to Algorithm 1 are:

- At line 10, we compute $LS_{c_j^a}(\bar{D})$, the local sensitivity of the centroid c_j^a at $\bar{D}$.
- At line 11, we draw the noise n_j^a from a $Laplace(0, LS_{c_j^a}(\bar{D})/\epsilon_a)$ distribution.

To compute $LS_{c_j^a}(\bar{D})$, we need to fix the way in which attribute centroids are computed. The following proposition gives the expression for the local sensitivity of a centroid when the centroid is computed as the arithmetic mean of the attribute values in the cluster.

Proposition 8. *Let $C^a = \{C_1^a, \ldots, C_p^a\}$ be a clustering of records created w.r.t. the values of attribute A^a. Let $c_j^a = \frac{1}{|C_j^a|}\sum_{x\in C_j^a} x^a$, for $j = 1, \ldots, p$, be the centroid associated with cluster C_j^a, where x^a is the value of record x for attribute A^a. The local sensitivity for each centroid, $LS_{c_j^a}(\bar{D})$, is*

$$\frac{\max\{\max A^a - \min_{x\in C_j^a} x^a,\ \max_{x\in C_j^a} x^a - \min A^a\}}{|C_j^a|}$$

where $\max A^a$ and $\min A^a$ are, respectively, the maximum and the minimum of the domain of A^a.

Proof. The local sensitivity of c_j^a measures the greatest change in c_j^a that can occur as a consequence of a change in one of the records in C_j^a. Since the centroid is computed as the arithmetic mean of the values of attribute A^a for all records in the cluster, the largest change in c_j^a happens when the change in the record for such attribute is greatest.

Specifically, the maximum change in a record $x \in C_j^a$ is reached when changing the record value x^a by one of the extremes of the domain: either $\max A^a$ or $\min A^a$. Thus, we can express the local sensitivity as

$$LS_{c_j^a}(\bar{D}) = \frac{\max_{x\in C_j^a}\{\max A^a - x^a,\ x^a - \min A^a\}}{|C_j^a|},$$

which is equivalent to

$$\frac{\max\{\max A^a - \min_{x\in C_j^a} x^a,\, \max_{x\in C_j^a} x^a - \min A^a\}}{|C_j^a|}.$$

$\square$

3.2.2 iDP Data Sets Using Cluster-based Local Sensitivity

As discussed in Section 3, to compute the global and local sensitivities we need the domains of attributes to be bounded. However, for some attributes (*e.g.*, income), there may not be a natural limit and we may need to artificially bound the attributes to apply the algorithm. Bounding the attribute domain to the maximum and minimum values in the data set is problematic, because those maximum and minimum values correspond to specific individuals and DP is designed precisely to hide information about any individual. Alternatively, we can fix the bounds in a way that is independent from the actual data set. However, if we fix the bounds without taking the original data set into account, the empirical distribution of the data set is likely to be more compact than the data-independent bounds. This problem also arises with attribute domains that are bounded but include a few outliers within their bounds. In either case, we will get local sensitivities much larger than the ones that correspond to the actual data, which leads to adding much noise and hence to poor accuracy. Notice that the approach proposed in the previous section to generate iDP data sets may also have the same outlier-related issues, because it uses the attribute bounds to compute the local sensitivity (see Proposition 8).

In this section we tackle these problems and improve the generation of iDP data sets by making the local sensitivity of a centroid depend only on the values within the corresponding cluster. This has two important advantages: (i) we avoid the shortcoming of unbounded attributes, and (ii) we reduce the local sensitivity (and thus the amount of noise required) when the attribute values in the cluster do not span the entire domain.

To make the local sensitivity of a centroid depend only on the values of the associated cluster, we apply the following pre-processing to the data set before using the method described in Section 3.2.1. For each attribute A^a and cluster C_j^a, we select one individual among those with the smallest value for A^a in C_j^a and replace its value by the second smallest value of A^a in C_j^a; similarly, we select one individual among those with the largest value for A^a in C_j^a and replace its value by the second largest value of A^a in C_j^a. It is important to note that our definition of second smallest (second largest) does not necessarily imply that it is a different value from the smallest (largest); if there are two or more individuals that have the smallest (largest) value, then the second smallest (largest) value is the same as the smallest (largest) value. For example in a cluster $\{3,3,3,4,5,6,6\}$, we take 3 as the second smallest value (rather than 4) and we take 6 as the second smallest value (rather than 5). More formally, the proposed replacements are:

- Let $I_{min_j^a} = \arg\min_{I_i\in C_j^a}\{x_i^a\}$ be one individual with the smallest value for A^a in C_j^a, and let $I_{min_j'^a} = \arg\min_{I_i\in C_j^a\setminus I_{min_j^a}}\{x_i^a\}$ be one individual with the second smallest value in C_j^a. We replace the value of $I_{min_j^a}$ for A^a by the value of $I_{min_j'^a}$, that is, $x^a_{min_j^a} = x^a_{min_j'^a}$.
- Let $I_{max_j^a} = \arg\max_{I_i\in C_j^a}\{x_i^a\}$ be one individual with the largest value for A^a in C_j^a, and let $I_{max_j'^a} = \arg\max_{I_i\in C_j^a\setminus I_{max_j^a}}\{x_i^a\}$ be one individual with the second largest

value in C_j^a. We replace the value of $I_{max_j^a}$ for A^a by the value of $I_{max'^a_j}$, that is, $x^a_{max_j^a} = x^a_{max'^a_j}$.

Let us call D' the data set that results from applying this pre-processing step to D. The purpose of the pre-processing is to make sure that modification of a single record of D keeps the values of cluster C_j^a in D' within the range $[x^a_{min_j^a}, x^a_{max_j^a}]$. Indeed, if we modify a record to a value smaller than $x^a_{min_j^a}$, resp. larger than $x^a_{max_j^a}$ (which causes the modified value to become the smallest, resp. the largest value in C_j^a) the pre-processing step will automatically replace the modified value by $x^a_{min_j^a}$, resp. $x^a_{max_j^a}$ (that is, the former smallest value, resp. largest value, which is now the second smallest value, resp. second largest value).

Composing the pre-processing step with the microaggregation can be viewed as an alternative microaggregation algorithm, which we expect to be substantially less sensitive to changes of the records in D. To attain iDP we need to adjust the noise to the sensitivity of this alternative microaggregation algorithm. Such a sensitivity is computed in the following proposition.

Proposition 9. *Let $C^a = \{C_1^a, \ldots, C_p^a\}$ be a clustering of records created w.r.t. the values of attribute A^a and assume each cluster contains at least three values (not necessarily different). Let $P^a = \{P_1^a, \ldots, P_p^a\}$ be the clustering that results from replacing the largest and the smallest values of attribute A^a in each cluster by the second largest and the second smallest value in the cluster (that is, the clusters that result from applying the previously described pre-processing step to C^a). Let $p_j^a = \frac{1}{|P_j^a|}\sum_{x\in P_j^a} x^a$, for $j = 1, \ldots, p$, be the centroid associated with cluster P_j^a, where x^a is the value of the record x for the attribute A^a. The local sensitivity for each centroid, $LS_{p_j^a}(\bar{D})$, is*

$$\max\{|x_{max_j^a} - x_{min'^a_j}| + |x_{min''^a_j} - x_{min'^a_j}| + |x_{max_j^a} - x_{max'^a_j}|,$$
$$|x_{min_j^a} - x_{max'^a_j}| + |x_{max''^a_j} - x_{max'^a_j}| + |x_{min_j^a} - x_{min'^a_j}|\} \quad (1)$$

divided by $|P_j^a|$, where $x_{min''^a_j}$ and $x_{max''^a_j}$ are, respectively, the third smallest and the third largest values in the cluster.

Proof. Let us assume the maximum possible change of a record. This occurs when the smallest value becomes greater than the largest value so far, or when the largest value becomes smaller than the smallest value so far. Let us begin with the first case.

Assume $x_{min_j^a}$ changes to a value greater than the largest value $x_{max_j^a}$. Initially, the pre-processed value was $x_{min'^a_j}$ and it becomes $x_{max_j^a}$ after the change (as $x_{max_j^a}$ is the second largest value of the cluster in the modified data set). Additionally, after pre-processing, the former second smallest value $x_{min'^a_j}$ is changed to the third smallest value $x_{min''^a_j}$ and $x_{max'^a_j}$ becomes $x_{max_j^a}$. Hence, the change in the sum of pre-processed cluster values is

$$|x_{max_j^a} - x_{min'^a_j}| + |x_{min''^a_j} - x_{min'^a_j}| + |x_{max_j^a} - x_{max'^a_j}|. \quad (2)$$

Symmetrically, if $x_{max_j^a}$ changes to a value less than the smallest value $x_{min_j^a}$, the change in the sum of pre-processed cluster values is

$$|x_{min_j^a} - x_{max'^a_j}| + |x_{max''^a_j} - x_{max'^a_j}| + |x_{min_j^a} - x_{min'^a_j}|. \quad (3)$$

Let us now consider changes in the second smallest or second largest values, which are also relevant for the range of the pre-processed cluster. If the second smallest record changes maximally, that is, becomes greater than the largest value so far, the pre-processed cluster will have values in the range $[x_{min_j''^a}, x_{max_j^a}]$. Hence, the change in the sum of pre-processed cluster values is the same as in Expression (2). Symmetrically, if the second largest value changes maximally, the change in the sum of values is the same as in Expression (3).

Thus, the maximum change in the sum of pre-processed cluster values is the maximum of Expressions (2) and (3). To obtain the sensitivity, we must divide by the cardinality $|P_j^a|$ of the cluster. This concludes the proof. □

In this way, if we add the pre-processing step to the approach proposed in Section 3.2.1, the local sensitivity of each centroid can be computed from the values in the corresponding cluster. Hence, we do not need the attributes to be bounded and we obtain smaller local sensitivities –because the cluster value range is usually narrower than the domain value range–.

By using the approach described in Section 3.2.1 on the pre-processed data set (D') and computing the local sensitivity as specified in Expression (1), we generate an ϵ-iDP data set.

4 Experimental Evaluation

This section reports the empirical evaluation of the two microaggregation strategies we propose for generating iDP data sets. Two well-known data sets have been used as evaluation data:

- *Census*: This is a test data set extracted from the U.S. Census Bureau 1995 Current Population Survey [27]. It contains 1,080 records with numerical attributes and it has been widely used for evaluating privacy-preserving methods [9, 10, 28]. The following (unbounded) attributes about finance and taxes have been used in our experiments: AFNLWGT, AGI, EMCONTRB, FEDTAX, STATETAX, TAXINC, POTHVAL, INTVAL, and FICA.
- *Wine Quality*: This is a double data set for classification/regression tasks found in UCI [29]. There is a data set related to red wines and another related to white wines. We used the white wine data set, which contains significantly more instances than the red wine data set (4,898 vs. 1,600). Attributes are discrete or numerical, and they describe physicochemical properties of wine.

All the attributes of the two data sets we used represent non-negative numerical magnitudes and most of them are unbounded. To compute the (global and local) sensitivities, we fixed the attribute domains as

$$[0, \ldots, (\alpha \times max\_attr.\_value\_in\_dataset)],$$

and took $\alpha \in \{1.5, 3.0\}$. By varying α we tested the influence of the size of the attribute domains on the sensitivity that applies for each method. Since the Laplace distribution takes values in the range $(-\infty, +\infty)$, for consistency we bounded noise-added outputs to the domain ranges defined above.

Two well-differentiated evaluation experiments were carried out. In the first one, we measured the information loss incurred by our methods w.r.t. that of standard DP with general metrics. In the second experiment, we assessed the utility retained by the masked data produced by our methods when employed in machine learning tasks.

4.1 Information Loss Evaluation

Information loss refers to the difference between the masked and the original data, that is, to the harm inflicted by masking to the analytical accuracy of the original data. The Sum of Squared Errors (SSE) is a standard measure of information loss [1], and it is defined as the sum of squares of distances between original and masked records:

$$SSE = \sum dist(r_i, (r_i)')^2, \tag{4}$$

where r_i is the i-th record in the original data set and $(r_i)'$ represents its masked version. The mean SSE, calculated by dividing the SSE by the number of records n, is usually more informative because it does not depend on the cardinality of the data set.

To compute SSE, we need a distance between records:

$$d((x_1, ..., x_m), (y_1, ..., y_m)) =$$

$$= (1/m)\sqrt{(d_1(x_1, y_1)/\sigma_1^2)^2 + \ldots + (d_m(x_m, y_m)/\sigma_m^2)^2}, \tag{5}$$

where $d_j(\cdot,\cdot)$ is the distance between values of the j-th attribute, σ_j^2 is the sample variance of the j-th attribute in the original data set and m is the number of attributes.

For differential privacy, we have considered ϵ values 0.01, 0.1 and 1.0, which cover the range of reasonably safe values [19]. In all the DP/iDP methods discussed/proposed in this paper, the sensitivity of the centroids obtained after the microaggregation step depends on the parameter k that establishes the minimum number of records in each cluster; that is, it defines the cardinality $|C|$ of the cluster associated with the centroid. In order to evaluate the influence of the microaggregation step, and considering the cardinality of the *Census* and *Wine Quality* data sets, we have taken parameter k between 3 and 100 for the former data set and between 3 and 400 for the latter; notice that $k = 3$ is the minimum microaggregation level supported by the method using cluster-based local sensitivity.

To put in context the results obtained with our methods –iDP using microaggregation and local sensitivity (*iDP-LS*) and iDP using microaggregation and cluster-based local sensitivity (*iDP-CBLS*)–, we compared them with the following alternatives:

- Plain Laplace noise addition for ϵ-differential privacy (*DP*) with no prior microaggregation, as described in Section 2. This is the naive mechanism to produce differentially private data sets. We used it as an upper bound for the information loss.

- Differential privacy using univariate microaggregation (*DP-UM*), as described in Section 3.1. This mechanism uses the prior microaggregation step to reduce the sensitivity, even though the noise applied to the centroids needs to be adjusted to the global sensitivity.

For each data set, each above-mentioned method and each method parameterization (in terms of α, k and ϵ values), we made 10 runs and computed the mean SSE over them. The result is depicted in Figures 3 and 4. Notice that the plain Laplace noise addition method (*DP*) is displayed as a horizontal line, because it entails no microaggregation and is hence independent of k. Also, since SSE values are quite diverse among the different methods, a $\log_{10}$ scale has been used in the ordinates.

The results obtained for the two data sets show that:

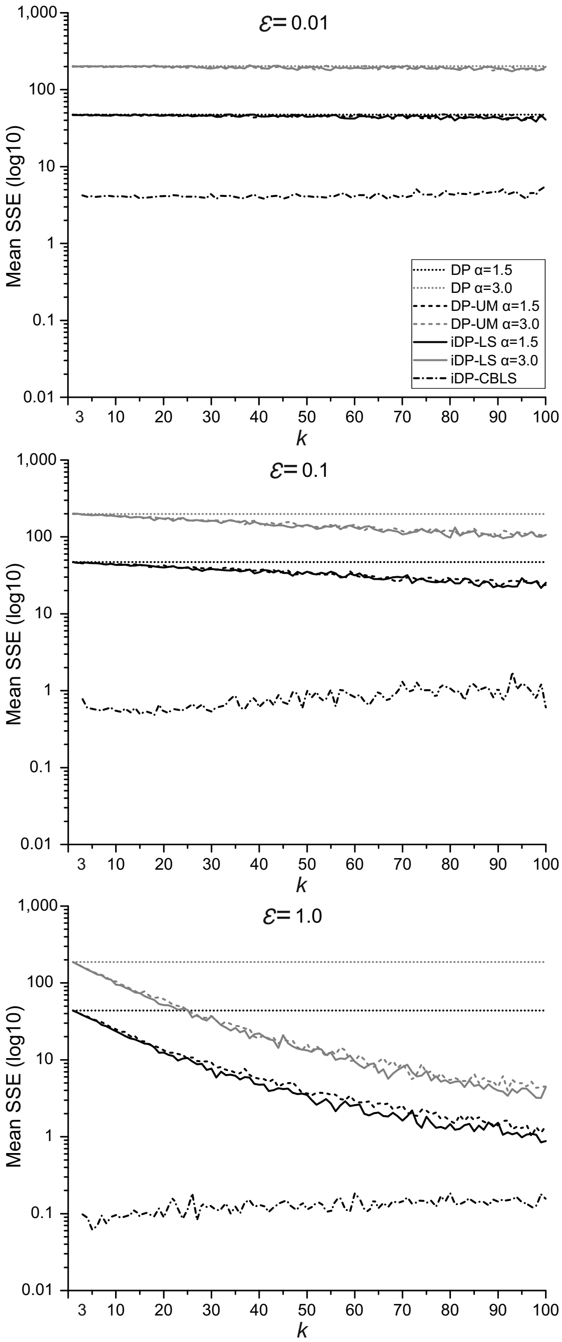

Figure 3: *Census* data set: mean SSE for the proposed methods (*iDP-LS* and *iDP-CBLS*) and baselines (*DP-UM* and *DP*) with $\epsilon = 0.01$ (top), $\epsilon = 0.1$ (center) and $\epsilon = 1.0$ (bottom), and $\alpha \in \{1.5, 3.0\}$, for microaggregation parameter k from 3 to 100

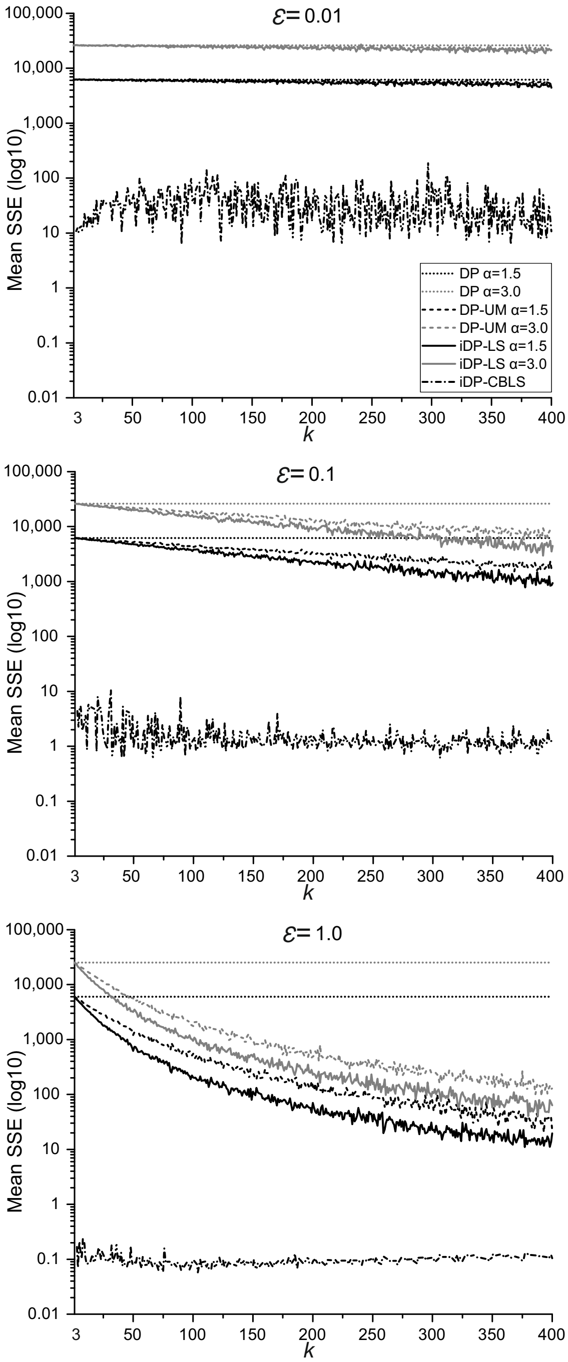

Figure 4: *Wine Quality* data set: mean SSE for the proposed methods (*iDP-LS* and *iDP-CBLS*) and baselines (*DP-UM* and *DP*) with $\epsilon = 0.01$ (top), $\epsilon = 0.1$ (center) and $\epsilon = 1.0$ (bottom), and $\alpha \in \{1.5, 3.0\}$, for microaggregation parameter k from 3 to 400

- Plain Laplace noise addition (*DP*), with no prior microaggregation, results in the highest SSE due to the large global sensitivity. In fact, SSE barely decreases when moving from $\epsilon = 0.01$ to $\epsilon = 1.0$ and, thus, we may consider the masked data nearly random in all cases.

- Both *DP-UM* and *iDP-LS* yield similar results, with SSE decreasing as the microaggregation level (k) increases. This shows the benefits of the prior microaggregation step to reduce the sensitivity. *iDP-LS* achieves lower SSE thanks to its ability to use the local sensitivity instead of the global sensitivity employed by *DP-UM*. The differences between the two methods are larger for higher values of ϵ, but stay proportional when varying the boundaries of the attribute domains (α). In this respect, even though *iDP-LS* uses the local sensitivity, it still depends on the domain boundaries, as stated in Proposition 8.

- For the three previous methods, SSE values decrease as the attribute domains get smaller (smaller α). This illustrates that large (or even unbounded) domains severely deteriorate the utility of the masked data when using the straightforward approach to generate differentially private data sets.

- Our most sophisticated strategy (*iDP-CBLS*) is able to improve the accuracy of the previous methods by several orders of magnitude. In fact, its SSE for $\epsilon = 0.01$ is similar to the ones of *DP-UM* and *iDP-LS* for $\epsilon = 1.0$ and large k, which shows that *iDP-CBLS* can accommodate stronger privacy requirements (smaller ϵ). Moreover, since the sensitivity calculation is based on clusters rather than on attribute domains (see Proposition 9), it is not affected by (large) domains. Also, unlike the former methods, *iDP-CBLS* requires very small microaggregation levels (k among 5-15) to obtain optimal results. Larger microaggregation levels produce a small SSE increase because the distortion caused by microaggregation is greater (for large k) than the benefits resulting from reducing the sensitivity (the cluster-based sensitivity is already quite small).

4.2 Evaluation in Data Classification

The second set of experiments aims at evaluating the utility retained by the masked outcomes in a specific machine learning task, namely data classification. This scenario is especially relevant because DP has also been adopted as the *de facto* standard for privacy protection in machine learning and, like in data releases, researchers struggle to reconcile meaningful DP guarantees with usable model accuracy [30].

We built a classification model using masked training data, and we compared its classification accuracy with that of a model built on original training data. The same original data were used in both cases as evaluation data to measure the classification accuracy. For both the *Census* and the *Wine Quality* data sets, we used the first 66% of (masked, resp. original) records for training and the rest of (original) records for evaluation. As in the former experiments, we report here the average result of 10 runs for the same parameter values (ϵ, k and α) and for the two methods we propose: *iDP-LS* and *iDP-CBLS*.

The classifier we chose is Random Forest, which is fast and easy to implement, produces highly accurate predictions and can handle a very large number of input variables without overfitting [31].

To measure the classification accuracy, we focused on the F-measure of the class attribute, which is the harmonic mean between precision and recall.

4.2.1 *Census* data set

Since *Census* is a general purpose data set (not specifically aimed at classification), we had to adopt ERNVAL (business or farm net earnings in the year of reference) as class attribute by categorizing it into "≤30K" and ">30K" if the balance is less/equal than 30,000 and greater than 30,000, respectively. Thus, the classification objective is to predict whether an individual obtains earnings below or above 30,000.

Fig. 5 depicts the F-measure for both classes with original training data and with the masked training data produced by our methods for the different parameters. In general, the results are coherent with the SSE figures reported above. On the one hand, the results of *iDP-LS* improve proportionally to the microaggregation level k and are better for narrower attribute domains (which result in a smaller sensitivity) with $\epsilon > 0.01$. For such values of ϵ, F-measures remain quite stable from $k = 50$ onwards, and the best results are around 20%, 10% and 5% worse for $\epsilon = 0.01$, $\epsilon = 0.1$ and $\epsilon = 1.0$, respectively, than the upper bound defined by the original training data. On the other hand, *iDP-CBLS* provides much better results, that for $\epsilon = 1.0$ are identical to the upper bound, and for $\epsilon = 0.1$ and $\epsilon = 0.01$ only 3% and 10% worse, respectively. In this scenario, the utility retained by *iDP-CBLS* for $\epsilon = 0.01$ is similar to that of *iDP-LS* for $\epsilon = 0.1$. As we also observed in the former experiments, F-measures for *iDP-CBLS* degrade for large k because the distortion added by the microaggregation step becomes comparatively larger than the reduction of sensitivity (and thus than the reduction of the noise, which is already very small).

4.2.2 *Wine Quality* data set

In the *Wine Quality* data set, the class attribute is QUALITY, ranging from 0 to 10, where 0 is the worst quality score and 10 the best. To simplify the classification, we considered two classes: "Not Excellent" and "Excellent", which correspond to QUALITY values ≤ 6 or > 6 respectively.

Fig. 6 depicts the F-measures for these two classes with the same methods and parameters as above. The tendencies observed in the results are similar to those for *Census*, but with some differences. On the one hand, the F-measures for the two classes are significantly different because the classes are unbalanced: "Not Excellent" has 3.5 times more records than "Excellent"; thus, the latter class is more difficult to classify due to the smaller amount of training data, and results in lower F-measures. On the other hand, *iDP-CBLS* for $\epsilon = 0.1$ and $\epsilon = 1.0$ was able not only to reach the upper bound, but also to slightly improve it for some values of k. This phenomenon can be explained by the fact that adding a small amount of noise to the training data, as is the case for *iDP-CBLS*, may improve the classification accuracy as long as the distribution of the data remains similar [32]. Thus, we can observe that *iDP-CBLS* was not only able to maintain a low SSE, but the (small) noise it added had even a positive effect on data classification in some cases.

5 Conclusions and Future Research

The strong privacy guarantees of DP only hold for small values of ϵ, which usually result in limited data accuracy in non-interactive settings, such as data releases and machine learning [12].

To tackle this problem, in this paper we have leveraged the (potential) advantages of iDP regarding data utility with the sensitivity (and noise) reduction enabled by microaggregation-

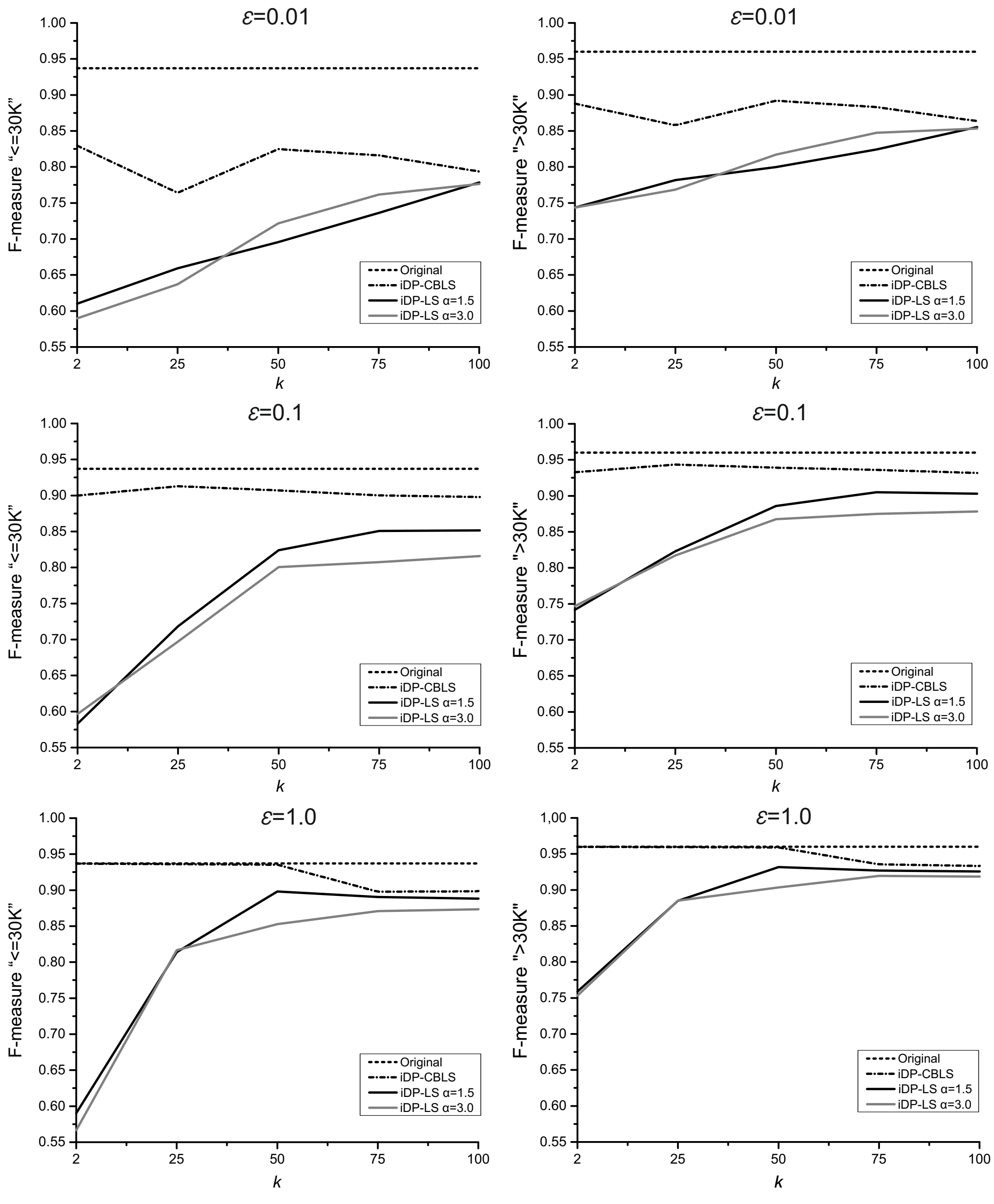

Figure 5: *Census* data set: F-measures for classes "≤30K" (left) and ">30K" (right) with original training data and masked training data for different parameters

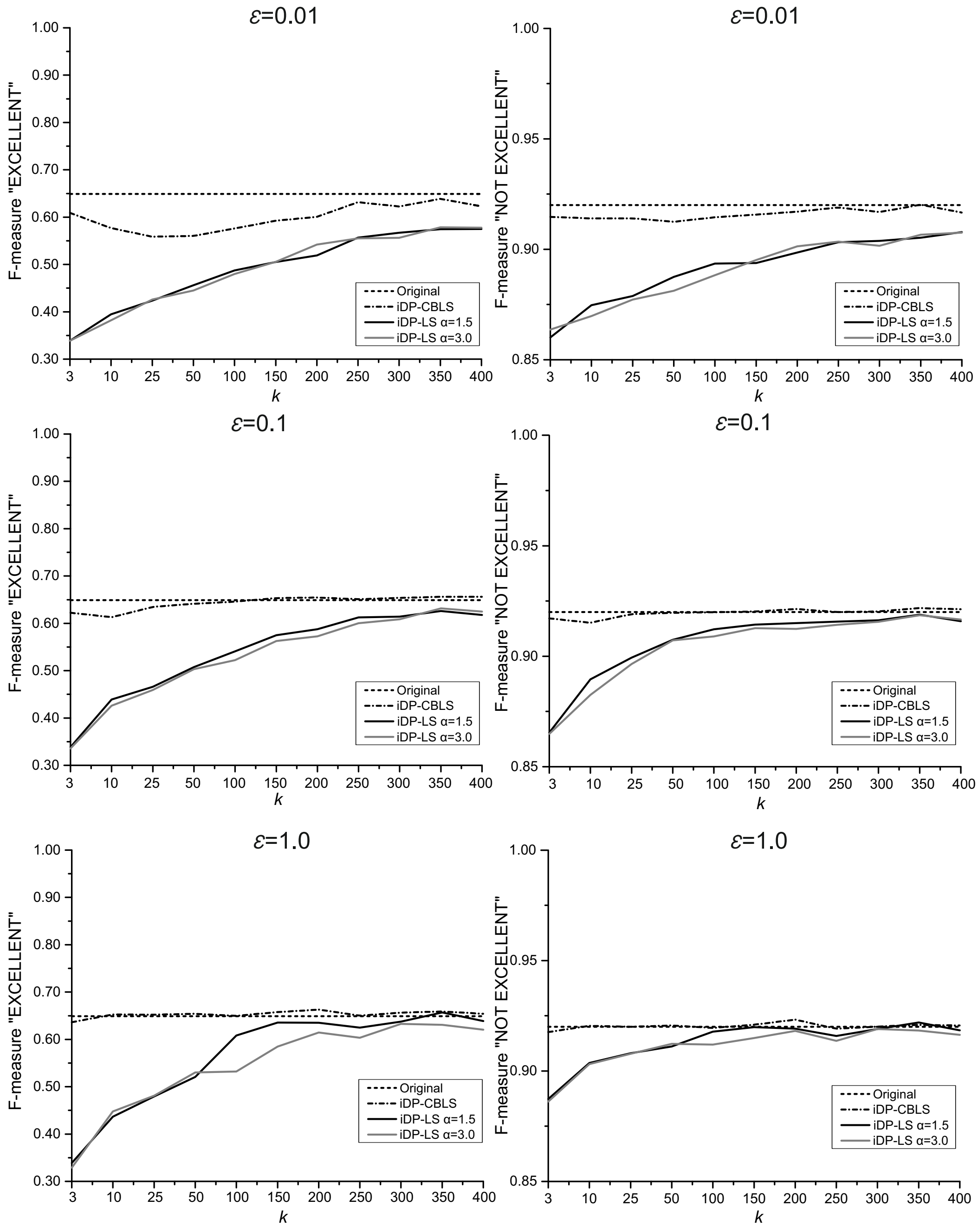

Figure 6: *Wine Quality* data set: F-measures for classes "Excellent" (left) and "Not Excellent" (right) with original training data and masked training data for different parameters

based DP masking. Two microaggregation strategies have been proposed, of which the second (that computes the local sensitivity based on clusters) is the one that brings the greatest utility benefits. In fact, with *iDP-CBLS* we were able to improve the utility of the masked outcomes for a given ϵ by several orders of magnitude in comparison with plain DP data sets. As shown in the experiments, for a given ϵ, the utility preserved by *iDP-CBLS* rivals the one preserved by DP data sets with 10-100 times larger ϵ values. Therefore, our method is able to reconcile the small values of ϵ (say $\epsilon < 0.1$) that are needed for DP to offer real privacy, with the data accuracy that is needed for data releases to be useful. Also, for larger ϵ values, *iDP-CBLS* was not only able to keep the noise low, but also to reach the accuracy upper bound for classification tasks (corresponding to the case in which original data are used for training). These results are significantly better than those obtained when applying standard DP in machine learning tasks, even with much larger ϵ [30].

As future work, we plan to design iDP-enforcing mechanisms (other than noise addition) for non-numerical discrete attributes, which are commonly found in microdata sets. We also plan to explore the use of multivariate microaggregation instead of individual-ranking microaggregation. Although multivariate microaggregation might cause more information loss, it might also guarantee that the protection offered by *iDP-CBLS* can be no less than the one offered by k-anonymity, no matter the value of ϵ chosen.

Acknowledgments and disclaimer

This research was funded by the European Commission (project H2020-871042 "SoBigData++"), the Government of Catalonia (ICREA Acadèmia Prizes to J.Domingo-Ferrer and to D. Sánchez), MCIN/AEI/ 10.13039/501100011033 and "ERDF A way of making Europe" under grant PID2021-123637NB-I00 "CURLING" and INCIBE-NextGenerationEU (project "HERMES" and INCIBE-URV cybersecurity chair).

References

[1] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E.S. Nordholt, K. Spicer, P.P. de Wolf, Statistical Disclosure Control, Wiley, 2012.

[2] J. Soria-Comas, J. Domingo-Ferrer, 2015, Big data privacy: challenges to privacy principles and models, Data Science and Engineering 1(1)1–8.

[3] C. Dwork, F. McSherry, K. Nissim, A.D. Smith, Calibrating noise to sensitivity in private data analysis, in: 3rd Theory of Cryptography Conference, Springer, 2006, pp. 265–284.

[4] P. Samarati, 2001, Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering 13(6)1010–1027.

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

[6] N. Li, T. Li, S. Venkatasubramanian, t-closeness: privacy beyond k-anonymity and l-diversity, in: Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE), 2007, pp. 106–115.

[7] J. Zhang, G. Cormode, C.M. Procopiuc, D. Srivastava, X. Xiao, Privbayes: private data release via Bayesian networks, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 2014, pp. 1423–1434.

[8] Y. Xiao, L. Xiong, C. Yuan, Differentially private data release through multidimensional partitioning, in: Secure Data Management (SDM), Springer, 2010, pp. 150–168.

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

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

[11] Z. Chu, J. He, J. Li, Q. Wang, X. Zhang, N. Zhu, 2023, SSKM_DP: Differential privacy data publishing method via SFLA-Kohonen network, Applied Sciences 13(6)3823.

[12] J. Domingo-Ferrer, D. Sánchez, A. Blanco-Justicia, 2021, The limits of differential privacy (and its misuse in data release and machine learning), Communications of the ACM 64(7)33–35.

[13] J.M. Abowd, M.B. Hawes, Confidentiality protection in the 2020 U.S. Census of Population and Housing, in: Annual Review of Statistics and Its Application, 2023, pp. 119–144.

[14] U.S. Census Bureau, Privacy-loss Budget Allocation, 2022, https://www2.census.gov/programs-surveys/decennial/2020/program-management/data-product-planning/2010-demonstration-data-products/02-Demographic_and_Housing_Characteristics/2022-03-16_Summary_File/2022-03-16_Privacy-Loss_Budget_Allocations.pdf (accessed 6 October 2023).

[15] C. Kenny, et al, 2021, The use of differential privacy for census data and its impact on redistricting: The case of the 2020 U.S. Census, Science Advances 7(41)eabk3283.

[16] C. Dwork, N. Kohli, M. Mulligan, 2019, Differential privacy in practice: expose your epsilons!, Journal of Privacy and Confidentiality 9(2).

[17] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, D. Megías, 2017, Individual differential privacy: a utility-preserving formulation of differential privacy guarantees, IEEE Transactions on Information Forensics and Security 12(6)1418–1429.

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

[19] C. Dwork, 2011, A firm foundation for private data analysis, Communications of the ACM 54(1)86–95.

[20] C. Dwork, Differential privacy, in: Automata, Languages and Programming (ICALP), Springer, 2006, pp. 1–12.

[21] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, N. Naor, Our data, ourselves: privacy via distributed noise generation, in: Advances in Cryptology - EUROCRYPT, Springer, 2006, pp. 486–503.

[22] C. Dwork, G.N. Rothblum, Concentrated differential privacy, CoRR abs/1603.01887, 2016, http://arxiv.org/abs/1603.01887

[23] M. Bun, T. Steinke, Concentrated differential privacy: Simplifications, extensions, and lower bounds, in: Theory of Cryptography Conference, Springer, 2016, pp. 635–658.

[24] I. Mironov, Rényi differential privacy, in: IEEE 30th Computer Security Foundations Symposium (CSF), IEEE, 2017, pp. 263–275.

[25] K. Nissim, S. Raskhodnikova, A. Smith, Smooth sensitivity and sampling in private data analysis, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007, pp. 75–84.

[26] J. Soria-Comas, J. Domingo-Ferrer, Differentially private data sets based on microaggregation and record perturbation, in: Modeling Decision for Artificial Intelligence (MDAI), Springer, 2017, pp. 119–131.

[27] R. Brand, J. Domingo-Ferrer, J.M. Mateo-Sanz, Reference data sets to test and compare SDC methods for protection of numerical microdata, CASC Project Deliverable, 2002, https://research.cbs.nl/casc/CASCrefmicrodata.pdf/ (accessed 8 October 2023).

[28] M. Laszlo, S. Mukherjee, 2005, Minimum spanning tree partitioning algorithm for microaggre-

gation, IEEE Transactions on Knowledge and Data Engineering 17(7)902–911.

[29] Wine Quality Data Set, UCI Machine Learning Repository, 2009, https://archive.ics.uci.edu/ml/datasets/wine+quality/ (accessed 6 October 2023).

[30] A. Blanco-Justicia, D. Sánchez, J. Domingo-Ferrer, K. Muralidhar, 2023, A critical review on the use (and misuse) of differential privacy in machine learning, ACM Computing Surveys 55(8)1–26.

[31] G. Biau, 2012, Analysis of a random forests model, Journal of Machine Learning Research 13(1)1063–1095.

[32] L. Maaten, M. Chen, S. Tyree, K. Weinberger, Learning with marginalized corrupted features, in: Proceedings of 30th International Conference on Machine Learning (ICML), 2013, pp. 410–418.