Novel differentially private mechanisms for graphs

Solenn Brunet, Sébastien Canard
Orange Labs, Applied Crypto Group
Caen, France
{solenn.brunet, sebastien.canard}@orange.com

Sébastien Gambs
Université de Quebec à Montréal
Montréal, Canada
gambs.sebastien@uqam.ca

Baptiste Olivier
Orange Labs, Applied Crypto Group
Cesson-Sévigné, France
baptiste.olivier@orange.com

***Abstract*—In this paper, we introduce new methods for releasing differentially private graphs. Our techniques are based on a new way to distribute noise among edges weights. More precisely, we rely on the addition of noise whose amplitude is edge-calibrated and optimize the distribution of the privacy budget among subsets of edges. The generic privacy framework that we propose can capture all privacy notions introduced so far in the literature to release graphs in a differentially private manner. Furthermore, experimental results on real datasets show that our methods outperform the standard existing techniques, in particular in terms of the preservation of utility. In addition, these experiments show that our mechanisms guarantee ϵ-differential privacy for a reasonable level of privacy ϵ, while preserving the spectral information of the input graph.**

I. Introduction

Nowadays, Online Social Networks (OSNs) are used by billions of users to connect and share information. On the one hand, OSNs can provide useful insights on societal phenomena such as epidemiology, information dissemination, marketing and sentiment analysis [20], [27], [34], [35]. On the other hand, OSNs usually refuse to publish the structure of their social network graphs due to privacy concerns. Indeed, social graphs can leak sensitive information about individuals such as their jobs, diseases or acquaintances, just to cite a few. In particular if the graph is not properly sanitized, reidentification attacks are possible [12], [26] as well as other type of inference attacks [10], [23], [37].

Thus, a special attention was paid in the literature to design sanitization mechanisms for graphs and their adjacency matrices. In this work, we focus particularly on differential privacy [6]. Originally introduced in the context of databases, differential privacy settles a rigorous framework to privately release data while permitting a control on the trade-off between utility and privacy. Because of its rigorous privacy guarantees, this notion is now at the heart of the research on data privacy, and the literature on this subject is quite extensive. For instance, techniques for releasing differentially private graphs were studied in several previous works [1], [13], [14], [24], [32].

Any differentially private mechanism gives privacy guarantees with respect to some predefined notion of privacy, which is concretely parametrized by a neighbouring relation between inputs. In many papers studying differential privacy on graphs (e.g., [14]), the privacy notion considered is *edge privacy*, which aims at hiding the possible addition or the subtraction of an edge in a graph. A generalization called *edge weight* was introduced in [29]. A stronger notion, called *node privacy*, has for objective to hide the presence or absence of a node in the graph [13]. In addition, other works have adopted another point of view by working on the adjacency matrix instead directly of the graph itself [33]. Several ways to release differentially private matrices were also studied: for private spectral graph analysis [11], [17], for Singular Value Decomposition (SVD) [4], [8] or for the Johnson-Lindenstrauss transform [2], [18]. The privacy notions adopted in these papers include *row privacy* or *coefficient privacy*.

In our work, we introduce a very generic framework that allows for applications fitting with any of the privacy notions mentioned previously. As a consequence, existing privacy notions for graphs can be used together with the novel edge-calibrated algorithms that we propose in the current paper. Furthermore, as illustrated by Example A in the next section, there are real-life situations for which the generality of our framework is necessary.

Our main objective is to propose new techniques for releasing differentially private (directed) weighted graphs. Any weighted graph admits an equivalent representation given by the adjacency matrix of the graph. For simplicity and without loss of generality, we design our model to work on this matrix representation. More formally, our model considers a space of databases $\mathcal{D}$, and a matrix-valued query $\psi : \mathcal{D} \to \mathbb{R}^{n\times n}$ for a parameter n. In particular, we associate a matrix $A = \psi(x) \in \mathbb{R}^{n\times n}$ to any database $x \subset \mathcal{D}$. Our aim is to release a private version $\tilde{A} \in \mathbb{R}^{n\times n}$ of A in a differentially private manner, considering the possible inclusion or exclusion of a single individual in the databases from $\mathcal{D}$. Going back to the graph representation, possible vertices of $\psi(x)$ are represented as integers in $[n]$, and the coefficient $A_{ij} = \psi_{ij}(x)$ corresponds to a *weight* on edge (i, j). For example, most social networks can be represented as a weighted graph in which vertices correspond to the individuals and an edge connecting two vertices i and j is weighted by the number of interactions between individuals i and j.

A differentially private mechanism is generally obtained by constructing a randomized algorithm whose noise is calibrated to some quantity measuring the impact on the output when adding or subtracting an individual in the database. The most common example of such a quantity is called the (global) *sensitivity* and was introduced in the seminal paper on differential privacy [6]. A refinement of this notion called *local sensitivity* was introduced in [28], and used in many subsequent papers to design new differentially private mechanisms [14], [36]. Our approach is not designed to subsume [28] (or other notions of local sensitivities invented so far), but rather is complementary to it. More precisely, techniques from [28] aim at answering successive queries $f(x_1), f(x_2), \ldots$ adapting the noise to each instance x_i considered. In contrast, our technique aims at improving the trade-off of a single instance of a matrix query $\psi(A)$. Thus, we are convinced that our framework and that of [28] can be combined together to answer multiple matrix queries $\psi(A_1), \psi(A_2), \ldots$ while decreasing the privacy budget required. But we leave this as future work.

Summary of our contributions. Our main contribution is a new method for sanitizing matrix queries that exploits the variations among sensitivities relative to some subsets of matrix coefficients to release differentially private matrices. Instead of sampling noise from the same distribution for each coefficient of the matrix, we make use of *coefficient-calibrated* sensitivities to tune the noise of our differentially private mechanism. To achieve ϵ-differential privacy for a fixed privacy budget ϵ, we optimize the distribution of the amplitude of the noise among coefficients so that coefficients with lower sensitivities are less perturbed than other ones. This contrasts with most current methods that do not make any adaptation of noise to the coefficients considered. More precisely, our main contributions can be summarized as follows.

1) A general framework to study differential privacy on weighted graphs. We provide a generic definition of the neighbouring relation, which can be used to capture most of the contexts already appearing in the literature (see Section II-B), but also situations not considered so far (see Example A). Our framework is generic in the sense that a single individual can affect not only a single coordinate but rather several coefficients at the same time, with weights that can vary for each of them. Our formalism is very close to the one already introduced in [29], although slightly different (see Section II-B for details).

2) Block Laplacian mechanism. Our main contribution is the design of a new type of mechanism, that we coin as *block noisy mechanism*. Although it can be applied with many types of noise, we illustrate our technique with Laplacian noise. Our new *"Block Laplacian"* mechanism is a variant of Laplacian mechanism that takes advantage of the possible inhomogeneity of sensitivities on coefficients of the considered matrix-query ψ. More precisely, Block Laplacian mechanism adds noise on coefficients adaptively with respect to their sensitivities. In Theorem 5, we describe explicitly the optimal parameters for Block Laplacian mechanism to be ϵ-differentially private.

3) Practical use of block noisy mechanisms. In many real-life situations, the graph structure at stake is rather complex, and the required knowledge on edges sensitivities is not always available. For such cases, we design a differentially private mechanism that, at the cost of a possible loss of accuracy, allows to give tight approximations of sensitivities. Moreover, as first noticed in [3] (see also [4], [8]), *additive-noise* differentially private mechanisms (so far Laplacian or Gaussian mechanisms) can be combined with rank k-approximation induced by SVD to obtain more accurate results (at least for small values of k). Thus, our mechanism can be post-processed by SVD as well, and we show in our experiments that this post-processing improves the resulting utility.

4) Experimental validation. When combining Block Laplacian mechanism with SVD post-processing, we call the resulting algorithm *BlockLaplacianThenSVD*. We use our implementations of the Block Laplacian mechanism and BlockLaplacianThenSVD to compare them experimentally to Laplacian mechanism and LaplacianThenSVD , as well as to *non-private* k-rank approximation, as a measure of the quality of these private algorithms. We apply these algorithms on real datasets of Call Detailed Records (CDRs) of a major mobile phone operator, in a real-life scenario explained in details in Example A. Our experiments show that for small values of the rank parameter k, our algorithm BlockLaplacianThenSVD require only a limited level of noise, for a reasonable level of privacy (i.e., $\epsilon \sim 1$). Moreover, we illustrate how the quality of these algorithms degrades as parameter k grows larger. We believe that this study is of independent interest to understand more deeply the level of privacy offered by differential privacy in real-life scenarii.

Outline. This paper is organized as follows. First, Section II introduces our model, our motivating example for concrete applications and the basic notions related to differential privacy. Then, Section III develops our new framework for Laplacian noise mechanisms. Afterwards, Section IV provides a framework for using our new algorithms in practice, in particular when sensitivities of the released graphs are not well understood. Our experiments are explained and analyzed in Section V. Finally, we compare our techniques to the existing literature in Section VI, before concluding in Section VII. We refer to the Appendix VIII for privacy proofs, additional experimental results, and an analysis of *"Block Gaussian"* mechanism .

II. Differential privacy on graphs and matrices

This section introduces the basic notions related to differential privacy used in this paper as well as describing our main motivating example for this work.

A. Matrix model for private graphs

In this paper, we consider the situation in which a *sanitizer* owns databases $x \subset \mathcal{D}$ and wants to release some graphs $(\psi(x))_{x \subset \mathcal{D}}$ in a differentially private manner. We chose to give an equivalent representation of graphs $\psi(x)$ as matrices via their adjacency matrices, for simplicity in mathematical manipulations (e.g., SVD).

To define our model and our neighbouring relation more formally, let n be a parameter and let $\psi : \mathcal{D} \to \mathbb{R}^{n \times n}$ be a query function mapping any (sub)database $x \subset \mathcal{D}$ to a $n \times n$

real-valued matrix $A = \psi(x) \in \mathbb{R}^{n\times n}$. We fix ψ once and for all, which is why we omit to write it in the sequel.

We study randomized mechanisms $\mathcal{A} : A \mapsto \tilde{A}$ releasing a differentially private version $\tilde{A}$ of matrix A with respect to the following notion of neighbourhood on matrices.

Definition 1 (Neighbouring relation): We say that two sub-databases $x, x' \subset \mathcal{D}$ are neighbours, which we denote by $x \sim x'$, if they differ from the records of a single individual from the database $\mathcal{D}$. In this situation, these two matrices $A, A' \in \mathbb{R}^{n\times n}$ are said to be neighbours, and we denote by $A \sim A'$ the fact that they come from two neighbouring sub-databases $x, x' \subset \mathcal{D}$ (i.e., if $A = \psi(x)$ and $A' = \psi(x')$ for $x \sim x'$).

We now describe one of the main example that motivated this work, namely the sanitization of mobility traces. Mobility traces are known to be privacy-sensitive due to re-identification and inference attacks possible on this type of data [30], [9].

Example A (Mobility analysis from mobile phone usage): In this example, $\mathcal{D}$ contains mobility data generated by phone usage, also named Call Details Records (CDRs). More precisely, assume that the *sanitizer* owns some datasets $x \subset \mathcal{D}$, each containing the following information related to calls of users: timestamp and location of calls (given by the location of the corresponding antenna) during some fixed period.

More formally, we consider a phone network composed of n antennas. The phone operator owns the information $x_{I_1}, \dots, x_{I_N}$ of N individuals $I_1, \dots, I_N$. For a given pair of antennas (i, j) (called *transition* (i, j) in the sequel), we count the number of times a call at antenna i was followed by a call at antenna j, during the observation period: we denote by $\psi_{ij}(x_{I_k}) = A_{ij}^{I_k} \in \mathbb{N}$ this particular variable for user I_k. For each transition (i, j), $i, j \in [n]$, we then aggregate the scores over all individuals as follows:

$$A_{ij} = \sum_{k=1}^{N} A_{ij}^{I_k}.$$

This aggregated value can be represented by a matrix $A = (A_{ij})_{ij}$, and the objective is to release A privately with respect to the impact of the addition or subtraction of an individual in the database.

Remark that the above modelling is very generic and could applied to many other situations, such as e.g., social networks or history of log files. A crucial notion when designing differentially private mechanisms is the sensitivity of the query or the object to release. In fact, any differentially private mechanism calibrates the amplitude of the noise applied to this sensitivity. Hereafter, we provide the definition of sensitivity that we have adopted in our methods, which is the analog of the sensitivity of a query, thinking of A as the answer of a query. We point out that we could define the sensitivity in other ways, depending on the mechanism we want to design and the type of noise used to perturb the output.

Definition 2 (ℓ_1-sensitivity for matrices): The *ℓ_1-sensitivity* for matrices Δ^{ℓ_1} is given by the formula

$$\Delta^{\ell_1} = \max_{A' \sim A} \sum_{i,j} |A_{ij} - A_{ij}'|$$

in which the $\max$ is taken over all possible pairs of neighbours $A \sim A'$.

B. Relationship with existing privacy notions for graphs and matrices

Many privacy notions for graphs can be interpreted in terms of Definition 1, by setting the appropriate notion of neighbouring relation. In particular, our methods can directly be applied in contexts in which the following notions of privacy occur.

Edge privacy for graphs [14]. In this notion, two graphs are neighbours if they differ by a single edge. This is a particular case of Definition 1 in which an *individual* is represented as a single edge.

Node privacy for graphs [13]. With this notion, two graphs are neighbours if they differ by a single vertex. Node privacy can also be modelled by Definition 1, in which we identify an individual as a single vertexacting only on the weights corresponding to edges connected to this vertex.

Row privacy for matrices [8]. Two matrices are neighbours if they differ by a single row. This notion is a particular instance of Definition 1 in which each row of the matrix can be perturbed by one and only one individual.

Remark that Example A cannot be modelled by any of the above privacy notion, since it allows individuals to act through weights on arbitrary edges of the graph.

Edge weight privacy [29] The closest neighbouring notion to ours is given by Definition 2.1 in [29], which introduced differential privacy with respect to edge weight for the first time. However, our notion is slightly different since we do not assume a uniform bound on $A_{ij} - A'_{ij}$, even after normalization. Rather, we provide a practical mechanism that can handle situations in which there is no a priori knowledge on such a bound on sensitivities (see Section IV). As in [29], our individuals can be represented as weight functions, but in practice (see Example A), we only use weight functions restricted to a small subset of edges.

C. Achieving differential privacy on matrices

We start with the definition of differential privacy stated in the context of matrices. Let $\mathbb{P}(E)$ denote the probability that the event E occurs.

Definition 3 (ϵ-differential privacy for matrices): Let

$$\begin{aligned} \mathcal{A} :\ & \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n} \\ & A \mapsto \tilde{A} \end{aligned}$$

be a randomized mechanism, and $\epsilon > 0$. We say that a mechanism $\mathcal{A}$ is ϵ-differentially private if

$$\frac{\mathbb{P}(\tilde{A} \in S)}{\mathbb{P}(\tilde{A'} \in S)} \le e^{\epsilon} \text{ for all } S \subset \mathbb{R}^{n\times n} \text{ and all } A \sim A'.$$

The most basic ϵ-differentially private mechanism [3] that releases a private version $\tilde{A}$ of a matrix A is called the Laplacian mechanism and is obtained by the following formula:

$$\tilde{A} = A + B\ ,$$

in which $B \in \mathbb{R}^{n\times n}$ is a random matrix such that coefficients $(B_{ij})_{ij}$ are independent Laplace random variables with parameter $\lambda = \frac{\epsilon}{\Delta^{\ell_1}}$.

In many real-life scenarii (such as the one described in Example A), it occurs that some coefficients of the matrix A are not sensitive by nature. In practice, this means that no individual from the database has an impact on such coefficients. To preserve the coherence of the output as well as the accuracy of the model, these non-sensitive coefficients should not be perturbed by the mechanism. In the sequel, we will use the notation

$$\mathcal{S} = \{\ (i,j) \in [n]^2\ |A_{ij} \neq A_{ij}' \text{ for some } A \sim A'\ \}\ ,$$

to represent the set of all sensitive coefficients of matrices resulting from our database $\mathcal{D}$. The complement of this set, which is the set of non-sensitive coefficients, is invariant under the neighbouring relation $\sim$. In particular, the Laplacian mechanism described above remains ϵ-differentially private if only coefficients $(i,j) \in \mathcal{S}$ are perturbed by Laplacian random variables.

Example A (Non-sensitive coefficients for CDRs): Going back to Example A about mobility data issued CDRs, it appears that many transitions (i,j) are non-sensitive (i.e., no transition occurs between antenna i and antenna j, which means $A_{ij} = 0$). Indeed, CDRs reflect the mobility patterns of the users, which results in a sparse transition graph and thus also a sparse matrix. Hence for Example A, we have the following description for the set of sensitive coefficients :

$$\mathcal{S} = \{\ (i,j)\ |\ A^I_{ij} \neq 0 \text{ for at least one user } I\ \}.$$

III. Block sensitivities and Block Laplacian mechanism on matrices

This section describes the main contribution of our work, which is a differentially private mechanism adapted to block sensitivities. For simplicity in the following privacy proofs, we investigate the case of Laplacian random variables to introduce our methods. The interested reader is referred to the Appendix for the analog results for Gaussian mechanisms.

The first part of this section provides explanations and gives the intuition behind our technique and the design of Block Laplacian mechanism depending on a given partition of the coefficients of matrices, and their corresponding sensitivities. Afterwards, the second part shows how our framework can be used in the situation in which only coefficient sensitivities are known, which is a more realistic case.

A. Sensitivity on groups of coefficients and Block Laplacian mechanism

When restricted to sensitive coefficients, the standard Laplacian mechanism on a matrix A uses the same amplitude perturbation $\lambda = \frac{\epsilon}{\Delta^{\ell_1}}$ for all coefficients A_{ij}, $(i,j) \in \mathcal{S}$ (see Section II-C). However, due to particular characteristics of the dataset $\mathcal{D}$ and the matrix query ψ, it may happen that the sensitivity is mostly located on some specific coefficients. In contrast, some other coefficients could be almost private while being sensitive, in the sense that the inclusion or exclusion of a single individual in databases from $\mathcal{D}$ does not significantly impact them. In this situation, almost private coefficients should not be perturbed as much as the most sensitive ones. Hereafter, we show that it is possible to design a mechanism that we call Block Laplacian mechanism, which perturbs the almost private coefficients with a lower level of noise. In a nutshell, the Block Laplacian mechanism allows for much better utility than standard Laplacian noise, while providing exactly the same privacy guarantees.

For simplicity in the rest of this section, we write the ℓ_1-sensitivity as Δ instead of Δ^{ℓ_1}. We also fix a partition $(S_k)_{k=1}^K$ of the set $\mathcal{S}$ of sensitive coefficients, and we denote by n_k the cardinality of the set S_k. We are interested in the changes occurring in each block of indices S_k of matrices $A = \psi(x)$, $x \subset \mathcal{D}$, when we add or subtract an individual in the database.

Definition 4 (Block sensitivities for matrices): The block sensitivities for matrices $(\Delta_{S_k})_k$ relative to the partition $(S_k)_{k=1}^K$ are defined as follows:

$$\Delta_{S_k} = \max_{A'\sim A} \sum_{(i,j)\in S_k} |A_{ij} - A_{ij}'|\ .$$

For convenience of notations, we denote Δ_{S_k} by Δ_k when the context is clear. If the partition has a single element (i.e., $S_1 = \mathcal{S}$), then we recover sensitivity as defined in Section II-A. We are now ready to state our main result.

Theorem 5 (Block Laplacian mechanism): Let $\epsilon > 0$ and let $(S_k)_{1\leq k\leq K}$ be a partition of the set $\mathcal{S}$ of sensitive coefficients. We define

$$\lambda_k = \frac{\epsilon}{\Delta_k} \times \frac{1}{\sum_{j=1}^K \sqrt{\frac{n_j \Delta_j}{n_k \Delta_k}}}\ .$$

The Block Laplacian mechanism $\mathcal{A} : A \mapsto \tilde{A}$ is defined by the following formula:

$$\tilde{A_{ij}} = A_{ij} + B_{ij}$$

in which :
- $B_{ij} = 0$ if $(i,j) \notin \mathcal{S}$;
- B_{ij} is a Laplace random variable of mean 0 and standard deviation $\sigma_k = \frac{\sqrt{2}}{\lambda_k}$ if $(i,j) \in S_k$. In this case, mechanism $\mathcal{A}$ is ϵ-differentially private. Moreover, $(\lambda_k)_k$ defined as above is an optimal choice in the following sense: writing $\lambda_k = \frac{\epsilon_k}{\Delta_k}$ for all k, our choice realizes the minimum of the mean-error function

$$\varphi(\epsilon_1, \ldots, \epsilon_K) = \sum_{k=1}^K n_k \times \frac{\Delta_k}{\epsilon_k}$$

under the constraint that $\epsilon = \sum_{k=1}^K \epsilon_k$.

Note that the second part of Theorem 5 asserts that, once a partition $(S_k)_k$ is fixed, our choice of $(\lambda_k)_k$ (or equivalently $(\epsilon_k)_k$) minimizes the mean-error among the possible other divisions $(\epsilon'_k)_k$ of the privacy budget $\epsilon = \sum_k \epsilon'_k$. More precisely, our choice of $(\lambda_k)_k$ is made to minimize the ℓ_1 mean-error on coefficients $\mathbb{E}(\sum_{i,j} |A_{ij} - A'_{ij}|)$. We chose the latter distance since it is natural when using mechanisms like the Laplacian one. However, one could prefer to minimize another distance [8], such as the ℓ_2 mean-error on coefficients $\sqrt{\mathbb{E}(|A_{ij} - A'_{ij}|^2)}$, also called the Frobenius norm. In this case, the optimal budget division requires other values of $(\lambda_k)_k$, obtained by minimizing the ℓ_2 theoretical error.

The choice of the partition $(S_k)_k$ is not trivial, and depends completely on the structural properties of the pair data/query we are looking at, which is the pair $\mathcal{D}/\psi$ with our notations. Hence to provide an accurate model, the owner of the sensitive data needs to have some knowledge about the localization of coefficient sensitivities of its possible output matrices $A = \psi(x)$. If the graph structure at stake is too complex to provide an a priori useful information on sensitivities, we propose in Section IV-B a mechanism that handles the computation of sensitivities while providing differential privacy guarantees.

B. Designing the partition $(S_k)_k$ from the knowledge of coefficients sensitivities

The quality of the Block Laplacian mechanism depends on a clever choice of some partition $(S_k)_k$ of the coefficients. In this section, we explain how to design such a good partition when only coefficient sensitivities Δ_{ij} are known to the sanitizer (and not all sensitivities Δ_k for all possible choice of partition $(S_k)_k$). This reduction for designing a partition is particularly interesting when *no knowledge at all* is available on sensitivities, as explained in the Section IV-B.

The sensitivity on coefficient (i, j) is defined as

$$\Delta_{ij} = \max_{A \sim A'} |A_{ij} - {A_{ij}}'|.$$

For a given threshold $\tau > 0$, we can easily define a partition $P_\tau = (S_1, S_2)$ as follows:

$$S_1 = \{\ (i,j) \mid \Delta_{ij} > \tau\ \} \text{ and } S_2 = \{\ (i,j) \mid \Delta_{ij} \leq \tau\ \}.$$

The best value τ for our purpose is the one that minimizes the mean error of Block Laplacian mechanism, which is straightforwardly and efficiently computable from Theorem 5. More details on the computation of the best τ, and generalization to the case $K > 2$ are given in the Appendix.

This design of partition $(S_k)_k$ provides good performance with Block Laplacian mechanism when a single individual affects a small number of coefficients of the matrix, as shown by our experiments in Section V. Indeed, if the action of a single individual is restricted to a small subset of coefficients, then sensitivities Δ_{ij} relative to each coefficient (i, j) can be used to approximate well-adapted partitions. Let m denote a bound on the maximum number of coefficients affected by a single individual. Then, fix $\tau > 0$ and let $P_\tau = (S_1, S_2)$ be as above. It is easily seen that $\Delta_2 \leq m \times \tau$, and that for small values of m, τ, a noise calibrated to Δ_2 certainly perturbs coefficients of S_2 much less than a noise calibrated to global sensitivity Δ.

Example A (m, S_1, S_2 in the case of CDRs): In Example A and for our data (see Section V), the value of m is rather small. More precisely, it is around $20 * 20$ for a matrix of size $n = 1666$, which means that an individual contributes to at most 400 coefficients from $1666 * 1666$ coefficients in total. Moreover, most of the calls are made *on site* (e.g., at home or at work), which means that most of the cells impacted are the ones related to transitions of the form (i, i), which correspond to diagonal coefficients. Thus, *sensitive* transitions in S_1 are more likely to be diagonal transitions (i, i), and *non-sensitive* transitions non-diagonal transitions (i, j) for $i \neq j$.

IV. Improvements for practical use of Block Laplacian mechanism

The aim of this section is two-fold. First, we explain how to design a differentially private mechanism when no information about sensitivities is known, and then we apply this principle to design a version of Block Laplacian mechanism for such situations. Second, we recall that a combination of Block Laplacian mechanism and a k-rank approximation can provide better results.

A. A differentially private mechanism for unknown sensitivities

In this section, we consider only one-dimensional queries $f : \mathcal{D} \to \mathbb{R}$ that are linear with respect to individual data, i.e., $f(x) = \sum_{I \in x} f(x_I)$ in which x_I is the data of individual I and $I \in x$ means that the data of this individual x_I is part of the dataset x. Now, we design a differentially private mechanism, that can be used in situations when no accurate approximation on the sensitivity $\Delta = \Delta(f)$ is known to the sanitizer. The idea of query-truncation behind this mechanism, which already appeared in Algorithm 1 from [13], is as follows. First, we choose a reference-database x_0 upon which our protocol depends. Afterwards, we compute $\Delta_{x_0} = \max_{I \in x_0} |f(x_I)|$, and we choose an individual I_0 from x_0 realizing this maximum, which means that $\Delta_{x_0} = f(x_{I_0})$. Then, we define f_{x_0} as $f_{x_0}(x_I) = f(x_I)$ if $|f(x_I)| \leq \Delta_{x_0}$, and $f_{x_0}(x_I) = f(x_{I_0})$ if $|f(x_I)| > \Delta_{x_0}$ (this correctly defines $f(x)$ for all $x \in \mathcal{D}$ by linearity). We also define the mechanism $\mathcal{A}_{x_0}$ by

$$\mathcal{A}_{x_0}(f)(x) = f_{x_0}(x) + Z_{x_0} \text{ for all } x \in \mathcal{D},$$

in which Z_{x_0} is a random Laplacian of parameter $\frac{\sqrt{2}\Delta_{x_0}}{\epsilon}$.

Theorem 6: The mechanism $\mathcal{A}_{x_0}$ as defined above is ϵ-differentially private.

The most important remark is that the sanitizer is not allowed to change the reference-database x_0 in order to preserve the differential privacy guarantees. For instance, if x_0 and x_1 are two distinct databases, and $\mathcal{A}_{x_0}$, $\mathcal{A}_{x_1}$ are each ϵ-differentially private, then the composition of $\mathcal{A}_{x_0}$ and $\mathcal{A}_{x_1}$ is not 2ϵ-differentially private (in contrast to the composition theorem in [7]). The reference-database x_0 should reflect the global behaviour of databases $x \subset \mathcal{D}$. In this case, the error due to the truncation operation $f \to f_{x_0}$ is small. For instance, this can be achieved by taking x_0 as large as possible (this depends on the amount of data owned by the sanitizer): only a few outliers I out of x_0 may satisfy $|f(x_I)| > M_{x_0}$. We highlight the fact that protocol $\mathcal{A}_{x_0}$ does not depend on a particular instance $x \in \mathcal{D}$, if the sanitizer fixes x_0 once and for all. This approach should not be mistaken with instance-based mechanisms of the form $f(x)+Z_x$, in which the noise Z_x depends on the instance value x [28].

B. Block Laplacian mechanism when coefficient sensitivities are unknown

Hereafter we use the idea of the previous section, combined with our results from Section III, to design a version of Block Laplacian mechanism when no information is known about the sensitivities Δ_{ij}. We only sketch how techniques described previously in this paper could be combined (see the Appendix for more details).

To apply the result of the previous section, we need to assume that each coefficient query ψ_{ij} is linear (as described in Section IV-A), and we denote the result of this query on individual I by A^I_{ij}. The various techniques seen so far could be combined as follows.

1) Choose a reference-database $x_0 \subset \mathcal{D}$.
2) Compute the sensitivities of the reference $\Delta_{x_0,ij} = \max_{I\in x_0} |A^I_{ij}|$
3) Compute a partition $(S_{x_0,1}, S_{x_0,2})$ and the corresponding sensitivities $(\Delta_{x_0,1}, \Delta_{x_0,2})$, using Search for 2-blocks partition with $\Delta_{x_0,ij}$ instead of Δ_{ij}.
4) For each $1 \le k \le K$, find I^k_0 realizing the maximum $\Delta_{x_0,k} = \sum_{(i,j)\in S_{x_0,k}} |A^{I_0}_{ij}|$.
5) Define the truncation version A_{x_0} of matrix A by the following formulae, for $(i,j) \in S_{x_0,k}$:
$$A^I_{x_0,ij} = A^I_{ij} \text{ if } \sum_{(i,j)\in S_{x_0,k}} |A^I_{ij}| \le \Delta_{x_0,k}$$
$$= A^{I^k_0}_{ij} \text{ otherwise.}$$
6) Given a privacy parameter ϵ, define the randomized mechanism $\mathcal{A}_{x_0} : A \to \tilde{A}$ by $\tilde{A}_{ij} = A_{x_0,ij} + B_{ij}$ in which B_{ij} is a random matrix defined as in the statement of Theorem 5, in which S_k (resp. Δ_k) is replaced by $S_{x_0,k}$ (resp. $\Delta_{x_0,k}$).

Theorem 7: Assuming the linearity of each coefficient query ψ_{ij}, the mechanism $\mathcal{A}_{x_0}$ defined above is ϵ-differentially private.

The previous protocol relies on the computation of sensitivities $\Delta_{x_0,ij}$, which is far more efficient and reasonable than computing all sensitivities $\Delta_{x_0,S_1}, \Delta_{x_0,S_2}$ for all possible choice of partitions $\mathcal{S} = S_1 \sqcup S_2$. Note that Example A meets the linearity assumptions of this section.

C. Improving the performance using the Singular Value Decomposition

A well-known fact is that the amount of noise in *additive-noise* mechanisms can be reduced by performing a k-rank approximation on the perturbed matrix for a rather small parameter k. Since this operation is performed after the addition of noise, differential privacy guarantees are still preserved. Such a post-processing was already used in previous works on differential privacy [3], and especially in [22] to produce recommendation systems with differential privacy guarantees. The benefits of using this approach will be demonstrate in our experiments conducted in the next section.

V. Experiments on real-life datasets

Our experiments have two main objectives. The first one is to provide experimental evidences that the use of block sensitivities in additive noise mechanisms, outperforms uniform amplitude noise mechanisms. Secondly, BlockLaplacian and BlockLaplacianThenSVD depend on several parameters ϵ, k, K, $(\tau_i)_i$, and our experiments show the dependence of the accuracy of our results on these parameters.

A. Experimental setting

In this section, we provide the details of our experimental setting such as the description of the algorithms and datasets used as well as the evaluation metrics upon which we rely.

Mechanisms and notations.

1) L stands for Laplacian mechanism (see Section II-C), and BL for Block Laplacian mechanism (see Section III-A).
2) $LSVD$ (respectively $BLSVD$) stands for LaplacianThenSVD (respectively BlockLaplacianThenSVD) discussed in Section IV-C.
3) SVD is the standard k-rank approximation.

The differentially private mechanisms L, $LSVD$ are compared with respect to the same level of privacy ϵ, and over a unique number N of individuals in the data. The parameters ϵ and N are detailed in the next section.

Evaluation of our results. The standard distance used in the literature to measure the closeness of two matrices is the Frobenius norm, which is induced by the ℓ_2-norm on coefficients. In this paper, we chose to evaluate our results with a close variant of the Frobenius distance, which is given by the ℓ_1-norm on coefficients and measures the distance between two matrices $A, B \in \mathbb{R}^{n\times n}$ by the following formula:
$$|A-B|_1 = \sum_{1\le i,j\le n} |A_{ij} - B_{ij}|.$$

This measure is more natural for algorithms using Laplacian random noise since it relates more closely to the subsequent definition of sensitivity, whereas the ℓ_2-norm would be more adapted to Gaussian random noise.

Moreover, as we consider large matrices and large datasets, the quality of the results may not be easy to interpret. To ease this interpretation, we normalize the above distance as follows. We choose randomly $2l$ datasets $x_1, \ldots, x_{2l} \subset \mathcal{D}$ all of the same size N, and we consider the associated matrices $A_1 = \psi(x_1), \ldots, A_{2l} = \psi(x_{2l})$. Afterwards, we define the normalization factor by
$$d_l = \frac{1}{l} \times \sum_{k=1}^{l} |A_{2i-1} - A_{2i}|_1 \ ,$$

and our distance d for evaluation by
$$d(A,B) = \frac{1}{d_l} \times |A-B|_1.$$

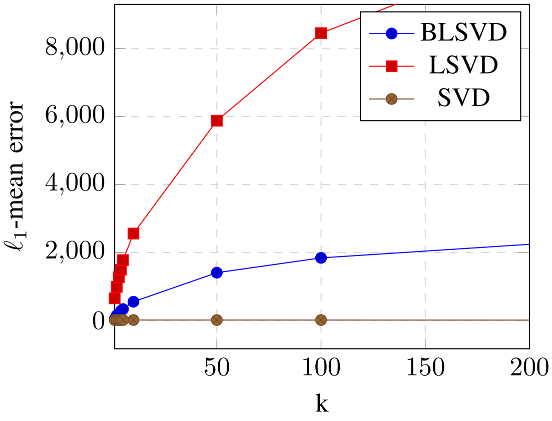
Assuming convergence of $(d_l)_l$ to some average value, the intuition behind our choice of metric d is that a value $d(A, \tilde{A})$ close to 1 should be interpreted as a *good* result, when large datasets are considered. Indeed, this means that the sanitized matrix $\tilde{A}$ is as close from A as two matrix queries over random samples of individuals both of the same size, and thus that $\tilde{A}$ captures statistical information contained in matrix A.

Call Details Records. Our dataset is composed of Call Details Records from a major telecom operator. In particular, CDRs contain timestamps and location of mobile phone calls (in terms of the antennas in which the calls transit). From these data, we can build the mobility matrices of users as explained in Example A. Afterwards, we count the number of *transitions* between antennas over a period of two weeks. The total number n of antennas is equal to 1600 in our dataset.

B. Results and analysis

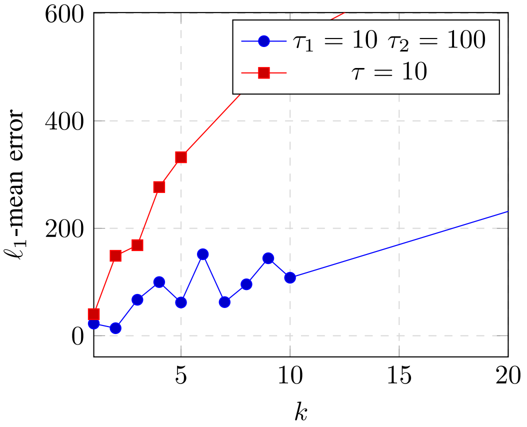
Experiments were implemented in Scala programming language using the BigData library Spark. The following results are obtained from CDRs of a population of $N = 33000$ mobile phone users. A significant convergence of $(d_l)_l$ is obtained for $l \sim 10$. For our dataset and queries, low values for parameter K are sufficient for Block Laplacian mechanism to outperform standard Laplacian mechanism (more precisely, $K = 2$ with $\tau = 10$, or $K = 3$ with $\tau_1 = 10, \tau_2 = 100$). We believe that for other applications with a more complex graph structure, larger values of K allow for even better improvements. We give details in the Appendix on how we proceed to choose K, which follows the heuristic introduced in Section III-B. Our experiments show that *block-sensitivities* mechanisms outperform their *global-sensitivity* analogs. However, results of BL are still far from being reasonable since we obtain in the best cases $d(A, \tilde{A}) \sim 2.10^3$ (see the Appendix for the display of results).

BLSVD for $\epsilon = 0.1$, $K = 2$ and $\tau = 10$



By contrast, the algorithms $LSVD$ and $BLSVD$ reach some admissible values (that is $d(A, \tilde{A}) \sim 1$), while providing a relatively high level of privacy ($\epsilon = 0.1$). Moreover, when k is chosen sufficiently small, the error is very close to that of k-rank approximation without noise perturbation (i.e, $d(A, \tilde{A}) \sim d(A, A_k)$). This means that the spectral information of a sanitized matrix (namely the smallest eigenvalues and their respective eigenvectors) is statistically close (in the sense of distance d) to that of an unperturbed matrix. Such results were already obtained in [22], using $LSVD$. Our experiments show that for all values of k, $BLSVD$ outperforms $LSVD$, which means that results from [22] can be improved by using block-sensitivities mechanisms.

BLSVD for $\epsilon = 0.1$, $K = 3$ VS $K = 2$



VI. Related work

To the best of our knowledge, the closest idea in spirit with our algorithms is the *matrix mechanism* introduced in [19]. For both our works, the aim is to optimize the privacy budget on coefficients. While optimizations from [19] need to solve rank-constrained semi-definite program, our distribution of noise among coefficients relies only on a theoretical result given in Theorem 5 (at least when the graph structure is well-understood by the sanitizer). Even if no knowledge on coefficients sensitivities is available, we propose in Section IV an efficient framework to apply our techniques. Compared to [19], we also use a much more general neighbouring relation, allowing for more applications on real-datasets.

The idea of using SVD as a post-processing of an *additive noise* mechanism appeared first in [3]. Other related techniques can be found in [4] and [8], and an application to recommendation systems was done in [22]. As mentioned before, algorithm BLSVD uses idea from [3] combined with our new *additive noise* mechanism, and outperforms (at least on the data we considered) prior techniques as explained in Section V. In particular, the authors of [22] proved that a combination of Laplacian mechanism and k-rank approximation may be used to release differentially private recommendation systems with reasonable accuracy. Since the combination of Block Laplacian mechanism and k-rank approximation results in a better accuracy for a same level of noise, our algorithms can a fortiori be used to sanitize recommendation systems.

Differentially private graphs were studied by various authors [1], [13], [14], [24], [32]. Most of these previous works aim at releasing graph statistics in a differentially private manner, and to obtain (if needed) a synthetic graph by sampling from these private statistics using ad-hoc techniques such as the Kronecker model [25], [15] or the exponential model [16], [21]. However, these sampling techniques do not fit with many real-life situations, such as for instance with the example that has motivated this work. Moreover, we point out that our methods are much more flexible regarding possible neighbouring notions of privacy (see Section II-B for a comparison with edge and node privacy from [14] and [13]).

Authors of [33] focused on privacy-preserving spectral graph analysis, which aims at publishing private eigenvectors and eigenvalues of the adjacency matrix. Other efforts were made in [11] and [17] to give theoretical bounds for differentially private spectral theory. The main drawback of the latter techniques is a lack of control on *neighbour* spectral projections (see for instance the bounds from Theorems 4 and 6 in [8]). For this reason, we prefer to adopt a rank k-approximation post-processing, which was already used successfully on real-datasets in [22].

VII. Conclusion

In this paper, we introduce new methods for releasing differentially private graphs that are based on a new way to distribute noise among edges weights. In addition, the generic privacy framework that we propose can capture all privacy notions introduced so far in the literature to release graphs in a differentially private manner. Experimental results on real datasets show that our methods outperform the standard existing techniques in particular with respect to utility.

References

[1] F. Ahmed, R. Jin and A. X. Liu. A random matrix approach to differential privacy and structure preserved social network graph publishing. *arxiv:*1307.0475, 2013.

[2] J. Blocki, A. Blum, A. Datta and O. Sheffet. The Johnson-Lindenstrauss transform itself preserves differential privacy. *Foundations of Computer Science (FOCS). IEEE 53rd Annual Symposium. IEEE*, p 410-419, 21012.

[3] A. Blum, C. Dwork, F. McSherry and K. Nissim. Practical privacy: the SuLQ framework. *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database system. ACM.* p 128-138, 2005.

[4] K. Chaudhuri, A. Sarwate and K. Sinha. Near-optimal differentially private principal components. *Advances in Neural Information Processing Systems*, p 989-997, 2012.

[5] C. Dwork. Differential privacy: A survey of results. *Theory and Applications of Models of Computation*, p 1-19, 2008.

[6] C. Dwork, F. Mc Sherry, K. Nissim and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, p 265-284, 2006.

[7] C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy.

[8] C. Dwork, K. Talwar, A. Thakurta and L. Zhang. Analyze Gauss: optimal bounds for privacy-preserving principal component analysis. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing. ACM*, p 11-20, 2014.

[9] S. Gambs, M.O. Killijian and M.N. del Prado Cortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 80(8), p. 1597-1614, 2014.

[10] J. He, W.W. Chu and Z.V. Liu. Inferring privacy information from social networks. *Intelligence and Security Informatics. Springer Berlin Heidelberg.* p. 154-165, 2006.

[11] M. Hardt and A. Roth. Beyond worst-case analysis in private singular vector computation. *Proceedings of the Forty-Fifth annual ACM Symposium on Theory of Computing. ACM*, p 331-340, 2013.

[12] A. Korolova. Privacy violations using microtargeted ads: A case study. *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE*, p 474-482, 2010.

[13] S. P. Kasivisiwanathan, K. Nissim, S. Raskhodnikova and A. Smith. Analyzing graphs with node differential privacy. *Theory of Cryptography, Springer Berlin Heidelberg*, p 457-476, 2013.

[14] V. Karwa, S. Raskhodnikova, A. Smith and G. Yaroslavtsev. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, vol. 4, no. 11, p 1146-1157, 2011.

[15] V. Karwa, S. Raskhodnikova, A. Smith and G. Yaroslavtsev. Private analysis of graph structure. *ACM Transactions on Database Systems (TODS)*, vol. 39, no. 3, p 22, 2014.

[16] V. Karwa, A. B. Slavković and P. Krivitsky. Differentially private exponential random graphs. *Privacy in Statistical Databases. Springer International Publishing*, p 143-155, 2014.

[17] M. Kapralov and K. Talwar. On differentially private low rank approximation. *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM*, p 1395-1414, 2013.

[18] K. Kenthapadi, A. Korolova, I. Mironov and N. Mishra. Privacy via the Johnson-Lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5, 2013.

[19] C. Li, M. Hay, V. Rastogi, G. Miklau and A. McGregor. Optimizing linear counting queries under differential privacy. *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM*, p 123-134, 2010.

[20] J. Lindamood, R. Heatherly, M. Kantarcioglu and B. Thuraisingham. Inferring private information using social network data. *Proceedings of the 18th international conference on World wide web. ACM*, p 1145-1146, 2009.

[21] W. Lu and G. Miklau. Exponential random graph estimation under differential privacy. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*, p 921-930, 2014.

[22] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM.*, p 627-636, 2015.

[23] A. Mislove, B. Viswanath, K.P. Gummadi and P. Druschel. You are who you know: inferring user profiles in online social networks. *Proceedings of the third ACM international conference on Web search and data mining. ACM.* p. 251-260, 2010.

[24] D. J. Mir and R. N. Wright. A differentially private graph estimator. *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on.* IEEE, p 122-129, 2009.

[25] D. J. Mir and R. N. Wright. A differentially private estimator for the stochastic Kronecker graph model. *Proceedings of the 2012 Joint EDBT/ICDT Workshops. ACM*, p 167-176, 2012.

[26] A. Narayanan and V. Shmatikov. De-anonymizing social networks. *Security and Privacy, 30th IEEE Symposium on. IEEE.* p. 173-187, 2009.

[27] M. E. J. Newman and M. Girvman. Finding and evaluating community structure in networks. *Physical Review E*, vol. 69, no. 2, p 026113, 2004.

[28] K. Nissim, S. Raskhodnikova and A. Smith. Smooth sensitivity and sampling in private data analysis. *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, p 75-84 , 2007.

[29] A. Sealfon. Shortest Paths and Distances with Differential Privacy. arXiv preprint arXiv:1511.04631. 2015.

[30] M. Srivatsa and M. Hicks. Deanonymizing mobility traces: Using social network as a side-channel. *Proceedings of the 2012 ACM conference on Computer and communications security. ACM*, p. 628-637, 2012.

[31] G. W. Stewart and J. Sun. Matrix perturbation theory. *Academic Press, San Diego*, 1990.

[32] Y. Wang and X. Wu. Preserving differential privacy in degree-correlation based graph generation. *Transactions on Data Privacy*, vol. 6, no. 2, p 127, 2013.

[33] Y. Wang, X. Wu and L. Wu. Differential privacy-preserving spectral graph analysis. *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, p 329-340, 2013.

[34] S. Wasserman and K. Faust. Social network analysis: Methods and applications. *Cambridge University Press*, 1994.

[35] W. Xu, X. Zhou and L. Li. Inferring privacy information via social relations. *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on. IEEE*, p 525-530, 2008.

[36] J. Zhang, G. Cormode, C.M. Procopiuc, D. Srivastava and X. Xiao. Private release of graph statistics using ladder functions. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, p. 731-745 (2015).

[37] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. *Proceedings of the 18th international conference on World wide web. ACM.*, p. 531-540, 2009.

VIII. Appendix

In this appendix, we provide the proofs of our theorems, theoretical results on Block Gaussian mechanism (the analog of Block Laplacian mechanism with Gaussian random variables), the pseudo-codes of our algorithms as well as more results about our experiments.

A. *Proofs*

In the following, we use the notation g_Z to denote the distribution of a random variable Z. First we recall the design of the standard Laplacian mechanism for matrices before giving its proof (to be compared to the proof of Theorem 5).

Theorem 8 (Laplacian mechanism for matrices [3]): Let $\epsilon > 0$ be the privacy parameter and $\lambda = \frac{\epsilon}{\Delta^{\ell_1}}$. The Laplacian mechanism for matrices is defined as $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, $A \mapsto \tilde{A}$ by

$$\tilde{A} = A + B$$

in which B is a random matrix such that coefficients $(B_{ij})_{ij}$ are independent random variables chosen as follows:
- $B_{ij} = 0$ for $(i,j) \notin \mathcal{S}$.
- B_{ij} is a Laplacian random variable with mean 0 and standard deviation $\sigma = \frac{\sqrt{2}}{\lambda}$ for $(i,j) \in \mathcal{S}$. In this case, mechanism $\mathcal{A}$ is ϵ-differentially private.

Proof of Theorem 8. By definition, the mechanism $\mathcal{A}$ is ϵ-differentially private if and only if for all matrices $A' \sim A$, and all subsets $S = (S_{ij})_{ij} \in \mathbb{R}^{n^2}$, we have

$$\frac{\mathbb{P}(\tilde{A} \in S)}{\mathbb{P}(\tilde{A'} \in S)} \leq e^{\epsilon}.$$

Since the $\tilde{A}_{ij}$ are independent random variables, and $\tilde{A}_{ij} = A_{ij} = {A_{ij}}'$ for $(i,j) \notin \mathcal{S}$, we need to show

$$\prod_{(i,j)\in\mathcal{S}} \frac{\mathbb{P}(\tilde{A}_{ij} \in S_{ij})}{\mathbb{P}(\tilde{A'}_{ij} \in S_{ij})} \leq e^{\epsilon} \text{ for all } S = (S_{ij})_{ij} \in \mathbb{R}^{n^2}.$$

We have $g_{\tilde{A}_{ij}}(y) = \mu \times e^{-\lambda|y-A_{ij}|}$ and $g_{\tilde{A'}_{ij}}(y) = \mu \times e^{-\lambda|y-{A'}_{ij}|}$ (μ being the relevant normalization coefficient). The previous condition on probabilities is equivalent to the following condition expressed in terms of the relevant distribution:

$$\prod_{(i,j)\in\mathcal{S}} \frac{g_{\tilde{A}_{ij}}(y_{ij})}{g_{\tilde{A'}_{ij}}(y_{ij})} \leq e^{\epsilon} \text{ for all } (y_{ij}) \in \mathbb{R}^{n^2}.$$

Using the triangle inequality, the following holds

$$\begin{aligned}\frac{g_{\tilde{A}_{ij}}(y_{ij})}{g_{\tilde{A'}_{ij}}(y_{ij})} &= e^{-\ \lambda\times(|y-A_{ij}|-|y-{A'}_{ij}|})\\ &\leq e^{-\lambda\times|A_{ij}-{A'}_{ij}|}.\end{aligned}$$

Hence, from the definition of sensitivity Δ^{ℓ_1}, it follows that

$$\begin{aligned}\prod_{(i,j)\in\mathcal{S}} \frac{g_{\tilde{A}_{ij}}(y_{ij})}{g_{\tilde{A'}_{ij}(y_{ij})}} &\leq e^{\lambda\times\sum_{(i,j)\in\mathcal{S}}|A_{ij}-{A'}_{ij}|}\\ &\leq e^{(\epsilon/\Delta^{\ell_1})\times\sum_{(i,j)\in\mathcal{S}}|A_{ij}-{A'}_{ij}|}\\ &\leq e^{\epsilon}.\end{aligned}$$

This completes the proof.

Now we provide the proof regarding Block Laplacian mechanism.

Proof of Theorem 5. First, we set $\epsilon_k = \frac{\epsilon\times\sqrt{n.k\Delta_k}}{\sum_{j=1}^{K}\sqrt{n_j\Delta_j}}$ so that $\lambda_k = \frac{\epsilon_k}{\Delta_k}$. From this, it is clear that $\sum_{k=1}^{K}\epsilon_k = \epsilon$. The proof of privacy of Theorem 5 is similar to that of Theorem 8, in which we replace λ by the relevant λ_{ij}. This change occurs at the end of the computation in the following manner.

$$\begin{aligned}\prod_{(i,j)\in\mathcal{S}} \frac{g_{\tilde{A}_{ij}}(y_{ij})}{g_{(\tilde{A'}_{ij}}(y_{ij})} &\leq \prod_{(i,j)\in\mathcal{S}} e^{\lambda_{ij}\times|A_{ij}-{A'}_{ij}|}\\ &= e^{\sum_{(i,j)\in\mathcal{S}}\lambda_{ij}\times|A_{ij}-{A'}_{ij}|}\\ &= e^{\sum_{k=1}^{K}\lambda_k\times\sum_{(i,j)\in S_k}|A_{ij}-{A'}_{ij}|}\\ &\leq e^{\sum_{k=1}^{K}\lambda_k\times\Delta_k}\\ &= e^{\sum_{k=1}^{K}\epsilon_k}\\ &\leq e^{\epsilon}.\end{aligned}$$

Hence $\mathcal{A}$ is ϵ-differentially private.

We are left with proving the second assertion of Theorem 5. To realize this, we need to prove that our choice of $(\lambda_k)_k$ (or equivalently $(\epsilon_k)_k$) minimizes the ℓ_1 mean-error on coefficients err $:= \mathbb{E}(\sum_{ij}|A_{ij} - \tilde{A}_{ij}|)$. First note that the following equalities hold:

$$\begin{aligned}\text{err} &= \sum_{(i,j)\in\mathcal{S}} \mathbb{E}(|B_{ij}|)\\ &= \sum_{k=1}^{K}\sum_{(i,j)\in S_k} \mathbb{E}(|B_{ij}|)\\ &= \sum_{k=1}^{K} n_k \times \frac{\Delta_k}{\epsilon_k}.\end{aligned}$$

It is now easy to show that our choice of $(\epsilon_k)_k$, that is $\epsilon_k = \frac{\epsilon\times\sqrt{n_k\Delta_k}}{\sum_{j=1}^{K}\sqrt{n_j\Delta_j}}$, minimizes the functional $\varphi(\epsilon_1, ...\epsilon_K) = \sum_{k=1}^{K} n_k \times \frac{\Delta_k}{\epsilon_k}$ under the constraint $\overline{\varphi}(\epsilon_1, ...\epsilon_K) = \sum_{k=1}^{K}\epsilon_k = \epsilon$. Using Lagrange multipliers, a local extremum for φ satisfies $\text{grad}\varphi(\epsilon_1, ...\epsilon_K) = \mu \times \text{grad}\overline{\varphi}(\epsilon_1, ...\epsilon_K)$ for some scalar μ. This equation together with the constraint gives the form of $(\epsilon_k)_k$ as stated in Theorem 5. In particular, such an extremum is unique and it is obviously a minimum, which concludes the proof.

In the following, we give proof of Theorem 6.

Proof of Theorem 6. The proof goes as the classical proof for Laplacian mechanism, once noticed that $|f_{x_0}(x) - f_{x_0}(x')| \leq \Delta_{x_0}$ for all $x \sim x'$, $x, x' \in \mathcal{D}$. It is clear that the latter inequalities hold by considering the two possible cases for $x \sim x'$:

1) $|f(x) - f(x')| = |f(x_I)| \leq \Delta_{x_0}$, then $f_{x_0}(x) - f_{x_0}(x') = f_{x_0}(x_I) = f(x_I)$.
2) $|f(x) - f(x')| = |f(x_I)| > \Delta_{x_0}$, then $f_{x_0}(x) - f_{x_0}(x') = f_{x_0}(x_I) = f(x_{I_0})$.

The proof of Theorem 7 is simply a combination of the previous proofs of Theorem 5 and Theorem 6.

Proof of Theorem 7. Let two neighbouring matrices $A \sim A'$ and let $(i,j) \in S_{x_0,k}$. By the linearity of coefficient-queries ψ_{ij}, we have $|A_{x_0,ij} - A'_{x_0,ij}| = |A^I_{x_0,ij}|$ for some user I. The latter quantity is always bounded by $\Delta_{x_0,k}$ by the definition of the truncation operation $A \mapsto A_{x_0}$. Set $\epsilon_{x_0,k} = \frac{\epsilon \times \sqrt{n_{x_0,k}\Delta_{x_0,k}}}{\sum_{j=1}^{K}\sqrt{n_{x_0,j}\Delta_{x_0,j}}}$ so that $\lambda_{x_0,k} = \frac{\epsilon_{x_0,k}}{\Delta_{x_0,k}}$. Using notations from Section IV-B $\tilde{A} = \mathcal{A}_{x_0} + B$, the following equalities hold:

$$\begin{aligned}\prod_{(i,j)\in\mathcal{S}} \frac{g_{\tilde{A}_{ij}}(y_{ij})}{g_{\tilde{A'}_{ij}}(y_{ij})} &\leq \prod_{(i,j)\in\mathcal{S}} e^{\lambda_{x_0,ij} \times |A_{x_0,ij} - {A'}_{x_0,ij}|} \\ &= e^{\sum_{(i,j)\in\mathcal{S}} \lambda_{x_0,ij} \times |A_{x_0,ij} - {A'}_{x_0,ij}|} \\ &= e^{\sum_{k=1}^{K} \lambda_{x_0,k} \times \sum_{(i,j)\in S_{x_0,k}} |A_{x_0,ij} - {A'}_{x_0,ij}|} \\ &\leq e^{\sum_{k=1}^{K} \lambda_{x_0,k} \times \Delta_{x_0,k}} \\ &= e^{\sum_{k=1}^{K} \epsilon_{x_0,k}} \\ &\leq e^{\epsilon}.\end{aligned}$$

Thus, $\mathcal{A}_{x_0}$ is ϵ-differentially private.

B. Designing blocks partition from coefficient sensitivities

Hereafter, we provide more details on the algorithm introduced in Section III-B to design a partition $(S_k)_k$ from sensitivities Δ_{ij}. For two given thresholds τ_1 and τ_2, it is easy to compare two partitions P_{τ_1}, P_{τ_2} once a norm is fixed to measure the output error. Indeed, the choice of the best partition should minimize the average error among all possible partitions. In the case of the Block Laplacian mechanism with $K = 2$ and the ℓ_1-mean error as a measure on outputs, the target function to minimize is $F(n_1, n_2, \Delta_1, \Delta_2) = \sqrt{n_1\Delta_1} + \sqrt{n_2\Delta_2}$ ($\sqrt{n_1\Delta_1} + \sqrt{n_2\Delta_2}$ is precisely the ℓ_1-mean error on coefficients of Block Laplacian mechanism). To automate the search of the best partition P_τ, we propose the following algorithm.

```
Search for 2-blocks partition
Input: Possible thresholds $\tau_1, \tau_2, ...\tau_r \in [0, \Delta]$,
error function $F$ to minimize
Output: Index $i$ of the best partition $P_{\tau_i}$ for Block noisy
(e.g., Laplacian or Gaussian) mechanism
1. Compute $\Delta_{ij}$ for all $i, j$
2. For $k$ in $1 : r$ do
3.   Compute $P_{\tau_k} = (S_1, S_2)$
4.   Compute $(\Delta_1, \Delta_2)$ associated to $P_{\tau_k}$
5.   Compute $F(n_1, n_2, \Delta_1, \Delta_2)$
     and denote by $T_k$ the result
6. Return the index of the minimal value in
   $T = [T_1, ...T_r]$.
```

It is straightforward to generalize the previous search for 2-blocks partition into a search for K-blocks partition, for $K > 2$. Indeed, given the thresholds $\tau^1, \tau^2, ...\tau^{K-1}$, we define the partition $P_{(\tau^i)_i} = (S_1, ...S_K)$ as follows:

$$\begin{aligned} S_1 &= \{\ (i,j) \mid \Delta_{ij} \leq \tau^1\ \} \\ S_2 &= \{\ (i,j) \mid \tau^1 < \Delta_{ij} \leq \tau^2\ \} \\ &\cdots \\ S_K &= \{\ (i,j) \mid \tau^{K-1} < \Delta_{ij}\ \}. \end{aligned}$$

Note that the algorithm Search for 2-blocks partition can be made much more efficient by using a dichotomous search of the index, instead of an exhaustive look at the thresholds $\tau^1, ...\tau^{K-1}$.

C. Block Gaussian mechanism

As mentioned in the core of the paper, block noisy mechanisms may be designed with other random variables than Laplacian ones. For instance, the results of the current section shows that one can use random Gaussian variables calibrated to block-sensitivities.

It appears that differentially private mechanisms based on Gaussian random variables satisfy a slightly weaker guarantee of privacy than ϵ-differential privacy, which is called (ϵ, δ)-differential privacy.

Definition 9: (**(ϵ, δ)-differential privacy**, [6]) A randomized mechanism $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is said to be (ϵ, δ)-differentially private if for all $A, A' \in \mathbb{R}^{n\times n}$, $A \sim A'$, and all $E \subset \mathbb{R}^{n\times n}$, we have:

$$\mathbb{P}(\mathcal{A}(A) \in E) \leq e^{\epsilon}\mathbb{P}(\mathcal{A}(A') \in E) + \delta.$$

The advantage of using Gaussian random variables in a differentially private mechanism instead of Laplacian random variables is that the mechanism can be calibrated to the ℓ_2-sensitivity Δ^{ℓ_2} on coefficients, rather than the ℓ_1-sensitivity Δ^{ℓ_1}.

Definition 10: (**Δ^{ℓ_2}-sensitivity for matrices**) The ℓ_2-sensitivity for matrices $\Delta^{\ell_2}_S$ for the block $S \subset \mathcal{S}$ is given by the formula

$$\Delta^{\ell_2}_S = \max_{A\sim A'} \sqrt{\sum_{(i,j)\in S} |A_{ij} - A'_{ij}|^2}$$

in which the max is taken over all pairs of neighbours $A \sim A'$. For $S = \mathcal{S}$, $\Delta^{\ell_2} = \Delta^{\ell_2}_S$ is simply called the ℓ_2-sensitivity.

Block ℓ_2-sensitivities can be defined in the same manner, by restricting the sum to the indices appearing in the corresponding block. For any $S \subset \mathbb{R}^{n\times n}$, we always have $\Delta^{\ell_2}_S \leq \Delta^{\ell_1}_S$ and in higher dimensions (that is for $|S| \geq 2$), sensitivity $\Delta^{\ell_2}_S$ can be much smaller than $\Delta^{\ell_1}_S$. Hence, by using Gaussian random variables instead of Laplacian random variables, one can hope for a much more precise model while incurring only a small loss in privacy.

The following theorem is the matrix version of the Gaussian mechanism used so far in the literature.

Theorem 11: (**Gaussian mechanism for matrices**, [8]) Let $\epsilon, \delta > 0$ be the privacy parameters, and $\mathcal{S}$ be the set of sensitive coefficients and let $\frac{1}{\sigma} = \sqrt{\lambda} = \frac{\epsilon}{\Delta^{\ell_2} \times \sqrt{2\ln(\frac{1.25}{\delta})}}$. The

Gaussian mechanism for matrices is defined as $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}, A \mapsto \tilde{A}$ by

$$\tilde{A} = A + B$$

in which B is a matrix whose coefficients are independent random variables chosen as follows:
- $B_{ij} = 0$ for $(i,j) \notin \mathcal{S}$;
- B_{ij} is a centered Gaussian random variable with standard deviation σ for $(i,j) \in \mathcal{S}$.
Then mechanism $\mathcal{A}$ is (ϵ,δ)-differentially private.

Like Laplacian random variables, Gaussian random variables can also be calibrated to block (ℓ_2-) sensitivities. We have designed the amplitudes of noise to minimize the ℓ_1 mea- error on coefficients. This choice of ℓ_1-norm instead of ℓ_2-norm allow us to compare Block Gaussian mechanism to Block Laplacian mechanism in the experimental part of the paper.

Theorem 12: (**Block Gaussian mechanism**) Let $\epsilon, \delta > 0$, and let $(S_k)_{1\leq k\leq K}$ be a partition of the set of sensitive coefficients $\mathcal{S}$. We define

$$\frac{1}{\sigma_k} = \sqrt{\lambda_k} = \frac{1}{\Delta_k^{\ell_2}} \times \frac{\epsilon_k}{\sqrt{2\ln(\frac{1.25}{\delta_k})}}$$

in which $(\delta_k)_k$ and $(\epsilon_k)_k$ satisfy the following conditions:
- $\prod_{k=1}^K (1-\delta_k) \geq 1-\delta$;
- $\sum_{k=1}^K \epsilon_k = \epsilon$.
The Block Gaussian mechanism $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}, A \mapsto \tilde{A}$ is defined as

$$\tilde{A} = A + B$$

in which the coefficients of matrix B are independent random variables given by:
- $B_{ij} = 0$ if $(i,j) \notin \mathcal{S}$;
- B_{ij} is a Gaussian random variable with standard deviation σ_k if $(i,j) \in S_k$.
Then mechanism $\mathcal{A}$ is (ϵ,δ)-differentially private. Moreover, if we set $\mu_k = n_k \times \Delta_k^{\ell_2} \times \sqrt{2\ln(\frac{1.25}{\delta_k})}$, then the choice $\epsilon_k = \frac{\epsilon \times \sqrt{\mu_k}}{\sum_{j=1}^K \sqrt{\mu_j}}$ realizes the minimum of the ℓ_1-mean error function

$$\varphi(\epsilon_1, \epsilon_2, ...) = \frac{1}{\sqrt{\pi}} \times \sum_{k=1}^K \frac{\mu_k}{\epsilon_k}$$

under the constraint that $\epsilon = \sum_{k=1}^K \epsilon_k$.

Proof of Theorem 12. To prove that our mechanism $\mathcal{A}$ is (ϵ,δ)-differentially private, it is sufficient to show that for all $A \in \mathbb{R}^{n\times n}$, there exist subsets $E_A \subset \mathbb{R}^{n\times n}$ such that:
(1) $\frac{g_{\tilde{A}}(Y)}{g_{\tilde{A}'}(Y)} \leq e^{\epsilon}$ for all $Y \in E_A$, $A' \sim A$;
(2) $\mathbb{P}(\tilde{A} \notin E_A) \leq \delta$.

We will prove the conditions (1) and (2) with the subsets E_A described as follows:

$$E_A = \cap_{1\leq k\leq K} E_{A,k}$$

in which

$$\begin{aligned} E_{A,k} = \{ \ & Y = (Y_{ij})_{ij} \in \mathbb{R}^{n\times n} \ | \\ & \lambda_k (\Delta_k^{\ell_2})^2 + (2\lambda_k \Delta_k^{\ell_1} \times \max_{(i,j)\in S_k} |A_{ij} - Y_{ij}|) \leq \epsilon_k \}. \end{aligned}$$

First, we prove that condition (1) is satisfied with our choice of subsets E_A, $A \in \mathbb{R}^{n\times n}$. Indeed, we can write each distribution of the ratio $\frac{g_A(Y)}{g_{A'}(Y)}$ as follows:

$$\begin{aligned} g_{\tilde{A}}(Y) &= \prod_{(i,j)\in\mathcal{S}} g_{\tilde{A}_{ij}}(Y_{ij}) \\ &= \prod_{k=1}^K \prod_{(i,j)\in S_k} e^{-\lambda_k |A_{ij} - Y_{ij}|^2} \\ &= e^{\sum_{k=1}^K \sum_{(i,j)\in S_k} -\lambda_k |A_{ij} - Y_{ij}|^2} \end{aligned}$$

Afterwards, condition (1) follows from the assumption $\sum_{k=1}^K \epsilon_k = \epsilon$, and the following inequalities that hold for all $A \sim A'$, $1 \leq k \leq K$:

$$\begin{aligned} & \sum_{(i,j)\in S_k} \lambda_k \times |(A_{ij} - Y_{ij})^2 - (A'_{ij} - Y_{ij})^2| \\ & \leq \sum_{(i,j)\in S_k} \lambda_k \times (|A_{ij} - A'_{ij}|^2 \ + \\ & 2 \times |A_{ij} - A'_{ij}| \times |A_{ij} - Y_{ij}|) \\ & \leq \lambda_k (\Delta_k^{\ell_2})^2 \ \ + (2\lambda_k \Delta_k^{\ell_1} \times \max_{(i,j)\in\mathcal{S}} |A_{ij} - Y_{ij}|) \end{aligned}$$

Afterwards, we prove that condition (2) is satisfied. First remark that we have $\mathbb{P}(\tilde{A} \in E_{A,k}) \geq 1 - \delta_k$ for all $1 \leq k \leq K$, by Theorem A.1 p261 in [7] (in which dimension d of the range space is n_k for our proof) and our choice of σ_k. Moreover, by our assumption on $(\delta_k)_k$ the following inequalities hold:

$$\begin{aligned} \mathbb{P}(\tilde{A} \notin E_A) &= 1 - \mathbb{P}(\tilde{A} \in E_A) \\ &= 1 - \prod_{k=1}^K \mathbb{P}(\tilde{A} \in E_{A,k}) \\ &\leq 1 - \prod_{k=1}^K (1 - \delta_k) \\ &\leq \delta. \end{aligned}$$

Thus condition (2) holds as well, which finishes the proof of the privacy statement of Theorem 12.

The proof of the second statement of Theorem 12 goes exactly as the similar proof of optimization under constraint used for Theorem 5, using that $\mathbb{E}(|Z|) = \frac{1}{\sqrt{\pi}\times\sqrt{\lambda}}$ for a Gaussian random variable of standard deviation $\sigma = \frac{1}{\sqrt{\lambda}}$.

A possible admissible choice of parameters is $\delta_k = \delta_1$ for all $1 \leq k \leq K$. Notice that we have $\delta_k \sim \frac{\delta}{K}$, for small values of the parameter δ and K small enough ($K \leq 3$ is used in our experiments). Hence for a sufficiently small parameter $\delta > 0$, the choice $\delta_k = \frac{2\times\delta}{K}$ is an admissible choice of $(\delta_k)_k$ ito achieve (ϵ,δ)-differential privacy.

Remark that in general, we have $\Delta_S^{\ell_2} < \Delta_S^{\ell_1}$ for a subset $S \subset \mathcal{S}$, whereas we always have $\Delta_{ij}^{\ell_2} = \Delta_{ij}^{\ell_1}$. In particular, our algorithm Search K-Blocks Partition is relevant for both algorithms Block Laplacian mechanism and Block Gaussian mechanism. As a consequence, a *good* choice for a partition $(S_k)_{1\leq k\leq K}$ of sensitive coefficients relative to Block Gaussian mechanism can still be obtained using Algorithm Search K-Blocks Partition. In our experiments, we made the choice of using the target function F that minimizes the ℓ_1-mean error on coefficients, since it enables us to compare experimentally Block Laplacian mechanism and Block Gaussian mechanism. In this situation, a formula for F is given by (see the second statement in Theorem 12),

$$F(n_1, n_2, ...\Delta_1, \Delta_2, ...) = \frac{1}{\sqrt{\pi} \times \epsilon} \times (\sum_{k=1}^{K} \sqrt{\mu_k})^2.$$

The Block Gaussian Algorithm is summarized in a pseudo-code form hereafter. Note that we can also apply some *hybrid* mechanisms. For instance, one could a Laplacian perturbation on coefficients in a subset S_2, and a Gaussian perturbation on coefficients in a subset S_1.

D. Pseudo-code of the algorithms

The following algorithm implements a version of Block Laplacian mechanism that aims to minimize the ℓ_1-mean error on coefficients. Recall that $\mathcal{S}$ is the set of sensitive coefficients, and n_k denotes the cardinality of the subset $S_k \subset \mathcal{S}$.

```
BlockLaplacian
Input: Matrix $A$, privacy parameter $\epsilon$,
partition $(S_k)_{1\leq k\leq K}$ of the set $\mathcal{S}$
Output: Matrix $\tilde{A}$ $\epsilon$-differentially private
1. Compute $\Delta_k^{\ell_1} = \max_{B\sim B'} \sum_{(i,j)\in S_k} |B_{ij} - B'_{ij}|$
2. Set $\lambda_k = \frac{\epsilon}{\Delta_k^{\ell_1}} \times \frac{1}{\sum_{j=1}^{K} \sqrt{\frac{n_j \Delta_j^{\ell_1}}{n_k \Delta_k^{\ell_1}}}}$
3. Sample $Z_k$, a 0-mean Laplacian random variable
   of standard deviation $\sigma_k = \frac{\sqrt{2}}{\lambda_k}$
4. Set $\tilde{A}_{ij} = A_{ij}$ for all $(i,j) \notin \mathcal{S}$
5. Set $\tilde{A}_{ij} = A_{ij} + Z_k$ for all $(i,j) \in S_k$
6. Output $\tilde{A}$
```

The following pseudo-code corresponds to the mechanism explained in Theorem 12 minimizing the ℓ_1-mean error on coefficients. Moreover, for simplicity, we state the algorithm for values of δ_k all equal to $\frac{2\times\delta}{K}$.

```
BlockGaussian
Input: Matrix $A$, privacy parameters $\epsilon$ and $\delta$,
partition $(S_k)_{1\leq k\leq K}$ of the set $\mathcal{S}$
Output: Matrix $\tilde{A}$ $(\epsilon,\delta)$-differentially private
1. Set $\Delta_k^{\ell_2} = \max_{B\sim B'} \sqrt{\sum_{(i,j)\in S_k} |B_{ij} - B'_{ij}|^2}$
2. Set $\mu_k = n_k \times \Delta_k^{\ell_2} \times \sqrt{2\ln(\frac{K\times 1.25}{2\times\delta})}$,
$\epsilon_k = \frac{\epsilon\times\sqrt{\mu_k}}{\sum_{j=1}^{K}\sqrt{\mu_j}}$,
and $\frac{1}{\sigma_k} = \frac{1}{\Delta_k^{\ell_2}} \times \frac{\epsilon_k}{\sqrt{2\ln(\frac{K\times 1.25}{2\times\delta})}}$
3. Sample $Z_k$, a 0-mean Gaussian random variable
   of standard deviation $\sigma_k$
4. Set $\tilde{A}_{ij} = A_{ij}$ for all $(i,j) \notin \mathcal{S}$
5. Set $\tilde{A}_{ij} = A_{ij} + \mathrm{Gauss}(\sigma_k)$ for all $(i,j) \in S_k$
6. Output $\tilde{A}$
```

Recall that rank k-approximation goes as follows. Let $A = UDV$ be a Singular Value Decomposition of some matrix A, in which U and V are unitary matrices and D is the diagonal of singular values $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. The k-rank approximation A_k of A is defined as $A_k = UD_kV$, in which the diagonal D_k is obtained from the singular values diagonal D by replacing the $n-k$ lowest singular values $\lambda_{k+1}, ...\lambda_n$ with 0. Simply applying rank k-approximation to the result of Block Laplacian mechanism could perturb a little bit the coefficients in the set of non-sensitive coefficients $\mathcal{S}^c$, which would destroy some useful information. This can be easily avoided by remembering the coefficients relative to $\mathcal{S}^c$, and by forcing the result to be unchanged on these coefficients after the k-rank approximation.

```
BlockLaplacianThenSVD
(resp. BlockGaussianThenSVD)
Input: Matrix $A$, privacy parameter $\epsilon$ (respectively
parameters $(\epsilon,\delta)$), approximation rank $k$
Output: Private matrix $\tilde{A}$ of rank $k$
1. Store the values $A_{ij}$, for $(i,j) \notin \mathcal{S}$
2. Apply BlockLaplacian($\epsilon$) (resp. BlockGaussian($\epsilon,\delta$))
to matrix $A$, and denote by $C = A + B$ the result
3. Compute the SVD $C = UDV$ of matrix $C$
4. Compute $D = C_k$ $k$-rank approximation $C_k = UD_kV$.
5. Set $\tilde{A}_{ij} = A_{ij}$ for $(i,j) \notin \mathcal{S}$,
   and $\tilde{A}_{ij} = D_{ij}$ for $(i,j) \in \mathcal{S}$
6. Output $\tilde{A}$
```

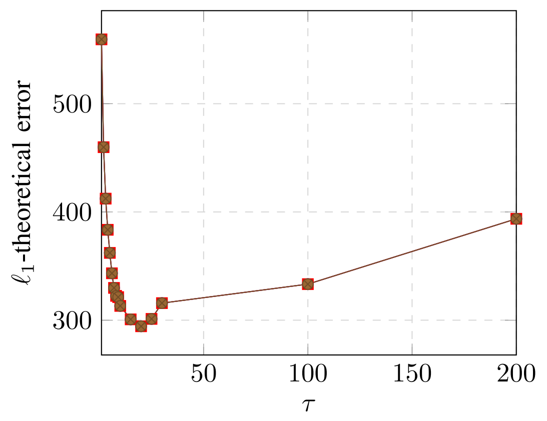
E. More experimental results

Finally, we provide additional results for our experiments, in particular for algorithm Search for partition and for Block Gaussian mechanism.

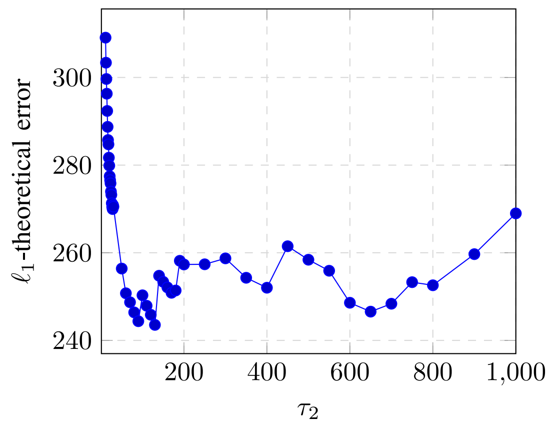
In the sequel, SP refers to the algorithm Search for (K-blocks) partition introduced in Section III-B. Given a number of partitions K and a target function F, this algorithm aims at computing an approximation of the best τ (or $(\tau_i)_i$ if $K > 2$). Our experiments on SP consider the target function F minimizing the ℓ_1-mean error for Block Laplacian mechanism as explained in Section II. For algorithm SP, we illustrate the choice of τ by drawing the dependence of the theoretical error of various mechanisms on the threshold τ.

For our dataset, and a choice of $K = 2$ partitions, optimal values for τ range between 10 and 20 for Block Laplacian mechanism and $\tau \sim 35$ for Block Gaussian mechanism. This result has two important consequences: the design of BL and BLSVD for $K = 2$, and the design of SP when we choose a larger $K > 2$ number of partitions. For instance when $K = 3$, we can chose $\tau_1 = 10$ and look at the variations of the error depending on the other threshold τ_2. However, the resulting curve is slightly more complex than for $K = 2$. Indeed, case $K = 3$ has more dependencies than case $K = 2$. In details, it depends on the cardinalities n_1, n_2 and n_3 of the elements of the partition S_1, S_2 and S_3, and their sensitivities $\Delta_1^{\ell_1}$, $\Delta_2^{\ell_1}$ and $\Delta_3^{\ell_1}$.

SP for $K = 2$, $\epsilon = 1$

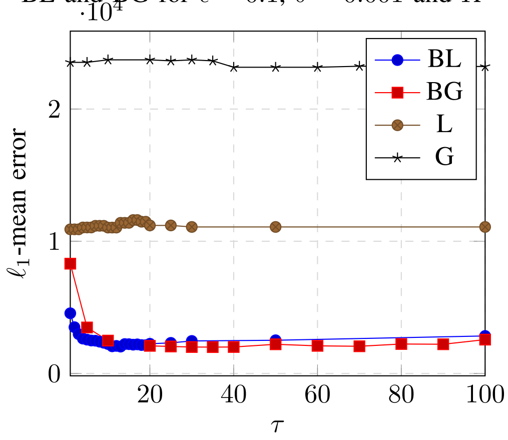


SP for $K = 3$, $\epsilon = 1$, $\tau_1 = 10$



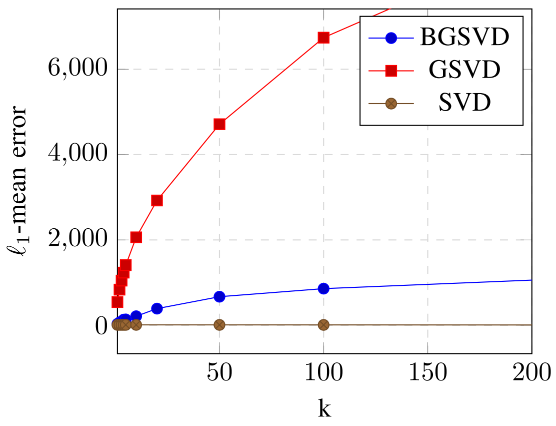
We now refer to Block Gaussian mechanism (resp. BlockGaussianThenSVD) by BG (respectively $BGSVD$). To compare *Gaussian type* algorithms to their Laplacian analogs, we use the ℓ_1-norm to measure all errors as defined in Section V.

BL and BG for $\epsilon = 0.1$, $\delta = 0.001$ and $K = 2$

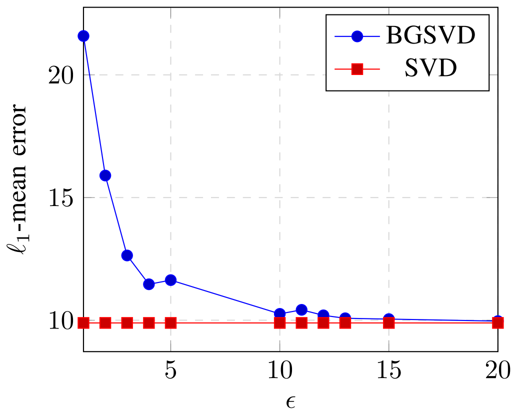


Unlike BL and BG, the dependence in ϵ for algorithms $BLSVD$ and $BGSVD$ is not linear. We illustrate this fact on our data in the following figures, and show more precisely the closeness between $BGSVD$ and unperturbed SVD (for a same rank k).

BGSVD for $\epsilon = 0.1$, $\delta = 0.001$, $K = 2$ and $\tau = 35$



BGSVD for $\delta = 0.001$, $k = 5$, $K = 2$ and $\tau = 35$



The curves above show that a significant spectral information (rank value $k = 5$) can be preserved using algorithm BlockGaussianThenSVD, while providing a high level of privacy (privacy parameters $\epsilon \sim 3$, $\delta \sim 0.001$).