DIFFERENTIALLY PRIVATE COMPRESSION K-MEANS

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

□ ICTTEAM/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 \( \mathcal{X} \triangleq \{x_i \in \mathbb{R}^d\}_{i=1}^n \) 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 \( \mathcal{C} \triangleq \{c_j \in \mathbb{R}^d\}_{j=1}^k \) minimizing the sum of squared errors (SSE):

\[
C^* = \arg \min_{\mathcal{C}} \text{SSE}(\mathcal{X}, \mathcal{C}) \triangleq \arg \min_{\mathcal{C}} \sum_{i \in \mathcal{X}} \min_{c_j \in \mathcal{C}} \|x_i - c_j\|^2. \tag{1}
\]

As solving (1) exactly is NP-hard [3], a variety of heuristics have been proposed [4], the most widely used being Lloyd’s 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 \( z_X \in \mathbb{C}^m \) (called the sketch) before performing the learning task. The sketch is defined as the sample average of random Fourier features [6], i.e.:

\[
z_X = \frac{1}{n} \sum_{x_i \in \mathcal{X}} z_{x_i}, \quad \text{with} \quad z_x \triangleq \frac{1}{\sqrt{m}} \exp(i\Omega^T x) \in \mathbb{C}^m, \tag{2}
\]

where \( m \) is the sketch dimension and \( \Omega = [\omega_1, \ldots, \omega_m] \) is a matrix composed of \( m \) frequency vectors \( \omega_j \in \mathbb{R}^d \), 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 \( z_X \) 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 \( x_i \) on \( z_X \) 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
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. $\mathcal{X} \sim \mathcal{X}' \iff (|\mathcal{X}| = |\mathcal{X}'| \text{ and } (|\mathcal{X} \cup \mathcal{X}'| \setminus (\mathcal{X} \cap \mathcal{X}')) \leq 2$).

**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 $\epsilon$ (noted $\epsilon$-DP) if for any measurable set $S$ of the co-domain of $f$:

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

(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 $n \equiv |\mathcal{X}| = L n_0$. 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 $\Omega$, the sketching parameters ($\alpha_\epsilon$ and $\sigma_\epsilon$, 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 $C^*$).

Compressing the local dataset $\mathcal{X}_l$ to the sketch $z_{\mathcal{X}_l}$ 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 $s_{\mathcal{X}_l}$, formally described hereafter. Our construction of $s_{\mathcal{X}_l}$ relies on the Laplace mechanism (a strategy to make an algorithm algorithmically 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 $\beta$ (denoted $z \sim C\mathcal{L}(\beta)$) if 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 $\sigma_z^2 = \beta^2/2$.

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

**Definition 3 (Local sketching mechanism).** The scrambled sketch $s_{\mathcal{X}_l}$ of a dataset $\mathcal{X}_l = \{x_j\}_{j=1}^{n_0}$ located on one device, using $r \in [1, m] \equiv \{1, \ldots, m\}$ measurements per record and a noise standard deviation $\sigma_\epsilon$, is defined as

$$s_{\mathcal{X}_l} \triangleq \frac{1}{\alpha_\epsilon n_0} \sum_{i=1}^{n_0} (z_{x_i} \odot b_{i}) + \frac{\xi}{\alpha_\epsilon mn},$$  

(4)

where $b_{xi} \equiv \mathcal{U}(\mathbb{B}_c)$ for all $i \in [1, n_0]$, $\xi_j \equiv \mathcal{U}(C(\sigma_\epsilon \sqrt{2}))$ for all $j \in [1, m]$, $\mathcal{B}_c \equiv \{b \in \{0, 1\}^m \mid \sum_{j=1}^m b_j = r\}$ is the set of binary masks $b$ selecting exactly $r$ coefficients, $\alpha_\epsilon \equiv r/m$ is called the subsampling parameter, and $\odot$ is the pointwise multiplication.

Normalizing by $\alpha_\epsilon n_0$ is needed to ensure $\mathbb{E}_{\mathcal{B}_c}[s_{\mathcal{X}_l}] = z_{\mathcal{X}_l}$. The noise is therefore normalized by $\sqrt{\text{var}(\xi)}$ 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:

$$s_{\mathcal{X}} \triangleq \frac{1}{L} \sum_{l=1}^{L} s_{\mathcal{X}_l}.$$  

(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 $\Omega$ 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 \( \sigma_x = \frac{\sqrt{\text{SNR}}}{\sqrt{m}} \), where \( \alpha_x \approx r/m \), achieves \( \epsilon \)-DP (Definition 1).

**Proof.** Let \( X \) and \( X' \) be two datasets such that \( X \sim X' \), that we rewrite as \( X = \{ x_i \}_{i=1}^{n_0} \cup \{ x \} \) and \( X' = \{ x_i \}_{i=1}^{n_0} \cup \{ x' \} \).

The scrambled sketch \( s_X \) of \( X \) can be written

\[
s_X = \frac{1}{\alpha_x} \sum_{i=1}^{n_0} (z_{x_i} \otimes b_{x_i}) + \frac{1}{\alpha_x} (z_x \otimes b_x) + \xi,
\]

where \( \xi \) is the rescaled noise. When conditioning on \( B \), the equivalent set of masks for \( X' \), i.e., such that \( b_{x_i}' = b_{x_i} \) for all samples \( x_i \), we denote the sketch \( s_X \) admits the density:

\[
p_{s_X}(s|B) = \exp\left( \frac{2}{\alpha_x} \right) \left( \left\| \bar{\mathbf{s}}_B - \frac{s_z \otimes b_x}{\alpha_x} \right\|_1 - \left\| \bar{\mathbf{s}}_B - \frac{s_{x_z} \otimes b_x}{\alpha_x} \right\|_1 \right) \exp(\epsilon),
\]

where \( \bar{\mathbf{s}}_B \) is defined similarly to \( s_B \). The binary masks being drawn with uniform probability, we get

\[
p_{s_X}(s) = \sum_{B \in \{0,1\}^n} \frac{1}{m} \exp(\epsilon) \cdot p_{s_X}(s|B).
\]

Integrating this density yields the desired result. \( \square \)

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 \( \alpha_x \), 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 \( \sigma_x = \frac{\sqrt{\text{SNR}}}{\sqrt{m}} \) on each device, achieves \( \epsilon \)-DP (Definition 1).

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 \( \sigma_x^2 \) and masking parameter \( r \).

**Definition of the SNR.** We first compute the variance of one value of the noisy global sketch of a distributed dataset \( X \), 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 \( X \) have been drawn according to a distribution \( \pi \) and denote \( \mathbf{z} = E_{x \sim \pi} \mathbf{x} \). From (5) we get

\[
\text{Var}(\mathbf{s}|x) = \frac{1}{N} \sum_{j=1}^{N} \text{Var}(\mathbf{x}_j) \approx \frac{1}{N} \sum_{j=1}^{N} \text{Var}(\mathbf{x}_j) = \frac{1}{N} \sum_{j=1}^{N} \left( \frac{1}{\alpha_x} \sum_{i=1}^{n_0} (z_{x_i} \otimes b_i) + \xi \right) = \frac{1}{N} \sum_{j=1}^{N} \left( \frac{1}{\alpha_x} \sum_{i=1}^{n_0} (z_{x_i} \otimes b_i) + \xi \right).
\]

Hence, the SNR is defined as follows:

\[
\text{SNR} \triangleq \sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)} = \frac{\sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)}}{\sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)}} = \frac{\sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)}}{\sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)}} = \frac{\sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)}}{\sqrt{\sum_{j=1}^{N} \text{Var}(\mathbf{s}|x)}}.
\]

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

**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 SketchML-box toolbox, and learning is performed using the CL-OMP algorithm (Algorithm 8). For all experiments in this paper, data is drawn in \( \mathbb{R}^d \) according to a mixture of \( k \) Gaussians \( \pi = \frac{1}{k} \sum_{i=1}^{k} N(\mu_i, I_3) \) (\( \mu_i \)s), where \( |\mu_i| \leq \sqrt{k} \sim N(0, (1.5k)^{1/2}I_3) \). Figure 2 (left) depicts the correlation between relative (w.r.t. Lloyd’s k-means) SSE and SNR for
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 $\sigma_{\epsilon}$ required to obtain $\epsilon$-DP (Proposition 1) into the SNR (6), we obtain an expression of the latter parameterized by $\epsilon$ and other parameters of the mechanism:

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

We show in Figure 3 the privacy-utility curves, i.e. the relative SSE as a function of $\epsilon$, obtained for different values of $n_0$, $L$, $\alpha$, 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 $\alpha_r$ 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 $\epsilon$ 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


