DIFFERENTIALLY PRIVATE COMPRESSIVE K-MEANS

V. Schellekens [□], *A. Chatalic* [△], *F. Houssiau* [◊], *Y.-A. de Montjoye* [◊], *L. Jacques* [□], *R. Gribonval* [△]

[□] ICTEAM/ELEN, UCLouvain [△] Univ Rennes, Inria, CNRS, IRISA [◊] Imperial College London

ABSTRACT

This work addresses the problem of learning from large collections of data with privacy guarantees. The sketched learning framework proposes to deal with the large scale of datasets by compressing them into a single vector of generalized random moments, from which the learning task is then performed. We modify the standard sketching mechanism to provide differential privacy, using addition of Laplace noise combined with a subsampling mechanism (each moment is computed from a subset of the dataset). The data can be divided between several sensors, each applying the privacy-preserving mechanism locally, yielding a differentially-private sketch of the whole dataset when reunited. We apply this framework to the k-means clustering problem, for which a measure of utility of the mechanism in terms of a signal-to-noise ratio is provided, and discuss the obtained privacy-utility tradeoff.

***Index Terms*—** Differential Privacy, Sketching, Sketched Learning, Compressive Learning, Clustering.

1. INTRODUCTION

In the last few decades, the size and availability of datasets has increased exponentially. While this data promises serious advances, its sheer size represents a challenge and calls for new machine learning methods able to process large datasets efficiently, both in time and memory. In addition to tractability issues, using and publishing this data raises serious privacy concerns, and there is a need for machine learning algorithms that guarantee the privacy of users.

In the compressive learning framework [1], the dataset is compressed into a vector of generalized random moments (sometimes referred to as the sketch), from which the learning phase can then be performed with greatly reduced computational resources. In this paper, we propose a mechanism based on this approach to learn from compressed datasets with provable privacy guarantees, by computing noisy moments from dataset subsamples. This mechanism is shown to provide differential privacy [2], a popular privacy definition introduced by Dwork et al. Informally, it ensures that the output of a machine learning algorithm does not depend on the presence of one individual in the dataset.

Our method operates in a distributed context, where the dataset is shared across multiple devices, each producing and releasing publicly, once and for all, a scrambled sketch from the data it has access to. This can be done on the fly, and with low memory consumption. The sketches from each device are then averaged into one global scrambled sketch by a central analyst, who uses it later for a machine learning task. The learning task is formulated as an inverse problem as in traditional compressive learning. While this paper focuses on k-means clustering, our method is general and can easily be applied to other learning tasks such as Gaussian mixture modeling or principal component analysis. In the case of clustering, we show that the utility of a noisy sketch, i.e. its quality for subsequent learning, can be measured by a signal-to-noise ratio; we use this quantity to explain the tradeoff existing between privacy and utility and motivate our choice of parameters.

We give some background on k-means clustering, compressive learning and differential privacy in Section 2, and introduce in Section 3 our attack model and data release mechanism. Privacy guarantees and utility measures are provided respectively in Sections 4 and 5, the tradeoff between the two is discussed in Section 6 and related works are presented in Section 7.

2. NOTATIONS AND BACKGROUND

The general goal of (unsupervised) machine learning tasks is to infer parameters of a mathematical model from a dataset $\mathcal{X} \triangleq \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ of n learning examples. In this paper, we focus on one such task to illustrate our method's capabilities: the k-means problem, that seeks, for a dataset $\mathcal{X}$, to find k cluster centroids $\mathcal{C} \triangleq \{\boldsymbol{c}_j \in \mathbb{R}^d\}_{j=1}^k$ minimizing the sum of squared errors (SSE):

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \ \mathrm{SSE}(\mathcal{X}, \mathcal{C}) \triangleq \arg\min_{\mathcal{C}} \sum_{\boldsymbol{x}_i \in \mathcal{X}} \min_{\boldsymbol{c}_j \in \mathcal{C}} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2^2. \quad (1)$$

As solving (1) exactly is NP-hard [3], a variety of heuristics have been proposed [4], the most widely used being Lloyds' k-means algorithm [5]. These methods typically require multiple passes on the entire dataset $\mathcal{X}$, which becomes prohibitive in time and memory when n is large. Compressive learning [1] bypasses this shortcoming by first compressing $\mathcal{X}$ into a single vector $\boldsymbol{z}_\mathcal{X} \in \mathbb{C}^m$ (called the *sketch*) before performing the learning task. The sketch is defined as the sample average of random Fourier features [6], i.e.

$$\boldsymbol{z}_\mathcal{X} = \tfrac{1}{n} \textstyle\sum_{\boldsymbol{x}_i \in \mathcal{X}} \boldsymbol{z}_{\boldsymbol{x}_i} \quad \text{with} \quad \boldsymbol{z}_{\boldsymbol{x}} \triangleq \tfrac{1}{\sqrt{m}} \exp\left(\mathrm{i}\Omega^T \boldsymbol{x}\right) \in \mathbb{C}^m, \quad (2)$$

where m is the sketch dimension and $\Omega = [\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m]$ is a matrix composed of m frequency vectors $\boldsymbol{\omega}_j \in \mathbb{R}^d$, generated randomly according to a specific distribution (see [7] for details). The most promising advantage of compressive learning is that the sketch size m required to learn models from $\boldsymbol{z}_\mathcal{X}$ does not depend on the dataset size n. For instance, it was shown that the compressive k-means method [8] achieves an SSE close to the one obtained with Lloyd's k-means provided $m = \Omega(dk)$, i.e. provided that m is of the order of the number of parameters to learn. Moreover, the sketch can be computed in one pass, in parallel, and is easy to update when additional data is available. Finally, when n is large, the impact of one single sample $\boldsymbol{x}_i$ on $\boldsymbol{z}_\mathcal{X}$ is small, due to the aggregation, leading us to believe that sketching methods are good candidates to build privacy-preserving methods.

Differential Privacy (DP) is a powerful privacy definition, introduced by Dwork et al. [2], that is widely recognized as a standard and robust definition. Intuitively, differential privacy ensures that the result of an algorithm running on private data does not depend significantly on the contribution of one specific person or record in

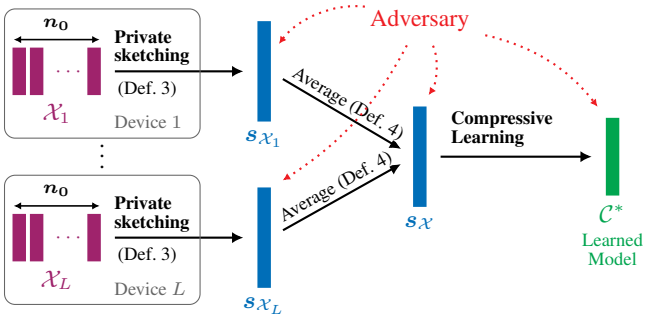

Fig. 1. Our attack model: L devices should protect the privacy of their local datasets $(\mathcal{X}_l)_{1\leq l\leq L}$ while allowing an algorithm to learn a model from it (in our case, the centroids $\mathcal{C}^*$). The (public) matrix of frequencies Ω is used for both "private sketching" and "compressive learning". All devices publish their scrambled sketches $\boldsymbol{s}_{\mathcal{X}_l}$, which are combined into the global sketch $\boldsymbol{s}_{\mathcal{X}}$.

the dataset. The definition of differential privacy involves a class of datasets and a neighboring relation over datasets. We denote $\mathcal{D}$ the set of all datasets in $\mathbb{R}^d$, and we define two datasets as neighbors if they differ by at most one record, i.e. $\mathcal{X} \sim \mathcal{X}' \Leftrightarrow (|\mathcal{X}| = |\mathcal{X}'|$ and $|(\mathcal{X} \cup \mathcal{X}') \setminus (\mathcal{X} \cap \mathcal{X}')| \leq 2)$.

Definition 1 (Differential Privacy)**.** *Given a neighboring relation $\sim$ between datasets in $\mathcal{D}$, a randomized algorithm f is said to achieve differential privacy with privacy parameter ϵ (noted $\epsilon-DP$) if for any measurable set S of the co-domain of f:*

$$\forall \mathcal{X}, \mathcal{X}' \in \mathcal{D} \text{ s.t. } \mathcal{X} \sim \mathcal{X}' : \ \mathbb{P}\left[f(\mathcal{X}) \in S\right] \leq e^{\epsilon} \mathbb{P}\left[f(\mathcal{X}') \in S\right]. \tag{3}$$

Differential privacy is widely recognized as a strong and particularly conservative definition of privacy. By definition, it protects the privacy of a record even against an adversary who knows all other records in the dataset. Furthermore, it has been shown to be resistant to many classes of attacks [9]. Formally, our definition provides *event-level DP* [10], which is equivalent to user-level if each person has only one record in the dataset. However, implementing differentially private mechanisms for general purpose analytics is still an open challenge [11]. Although some statistics cannot be computed anymore from the (scrambled) sketch, our mechanism still allows to solve multiple learning tasks [1].

3. ATTACK MODEL AND PRIVACY MECHANISMS

In our setup, represented in Figure 1, the dataset $\mathcal{X}$ is distributed across L devices (e.g. a personal computer or smartphone, a server, or a sensor). Each device l locally stores a dataset $\mathcal{X}_l$ (disjoint of all others), assumed to contain $|\mathcal{X}_l| = n_0$ learning examples each, for a total number of records of $n \triangleq |\mathcal{X}| = L\, n_0$. Each of these devices is trusted, in that it holds data in clear. Each device compresses its data, and publishes its scrambled sketch to a central agent, that will compute the total sketch and perform computations on it. Once the local sketch has been computed, the device can delete the data used. In our attack model, an attacker is able to observe all scrambled sketches, but cannot interact further with the devices. Additionally, the sketching mechanism is not kept secret, i.e. the adversary knows the random matrix of frequencies Ω, the sketching parameters (α_r and σ_ξ, introduced in Definition 3), and the dataset shape (d and n_0). For simplicity, in our analysis we assume that all datasets have the same size, but in practice each device can actually use the mechanism with a different value of n_0, corresponding to the size of its local dataset. Our method also works if the dataset is not distributed, i.e. if $L = 1$ (in which case the data holder can also be the analyst, who only publishes the result of the computation $\mathcal{C}^*$).

Compressing the local dataset $\mathcal{X}_l$ to the sketch $\boldsymbol{z}_{\mathcal{X}_l}$ reduces the transmission costs and computational resources required during learning, but cannot guarantee differential privacy as such (consider for instance the extreme case $n_0 = 1$). Instead, the devices thus publish a distorted, scrambled sketch $\boldsymbol{s}_{\mathcal{X}_l}$, formally described hereafter. Our construction of $\boldsymbol{s}_{\mathcal{X}_l}$ relies on the Laplace mechanism (a strategy to make an algorithm differentially private by adding Laplacian noise on its output [12]), that we combine with random subsampling. The sketch being complex, we first introduce a complex Laplace distribution from which the noise will be drawn.

Definition 2. *A random variable z follows a complex Laplace distribution of parameter β (denoted $z \sim \mathcal{L}^{\mathbb{C}}(\beta)$) iff its real and imaginary parts follow independently a real Laplace distribution of parameter $\beta/\sqrt{2}$. In that case, z admits a density $p_z(z) \propto \exp(-(|\Re z| + |\Im z|)\sqrt{2}/\beta)$ and $\sigma_z^2 = 2\beta^2$.*

We now define, for a standard deviation $\sigma_\xi \in \mathbb{R}^+$ and a number of measurements r, the local and global scrambled sketches.

Definition 3 (Local sketching mechanism)**.** *The scrambled sketch $\boldsymbol{s}_{\mathcal{X}_l}$ of a dataset $\mathcal{X}_l = \{\boldsymbol{x}_i\}_{i=1}^{n_0}$ located on one device, using $r \in [\![1, m]\!] \triangleq \{1, \cdots, m\}$ measurements per record and a noise standard deviation σ_ξ, is defined as*

$$\boldsymbol{s}_{\mathcal{X}_l} \triangleq \frac{1}{\alpha_r n_0} \sum_{i=1}^{n_0} (\boldsymbol{z}_{\boldsymbol{x}_i} \odot \boldsymbol{b}_{\boldsymbol{x}_i}) + \frac{\boldsymbol{\xi}}{\sqrt{\alpha_r m n_0}}, \tag{4}$$

where $\boldsymbol{b}_{\boldsymbol{x}_i} \overset{iid}{\sim} \mathcal{U}(\mathcal{B}_r)$ for all $i \in [\![1, n_0]\!]$, $\xi_j \overset{iid}{\sim} \mathcal{L}^{\mathbb{C}}\left(\sigma_\xi/\sqrt{2}\right)$ for all $j \in [\![1, m]\!]$, $\mathcal{B}_r \triangleq \{\boldsymbol{b} \in \{0,1\}^m | \sum_{j=1}^m b_j = r\}$ is the set of binary masks $\boldsymbol{b}$ selecting exactly r coefficients, $\alpha_r \triangleq r/m$ is called the subsampling parameter*, and $\odot$ is the pointwise multiplication.*

Normalizing by $\alpha_r n_0$ is needed to ensure $\mathbb{E}_{\boldsymbol{b},\boldsymbol{\xi}}[\boldsymbol{s}_{\mathcal{X}_l}] = \boldsymbol{z}_{\mathcal{X}_l}$. The noise is therefore normalized by $\sqrt{\alpha_r n_0}$ for coherence, and also by $\sqrt{m}$ to be consistent with the individual sketches defined in (2).

Intuitively, masking records improves the privacy of the mechanism given that each record contributes to only r components instead of m. This is however not sufficient to ensure differential privacy: knowing all but one record still allows an attacker to isolate a record's contribution, which justifies the addition of noise in (4). In practice, masking also helps to drastically reduce the sketching computational time. We now explain how to merge the different sketches to compute the global scrambled sketch, which is used for learning.

Definition 4 (Global sketching mechanism)**.** *If $\mathcal{X}_1, \ldots, \mathcal{X}_L$ are L local datasets of n_0 samples each, stored on different devices, the global scrambled sketch of $\mathcal{X} = \cup_{l=1}^L \mathcal{X}_l$ is defined as follows:*

$$\boldsymbol{s}_{\mathcal{X}} \triangleq \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{s}_{\mathcal{X}_l}. \tag{5}$$

Note that if the local datasets have different sizes, each device can additionally release its dataset size, so that we can compute the mean sketch with the correct ponderations. Releasing the size does not affect differential privacy (if $\mathcal{X} \sim \mathcal{X}'$, then $\mathcal{X}$ and $\mathcal{X}'$ have the same size). For the rest of the paper, we stick to our simplified scenario for easier readability, i.e. all datasets have n_0 samples.

Note that if the devices can communicate privately (e.g., using cryptography), we can also consider making Ω private. In this setting, sketches could be sent privately and noise added after averaging, which would reduce the amount of noise needed for privacy.

4. ACHIEVING DIFFERENTIAL PRIVACY

We now show formally that the proposed local and global mechanisms to compute scrambled sketches are differentially private.

Proposition 1. *The local sketching mechanism given in Definition 3 with r measurements per input sample and noise standard deviation $\sigma_\xi = \frac{4\sqrt{2\alpha_r m}}{\sqrt{n_0}\epsilon}$, where $\alpha_r \triangleq r/m$, achieves ϵ-DP (Definition 1).*

Proof. Let $\mathcal{X}$ and $\mathcal{X}'$ be two datasets such that $\mathcal{X} \sim \mathcal{X}'$, that we rewrite as $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{n_0-1} \cup \{\boldsymbol{x}\}$ and $\mathcal{X}' = \{\boldsymbol{x}_i\}_{i=1}^{n_0-1} \cup \{\boldsymbol{x}'\}$.

The scrambled sketch $\boldsymbol{s}_\mathcal{X}$ of $\mathcal{X}$ can be written

$$\boldsymbol{s}_\mathcal{X} = \tfrac{1}{\alpha_r n_0} \textstyle\sum_{i=1}^{n_0-1} (\boldsymbol{z}_{\boldsymbol{x}_i} \odot \boldsymbol{b}_{\boldsymbol{x}_i}) + \tfrac{1}{\alpha_r n_0} (\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b}_{\boldsymbol{x}}) + \boldsymbol{\xi}',$$

where $\boldsymbol{\xi}' \triangleq (\alpha_r m n_0)^{-1/2} \boldsymbol{\xi}$ is the rescaled noise. When conditioning on $B \triangleq \{\boldsymbol{b}_{\boldsymbol{x}_i}\}_{i=1}^{n_0-1} \cup \{\boldsymbol{b}_{\boldsymbol{x}}\}$, the sketch $\boldsymbol{s}_\mathcal{X}$ admits the density:

$$p_{\boldsymbol{s}_\mathcal{X}}(\boldsymbol{s}|B) = p_{\boldsymbol{\xi}'}\Big(\boldsymbol{s} - \tfrac{1}{\alpha_r n_0} \textstyle\sum_{i=1}^{n_0-1} (\boldsymbol{z}_{\boldsymbol{x}_i} \odot \boldsymbol{b}_{\boldsymbol{x}_i}) - \tfrac{1}{\alpha_r n_0} (\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b}_{\boldsymbol{x}})\Big)$$
$$\propto \exp\Big(-\tfrac{2}{\sigma_{\xi'}} \Big[\Big\|\Re\Big(\tilde{\boldsymbol{s}}_B - \tfrac{\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b}_{\boldsymbol{x}}}{\alpha_r n_0}\Big)\Big\|_1 + \Big\|\Im\Big(\tilde{\boldsymbol{s}}_B - \tfrac{\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b}_{\boldsymbol{x}}}{\alpha_r n_0}\Big)\Big\|_1\Big]\Big).$$

where $\tilde{\boldsymbol{s}}_B \triangleq \boldsymbol{s} - \frac{1}{\alpha_r n_0} \sum_{i=1}^{n_0-1} (\boldsymbol{z}_{\boldsymbol{x}_i} \odot \boldsymbol{b}_{\boldsymbol{x}_i})$.

Now for a set of masks B, we denote $B' = \{\boldsymbol{b}'_{\boldsymbol{x}_i}\}_{i=1}^{n_0-1} \cup \{\boldsymbol{b}'_{\boldsymbol{x}'}\}$ the equivalent set of masks for $\mathcal{X}'$, i.e. such that $\boldsymbol{b}'_{\boldsymbol{x}_i} = \boldsymbol{b}_{\boldsymbol{x}_i}$ for all samples $\boldsymbol{x}_i$ appearing in both datasets, and $\boldsymbol{b}'_{\boldsymbol{x}'} = \boldsymbol{b}_{\boldsymbol{x}}$. We get

$$\frac{p_{\boldsymbol{s}_\mathcal{X}}(\boldsymbol{s}|B)}{p_{\boldsymbol{s}_{\mathcal{X}'}}(\boldsymbol{s}|B')} = \exp\Big(\tfrac{2}{\sigma_{\xi'}} \Big[\Big\|\Re\Big(\tilde{\boldsymbol{s}}_B - \tfrac{\boldsymbol{z}_{\boldsymbol{x}'} \odot \boldsymbol{b}_{\boldsymbol{x}}}{\alpha_r n_0}\Big)\Big\|_1 - \Big\|\Re\Big(\tilde{\boldsymbol{s}}_B - \tfrac{\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b}_{\boldsymbol{x}}}{\alpha_r n_0}\Big)\Big\|_1$$
$$+ \Big\|\Im\Big(\tilde{\boldsymbol{s}}_B - \tfrac{\boldsymbol{z}_{\boldsymbol{x}'} \odot \boldsymbol{b}_{\boldsymbol{x}}}{\alpha_r n_0}\Big)\Big\|_1 - \Big\|\Im\Big(\tilde{\boldsymbol{s}}_B - \tfrac{\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b}_{\boldsymbol{x}}}{\alpha_r n_0}\Big)\Big\|_1\Big]\Big)$$
$$\overset{(i)}{\leq} \exp\Big(\tfrac{2}{\alpha_r n_0 \sigma_{\xi'}} \big[\|\Re(\boldsymbol{z}_{\boldsymbol{x}'} - \boldsymbol{z}_{\boldsymbol{x}}) \odot \boldsymbol{b}_{\boldsymbol{x}}\|_1 + \|\Im(\boldsymbol{z}_{\boldsymbol{x}'} - \boldsymbol{z}_{\boldsymbol{x}}) \odot \boldsymbol{b}_{\boldsymbol{x}}\|_1\big]\Big)$$
$$\overset{(ii)}{\leq} \exp\Big(\tfrac{2}{\alpha_r n_0 \sigma_{\xi'}} \cdot r \cdot \tfrac{2\sqrt{2}}{\sqrt{m}}\Big) \overset{(iii)}{=} \exp\Big(\tfrac{1}{\sigma_\xi} \tfrac{4\sqrt{2\alpha_r m}}{\sqrt{n_0}}\Big) = \exp(\epsilon),$$

where (i) comes from the triangle inequality, (ii) from $\sum_{i=1}^m b_i = r$, the normalization of the sketches by $1/\sqrt{m}$ and $\max_{\theta,\theta'} \cos(\theta) - \cos(\theta') + \sin(\theta) - \sin(\theta') = 2\sqrt{2}$, and (iii) results from the fact that $\sigma_{\xi'} = (\alpha_r m n_0)^{-1/2} \sigma_\xi$ by definition of $\boldsymbol{\xi}'$.

The binary masks being drawn with uniform probability, we get

$$p_{\boldsymbol{s}_\mathcal{X}}(\boldsymbol{s}) = \textstyle\sum_{B \in (\mathcal{B}_r)^{n_0}} |\mathcal{B}_r|^{-n_0} \, p_{\boldsymbol{s}_\mathcal{X}}(\boldsymbol{s}|B)$$
$$\leq \textstyle\sum_{B' \in (\mathcal{B}_r)^{n_0}} |\mathcal{B}_r|^{-n_0} \exp(\epsilon) \, p_{\boldsymbol{s}_{\mathcal{X}'}}(\boldsymbol{s}|B') = \exp(\epsilon) \cdot p_{\boldsymbol{s}_{\mathcal{X}'}}(\boldsymbol{s}).$$

Integrating this density yields the desired result. □

Note that the $n_0^{-1/2}$ factor in the noise standard deviation comes from the fact that the local mechanism itself is adaptive to n_0 in its definition. Our mechanism only uses the size of the dataset (as opposed to other approaches relying on local and smooth sensitivities [13]), which under our definition of neighbouring datasets can be published without breaking differential privacy.

Masking impacts the parameter α_r, which helps to reduce the amount of noise required to get differential privacy, but will also impact the utility as we show in Section 6.

When the dataset is distributed across multiple devices, the composition properties of differential privacy can be used to get a result on the global mechanism, as shown in the following proposition.

Proposition 2. *The global sketching mechanism given in Definition 4, combining the sketches obtained from the different devices with the local mechanism (Definition 3) with r measurements per input sample and noise standard deviation $\sigma_\xi = \frac{4\sqrt{2\alpha_r m}}{\sqrt{n_0}\epsilon}$ on each device, achieves ϵ-DP (Definition 1).*

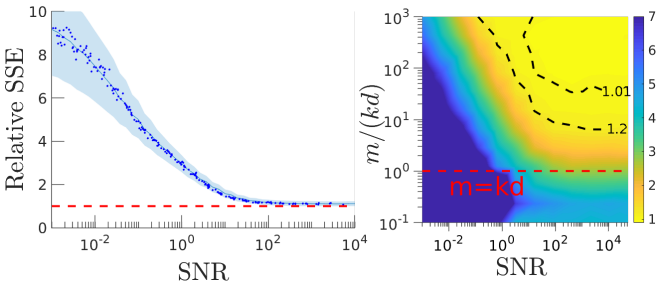

Fig. 2. (Left) Correlation between relative (w.r.t. k-means with 3 replicates) SSE and SNR for different values of α_r and σ_ξ, using $m = 10kd$. Medians of 40 trials, blue area shows the standard deviation. (Right) SSE as a function of SNR and m/kd, using $n = 10^5$, interpolated from a 12x12 grid, means of 40 trials, $k = d = 10$.

Proof. The devices operate on disjoint subsets of the global (distributed) dataset, so this is a direct consequence of the parallel composition property of differential privacy [14, Section 2.4.2]. □

5. EXPERIMENTAL MEASURE OF UTILITY

Having defined our mechanism, and shown that it achieves differential privacy, we now show how the noise and masking impact utility, defined for k-means as the SSE. For this, we define the signal-to-noise ratio (SNR) of the mean sketch and show experimentally that the SSE is directly driven by this SNR and the sketch size m. These results are formulated for an arbitrary noise variance σ_ξ^2 and masking parameter r. In section 6, we will use the theoretical values found in Section 4 to study how $\epsilon-$DP affects utility.

Definition of the SNR. We first compute the variance of one value of the noisy global sketch of a distributed dataset $\mathcal{X}$, computed using Definition 4 (i.e., using L devices with n_0 samples each) and Definition 3 with r measurements per input sample. We make the assumption that the samples of $\mathcal{X}$ have been drawn according to a distribution π, and denote $\boldsymbol{z} = \mathbb{E}_{\boldsymbol{x} \sim \pi} \boldsymbol{z}_{\boldsymbol{x}}$. From (5) we get

$$\mathrm{Var}((\boldsymbol{s}_\mathcal{X})_j) = \tfrac{1}{L} \mathrm{Var}_{\mathcal{X}, (\boldsymbol{b}_i)_{i=1}^{n_0}, \boldsymbol{\xi}} \Big[\tfrac{1}{\alpha_r n_0} \textstyle\sum_{i=1}^{n_0} (\boldsymbol{z}_{\boldsymbol{x}_i} \odot \boldsymbol{b}_i)_j + \tfrac{\boldsymbol{\xi}_j}{\sqrt{\alpha_r m n_0}}\Big]$$
$$= \tfrac{1}{L} \Big(\tfrac{1}{(\alpha_r n_0)^2} n_0 \mathrm{Var}_{\boldsymbol{x} \sim \pi, b \sim \mathrm{Bern}(\alpha_r)}[(\boldsymbol{z}_{\boldsymbol{x}})_j b] + \tfrac{\sigma_\xi^2}{\alpha_r m n_0}\Big)$$
$$= \tfrac{1}{\alpha_r n_0 L} \Big(\tfrac{1}{\alpha_r} \big(\tfrac{\alpha_r}{m} - |\alpha_r \boldsymbol{z}_j|^2\big) + \tfrac{\sigma_\xi^2}{m}\Big) = \tfrac{\frac{1}{m} - \alpha_r |\boldsymbol{z}_j|^2 + \frac{\sigma_\xi^2}{m}}{\alpha_r n_0 L}.$$

Hence, the SNR is defined as follows:

$$\mathrm{SNR} \triangleq \frac{\|\boldsymbol{z}\|^2}{\sum_{j=1}^m \mathrm{Var}((\boldsymbol{s}_\mathcal{X})_j)} = \frac{\alpha_r n_0 L \|\boldsymbol{z}\|^2}{1 - \alpha_r \|\boldsymbol{z}\|^2 + \sigma_\xi^2}. \tag{6}$$

Of these parameters, n_0 and L are fixed by the context (data format and application), and α_r, σ_ξ and m are free parameters of the mechanism. Provided that σ_ξ is independent of α_r, the SNR is therefore maximized when σ_ξ is as small as possible, and when $\alpha_r = 1$.

Measuring utility with the SNR. We now provide experimental simulations to exhibit the relation between the SNR and the SSE. All experiments are run in Matlab and based on the SketchMLbox toolbox, and learning is performed using the CL-OMPR algorithm [8]. For all experiments in this paper, data is drawn in $\mathbb{R}^d$ according to a mixture of k Gaussians $\pi = \frac{1}{k} \sum_{i=1}^k \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{I}_d)$, where $(\boldsymbol{\mu}_i)_{1 \leq i \leq k} \sim \mathcal{N}(0, (1.5k^{1/d})^2 \boldsymbol{I}_d)$. Figure 2 (left) depicts the correlation between relative (w.r.t. Lloyd's k-means) SSE and SNR for

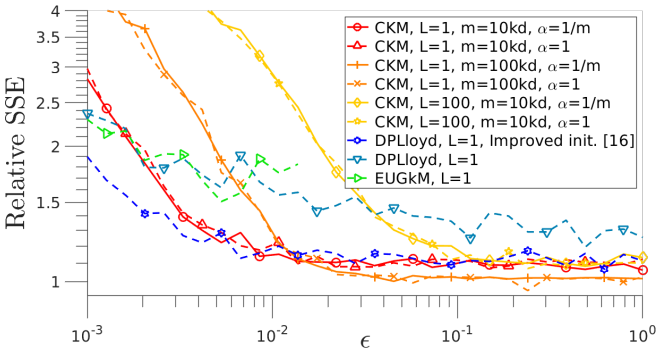

Fig. 3. Relative SSE (utility) vs. ϵ (privacy), for different values of L, α_r and m. Variances are similar for all methods. CKM stands for "Compressive k-means", and in that case the dashed lines correspond to $\alpha_r = 1$. Medians over 20 trials (less for EUGkM). Synthetic data, $d = k = 10$, $n = 10^7$. Best viewed in colors.

$k = d = 10$ and $m = 10kd$, for different values of r and σ_ξ. Note that in the definition (6) of the SNR, the quantity $\|\boldsymbol{z}\|^2$ is always smaller than 1 because the features (2) are scaled by $m^{-1/2}$, and depend on the data and frequencies distribution. Experiments performed on data drawn according to mixtures of $k > 1$ Gaussians showed that we always obtain $\|\boldsymbol{z}\|^2 \approx \delta \triangleq 0.35$ for the chosen distribution of the frequencies [7], independently of the mixture's parameters. In our experiments, we use this numerical approximation to generate values of α_r or σ_ξ for a given SNR. The correlation observed between the two quantities underscores the soundness of using the SNR as a measure of utility for a given sketch size.

Role of the sketch size. In Figure 2 (right), we plot the relative SSE (i.e. the ratio between the SSE obtained with our method and the SSE obtained with Lloyd's algorithm), for $n = 10^5$ using different sketch sizes and SNRs, computed using additive noise and without subsampling ($r = m$). The red dashed line corresponds to $m = kd$, and as expected [8] the reconstruction fails below this line. From this plot, we derive that when m is greater than $10kd$, one can consider that the reconstruction is successful provided that $\text{SNR} \times m/(kd) \geq 100$ (which corresponds to the yellow area, i.e. a relative SSE below 1.2).

6. PRIVACY-UTILITY TRADEOFF

In this section, we study the tradeoff existing between privacy guarantees and utility measures proposed respectively in Sections 4 and 5. When plugging the value of σ_ξ required to obtain ϵ-DP (Proposition 1) into the SNR (6), we obtain an expression of the latter parameterized by ϵ and other parameters of the mechanism:

$$\text{SNR}(\epsilon; n_0, L, \alpha_r, m) = \frac{\alpha_r \, n_0 \, L \, \delta}{1 - \alpha_r \delta + \frac{32 \alpha_r m^2}{n_0 \epsilon^2}}. \qquad (7)$$

We show in Figure 3 the privacy-utility curves, i.e. the relative SSE as a function of ϵ, obtained for different values of n_0, L, α_r and m, and we use this expression of the SNR (7) to explain the results.

We also compare our method with DPLloyd [15], a noisy variant of Lloyd's iterative algorithm, and EUGkM [16], which builds a noisy histogram of the data. DPLloyd was implemented in matlab according to [17], using 5 iterations as suggested by the authors. Initialization is performed either uniformly, or using the method proposed in [16]. We used the python EUGkM implementation[1] of the authors, with the suggested initialization.

When using $L = 1$ devices (centralized dataset), we obtain good clustering results, i.e. a relative SSE below 1.2, provided that $\epsilon > 10^{-2}$. The parameter α_r does not change much the results, as the variance term induced by the subsampling operation (i.e. using $\alpha_r < 1$) is small compared to the one induced by the additive noise in (7). Therefore we are free to use $\alpha_r < 1$ to reduce the computational cost, without degrading utility. Standard DPLloyd seems to perform poorly compared to our method, but using the improved initialization [16] does help significantly. For EUGkM, only points corresponding to lower values of ϵ could be calculated due to the histogram-based nature of the algorithm, the number of bins becoming substantial; such methods are mostly suited for low-dimensional datasets of moderate size. We thus conclude that our method performs at least as well as previously developed ones, while requiring less computations, having a controlled memory usage, and being usable in a distributed setting.

7. RELATED WORK

Differentially private k-means has already received attention in the literature. Addition of Laplacian noise has for instance been used in the SuLQ framework [15] that proposes a noisy version of Lloyd's iterative algorithm (DPLloyd), or in non-interactive approaches, such as EUGkM, that release noisy histograms of the data [18, 16]. The noise level is sometimes adapted to the instance on which the algorithm is performed, such as in the sample and aggregate approach [13]. The exponential mechanism, another standard approach choosing the output probability w.r.t. a user-defined utility measure, has also been used with genetic algorithms [19]. Private coresets have been investigated by Feldman et al. [20, 21]. The popular k-means++ seeding method has also been generalized to a private framework [22]. Balcan et al. investigated the large-scale high-dimensional setting with an approach based on Johnson-Lindenstrauss dimensionality reduction [23]. Many other embeddings based on random projections have been proposed as privacy-preserving encoding of data, see e.g. [24], often assuming that the projection matrix is unknown to the adversary.

Closer to our work, Balog et al. recently proposed to release kernel mean embeddings [25], either as sets of synthetic data points in the input space or using feature maps (as we do). Their work rely on the Gaussian mechanism, resulting in a weaker definition of differential privacy with an additive term $\delta > 0$. Studying the utility of a mechanism based on mean embeddings is, to the best of our knowledge, something new in the literature.

8. CONCLUSION

Compressing a dataset not only reduces the resources required for learning, but also intrinsically helps to build private algorithms. We propose a sketching method based on the addition of Laplacian noise, coupled to a subsampling mechanism for efficiency, which is proven to be differentially private.

Although we focused on k-means clustering, other learning tasks can be solved in a compressive manner and should be investigated, such as Gaussian mixtures fitting or principal components analysis. We leave for future work the idea of using quantized sketches [26] (quantization for privacy has already been considered [27, 28]), and leveraging fast transforms to speed-up the process [29]. Using additive noise on the data samples themselves is also a possibility that should be investigated.

[1] https://github.com/DongSuIBM/PrivKmeans

9. REFERENCES

[1] Rémi Gribonval, Gilles Blanchard, Nicolas Keriven, and Yann Traonmilin, "Compressive statistical learning with random feature moments," *arXiv preprint arXiv:1706.07180*, 2017.

[2] Cynthia Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[3] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine learning*, vol. 75, no. 2, pp. 245–248, 2009.

[4] Anil K Jain, "Data clustering: 50 years beyond K-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[5] Stuart Lloyd, "Least squares quantization in PCM," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[6] Ali Rahimi and Benjamin Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2008, pp. 1177–1184.

[7] Nicolas Keriven, Anthony Bourrier, Rémi Gribonval, and Patrick Pérez, "Sketching for large-scale learning of mixture models," *Information and Inference: A Journal of the IMA*, vol. 7, no. 3, pp. 447–508, 2017.

[8] Nicolas Keriven, Nicolas Tremblay, Yann Traonmilin, and Rémi Gribonval, "Compressive K-means," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 6369–6373.

[9] Shiva P Kasiviswanathan and Adam Smith, "On the'semantics' of differential privacy: A bayesian formulation," *Journal of Privacy and Confidentiality*, vol. 6, no. 1, 2014.

[10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 715–724.

[11] Noah Johnson, Joseph P Near, and Dawn Song, "Practical differential privacy for sql queries using elastic sensitivity," *arXiv preprint arXiv:1706.09479*, 2017.

[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, p. 20.

[13] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. pp. 75–84, ACM.

[14] Frank D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. pp. 19–30, ACM.

[15] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim, "Practical privacy: the SuLQ framework," in *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2005, pp. 128–138.

[16] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin, "Differentially private k-means clustering," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*. pp. 26–37, ACM.

[17] Ninghui Li, Min Lyu, Dong Su, and Weining Yang, "Differential privacy: From theory to practice," *Synthesis Lectures on Information Security, Privacy, & Trust*, vol. 8, no. 4, pp. 1–138, 2016.

[18] Wahbeh Qardaji, Weining Yang, and Ninghui Li, "Differentially private grids for geospatial data," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. pp. 757–768, IEEE.

[19] Jun Zhang, Xiaokui Xiao, Yin Yang, Zhenjie Zhang, and Marianne Winslett, "PrivGene: differentially private model fitting using genetic algorithms," in *Proceedings of the 2013 international conference on Management of data - SIGMOD '13*. p. 665, ACM Press.

[20] Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim, "Private coresets," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*. STOC '09, pp. 361–370, ACM.

[21] Dan Feldman, Chongyuan Xiang, Ruihao Zhu, and Daniela Rus, "Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks," in *Information Processing in Sensor Networks (IPSN), 2017 16th ACM/IEEE International Conference on*. pp. 3–16, IEEE.

[22] Richard Nock, Raphaël Canyasse, Roksana Boreli, and Frank Nielsen, "k-variates++: more pluses in the k-means++," in *International Conference on Machine Learning*, pp. 145–154.

[23] Maria-Florina Balcan, Travis Dick, Yingyu Liang, Wenlong Mou, and Hongyang Zhang, "Differentially private clustering in high-dimensional euclidean spaces," in *International Conference on Machine Learning*, pp. 322–331.

[24] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra, "Privacy via the Johnson-Lindenstrauss Transform," *Journal of Privacy and Confidentiality*, vol. 5, no. 1, 2013.

[25] Matej Balog, Ilya Tolstikhin, and Bernhard Schölkopf, "Differentially private database release via kernel mean embeddings," *arXiv:1710.01641 [stat]*.

[26] Vincent Schellekens and Laurent Jacques, "Quantized compressive k-means," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1211–1215.

[27] Petros Boufounos and Shantanu Rane, "Secure binary embeddings for privacy preserving nearest neighbors," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*. IEEE, 2011, pp. 1–6.

[28] Shantanu Rane and Petros T Boufounos, "Privacy-preserving nearest neighbor methods: Comparing signals without revealing them," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 18–28, 2013.

[29] Antoine Chatalic, Rémi Gribonval, and Nicolas Keriven, "Large-scale high-dimensional clustering with fast sketching," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.