Differential Privacy in distribution and instance-based noise mechanisms

Sébastien Canard and Baptiste Olivier
Orange Labs, Applied Crypto Group, Caen, France
{sebastien.canard, baptiste.olivier}@orange.com

Abstract

In this paper, we introduce the notion of (ϵ, δ)-differential privacy *in distribution*, a strong version of the existing (ϵ, δ)-differential privacy, used to mathematically ensure that private data of an individual are protected when embedded into a queried database. In practice, such property is obtained by adding some relevant noise. Our new notion permits to simplify proofs of (ϵ, δ) privacy for mechanisms adding noise with a continuous distribution. As a first example, we give a simple proof that the Gaussian mechanism is (ϵ, δ)-differentially private *in distribution*.
Using differential privacy *in distribution*, we then give simple conditions for an instance-based noise mechanism to be (ϵ, δ)-differentially private. After that, we exploit these conditions to design a new (ϵ, δ)-differentially private instance-based noise algorithm. Compare to existing ones, our algorithm have a better accuracy when used to answer a query in a differentially private manner.
In particular, our algorithm does not require the computation of the so-called Smooth Sensitivity, usually used in instance-based noise algorithms, and which was proved to be NP hard to compute in some cases, namely statistics queries on some graphs. Our algorithm handles such situations and in particular some cases for which no instance-based noise mechanism were known to perform well.

1 Introduction

1.1 Context and related work

One big concern in data publishing is the privacy of the individuals concerned with these data. As the opportunities and the means to release useful information from individual data (*a.k.a.* personal data) grow wider, the leakage of information threatens more and more these individuals. That is the reason why researchers have proposed several rigorous notions of privacy in the last few years and, among them, one of the most promising is *differential privacy*. This notion, usually referred as ϵ-differential privacy, was introduced by Dwork, McSherry, Nissim and Smith in [3]. It provides strong guarantees of privacy, controlled by the parameter ϵ, and effective even against adversaries having arbitrary side information. Informally speaking, a differentially private mechanism ensures that any of its outputs is essentially likely to occur, independent of the presence or absence of any individual in the database.
A common way to design an ϵ-differentially private randomized mechanism $\mathcal{A}$ is to add

noise to some query f on the considered database x: $\mathcal{A}(x) = f(x) + Z$ for some well-chosen random variable Z (independent of x). It may happen that some situations require noise which cannot provide ϵ-differential privacy, but satisfy some weaker notions of privacy. The most known and widely used weakening of ϵ-differential privacy was introduced in [2] and is called (ϵ, δ)-differential privacy, where δ is an additional approximation used to relax the strict relative shift in the case of events that are not especially likely to occur. The standard example satisfying (ϵ, δ)-differential privacy, but not ϵ-differential privacy, is the so-called Gaussian mechanism which already appeared in numerous designs of private mechanisms (see for instance [2], [6] or [12]).
The first contribution of this paper is to show how a simple condition, that we called (ϵ, δ)-differential privacy *in distribution*, can be used to give an easy framework for proofs of (ϵ, δ)-differential privacy. This notion appeared implicitly in (possibly all) proofs of (ϵ, δ)-differential privacy in the literature, but is not stated: we show on two common examples how proofs of (ϵ, δ)-differential privacy are easily obtained from this framework.
Another important family of (ϵ, δ)-differentially private mechanisms is given by *instance-based noise* mechanisms [13] which take the following form: $\mathcal{A}(x) = f(x) + Z_x$ for some random variable Z_x depending on the considered dataset x. In [13], the differentially private mechanism is calibrated to a new kind of sensitivity called *Smooth Sensitivity*, while previous algorithms were always designed with respect to global sensitivity (see Section 2 for formal definitions). Such schemes are widely used, in particular to release differentially private (statistics of) graphs (see [11], [7], [9], [14]). To the best of our knowledge, the only other instance-based noise technique that exists in the literature appeared in [9], where another kind of sensitivity (namely local sensitivity of higher order) was used to calibrate noise in a differentially private manner. In particular, the authors show that the Smooth Sensitivity of counts of k-triangles in graphs is NP-hard to compute (relative to edge-privacy). As an alternative, they designed an instance-based noise mechanism for such queries, that was differentially private and did not rely on the computation of Smooth Sensitivity.
Our main contribution in this paper is a new method to design instance-based noise differentially private algorithms. First, we use our new notion of (ϵ, δ)-differential privacy in distribution to give very simple conditions for an instance-based noise mechanism to be (ϵ, δ)-differentially private. Then these conditions are used to design an instance-based noise mechanism, for which we give an algorithm for practical implementations. This algorithm is very simple, guarantees (ϵ, δ)-differential privacy, and is easy to deploy in practice. Our result can really improve the accuracy of an answer to a query, compared to related work algorithms. It moreover handles new cases on which no instance-based noise differentially private mechanism were known to apply. Finally, we also discuss some typical cases to which our technique can be usefully applied.

1.2 Details on our contributions

To summarize, our contribution is two-fold: we first introduce the notion of (ϵ, δ)-differential privacy *in distribution*, and we then study instance-based noise mechanisms theoretically first, and thereafter on examples. More precisely, we have three important results:

• we give a precise definition of (ϵ, δ)-differential privacy *in distribution* (see Definition 8), and prove that it implies (ϵ, δ)-differential privacy. Existing proofs of (ϵ, δ)-differential privacy suffer from a lack of standardized method, leading sometimes to some confusion. Using (ϵ, δ)-differential privacy in distribution, we give a simple framework that can be applied to all existing proofs of (ϵ, δ)-differential privacy, and we illustrate this on two examples: Gaussian mechanism (see section 3.2) and instance-based noise mechanism (see section 4.1);
• using our new introduced notion, we give simple conditions for a Laplace instance-based noise mechanism to be (ϵ, δ)-differentially private (see Theorem 14). Then we use these conditions to design an algorithm (referred as Algorithm 1 in the sequel) which is (ϵ, δ)-differentially private and allows for significantly reducing noise compared to standard Laplace noise mechanism. Algorithm 1 only needs to compute (or approximate) a restricted number of local sensitivities, and not all. This is much better that the so-called Smooth Sensitivity used so far to design instance-based noise algorithms;
• Algorithm 1 can in fact be applied in many more situations than existing instance-based noise techniques. We then discuss specific examples where Algorithm 1 gives significantly less noise than existing work, and should be used. In particular, it is useful to release the number of k-triangles in a graph, even in cases where the graph is not too much connected. To the best of our knowledge, Algorithm 1 is the best algorithm to handle such situations, in terms of a trade-off utility/privacy.

1.3 Organization of the paper

In section 2, we recall relevant notions of differential privacy for this paper, and we state our notations. In section 3, we introduce and discuss our new notion of (ϵ, δ)-differential privacy in distribution. Section 4 is devoted to general privacy results for instance-based noise mechanisms. In section 5, we design our Algorithm 1 and discuss some possible applications. Complete proofs of all the results are finally given in the appendix.

2 Differential Privacy

In this section, we recall the relevant notions of differential privacy we will need in the sequel. We refer to [1] and [4] for basic notions about differential privacy.

2.1 Databases and neighbors

All along the paper, we will denote by $\mathcal{D}^n$ the database from which we want to extract data subsets $x, y \subset \mathcal{D}^n$. Given a query f defined on $\mathcal{D}^n$, our interest is to understand how to release privately some values $f(x)$, $f(y)$... In the sequel, n stands for the number of individuals in the database. Its value is necessary to tune the privacy parameters (see [5]).
Differential privacy relies on a notion of neighboring that can be defined as follows.

Definition 1 Let $\mathcal{D}^n$ be a database. Two sub-databases x, x' of $\mathcal{D}^n$ are said to be *adjacent* if they differ from each other by at most one individual. Then we denote $x \sim x'$.

We say that the database $\mathcal{D}^n$ is *connected* if for all $x, x' \subset \mathcal{D}^n$, there exist a sequence $(x_i)_{1\leq i\leq N}$ such that $x_1 = x$, $x_N = x'$ and

$$x_i \sim x_{i+1} \text{ for all } 1 \leq i \leq N-1.$$

In the sequel, we will always assume that the database $\mathcal{D}^n$ is connected. Our results are very general, so this is the only assumption on the database $\mathcal{D}^n$. For our purpose, this assumption can be removed since one can handle each connected component in parallel without loss of privacy (see for instance Theorem 4 in [10]).

2.2 (ϵ, δ)-differential privacy

The idea of differential privacy is the following: a randomized mechanism is ϵ-differentially private if adding or removing a single individual in the underlying database changes the probability of each mechanism output by at most a e^ϵ-factor. Here is the formal definition.

Definition 2 ([3]) Let $\mathcal{D}^n$ be a database. A randomized algorithm $\mathcal{A} : \mathcal{D}^n \to R$ is *ϵ-differentially private* if, for all subsets $S \subset R$, we have

$$\mathbb{P}(\mathcal{A}(x) \in S) \leq e^\epsilon \, \mathbb{P}(\mathcal{A}(x') \in S) \text{ whenever } x \sim x'.$$

This definition of ϵ-differential privacy is sometimes called *pure privacy*, by contrast with the following weaker notion of (ϵ, δ)-differential privacy, which adds some approximation δ.

Definition 3 ([2]) Let $\mathcal{D}^n$ be a database. A randomized algorithm $\mathcal{A} : \mathcal{D}^n \to R$ is *(ϵ, δ)-differentially private* if, for all subsets $S \subset R$, we have

$$\mathbb{P}(\mathcal{A}(x) \in S) \leq e^\epsilon \, \mathbb{P}(\mathcal{A}(x') \in S) + \delta \text{ whenever } x \sim x'.$$

In [8] and [5], it is required to have $\delta < \epsilon^2/n$ in order to satisfy some meaningful privacy, called semantic differential privacy.
For almost all differentially private mechanisms from the literature which have noise with continuous distributions (Laplace, Gaussian, ...), proofs of privacy relied on an estimation of the *privacy loss*

$$\frac{g_x(y)}{g_{x'}(y)} \text{ for all } x \sim x', y \in R,$$

where g_x denotes the distribution of $\mathcal{A}(x) = f(x) + Z$. Indeed, the quantity $g_x(y)/g_{x'}(y)$ is usually much easier to compute than the ratio $\mathbb{P}(\mathcal{A}(x) \in S)/\mathbb{P}(\mathcal{A}(x') \in S)$, when considering continuous distributions. Moreover, it appears that estimates on the privacy loss characterizes ϵ-differential privacy for such mechanisms. This motivates the following definition.

Definition 4 Let $\mathcal{A} : \mathcal{D}^n \to R$ be a randomized algorithm. Then $\mathcal{A}$ is said to be *ϵ-differentially private in distribution* if the following holds:

$$\frac{g_x(y)}{g_{x'}(y)} \leq e^\epsilon \text{ for all } x \sim x', y \in R.$$

As given by the following proposition (which proof is given in the appendices), this notion is exactly ϵ-differential privacy.

Proposition 5 *Let $\mathcal{A} : \mathcal{D}^n \to R$ be a randomized algorithm. Then $\mathcal{A}$ is ϵ-differentially private if and only if $\mathcal{A}$ is ϵ-differentially private in distribution.*

It is now tempting to define (ϵ, δ)-differential privacy in distribution as follows: control the privacy loss $g_x(y)/g_{x'}(y)$ by a factor e^ϵ for all y in some subset $E \subset R$ such that $\mathbb{P}(\mathcal{A}(x) \notin E)$ is small enough, controlled by the parameter δ. Unfortunately, non-pure (ϵ, δ)-private mechanisms (such as Gaussian mechanism) do not satisfy such a condition in general. In fact, due to possible translations of the set E by the $f(x)$'s, the control on all the probabilities $\mathbb{P}(\mathcal{A}(x) \notin E)$ requires more flexibility, that is a dependence $E = E_x$ on x (see Definition 8).

2.3 Sensitivity of a query

The notion of sensitivity of a query is crucial to design differentially private algorithms that are useful in practice. In our paper, for simplicity, we will only work on queries $f : \mathcal{D}^n \to R$ with real values. One can easily generalize the following definitions and our results to any range space R as soon as it is equipped with a distance.

Definition 6 Let $f : \mathcal{D}^n \to R$ be a query. The global sensitivity $GS(f)$ of f (denoted GS if there is no confusion) is defined by

$$GS(f) = \sup_{x \sim x'} |f(x) - f(x')|.$$

In some specific situations, the following notion of local sensitivity yields more accurate results. We will explain later in the paper how local sensitivity can be used to design instance-based noise (ϵ, δ)-differentially private mechanisms.

Definition 7 For all $x \subset \mathcal{D}^n$, the local sensitivity of a query $f : \mathcal{D}^n \to R$ at x, denoted by $LS(f)(x)$ (or simply $LS(x)$), is defined by

$$LS(f)(x) = \sup_{x \sim x'} |f(x) - f(x')|.$$

Note that local sensitivity depends on the participation of an individual in the database, hence it is not a private notion on its own. Nevertheless, one can use it in order to design differentially private mechanisms, which are more accurate than their analogs calibrated to global sensitivity.

2.4 Notations for Laplace and Gaussian mechanisms

In our results, we will calibrate the privacy by a parameter λ defining the considered distribution, that is distribution $g(y)$ proportional to $e^{-\lambda y}$ for Laplace random variable, and proportional to $e^{-\lambda y^2}$ for Gaussian random variable. Laplace (resp. Gaussian) mechanism is defined by

$$\mathcal{A}(x) = f(x) + Z_x$$

where Z_x is a Laplace (resp. Gaussian) random variable of parameter λ_x, possibly depending on $x \subset \mathcal{D}^n$.

3 (ϵ, δ)-differential privacy in distribution

In this section, we introduce (ϵ, δ)-differential privacy in distribution, a strenghtening of (ϵ, δ)-differential privacy. We then show on the example of Gaussian mechanism how the statement of (ϵ, δ)-differential privacy in distribution permits clear and short proofs of (ϵ, δ)-differential for any mechanism whose noise has a continuous distribution.

3.1 Main definition

We denote by A^c the complementary set of a subset $A \subset R$.

Definition 8 Let $f : \mathcal{D}^n \to R$ be a query function. Let $(Z_x)_{x \subset \mathcal{D}^n}$ be a family of random variables. Let $\mathcal{A}$ be a randomized mechanism, given by $\mathcal{A}(x) = f(x) + Z_x$ for all $x \subset \mathcal{D}^n$. Denote by g_x the distribution of $\mathcal{A}(x)$. Then we say that the mechanism $\mathcal{A}$ is (ϵ, δ)-*differentially private in distribution* if there exist subsets $E_x \subset R$ such that:

1. $\frac{g_x(y)}{g_{x'}(y)} \leq e^\epsilon$ for all $x' \sim x$, and all $y \in E_x$;
2. $\mathbb{P}(\mathcal{A}(x) \notin E_x) = \mathbb{P}(Z_x \in (E_x - f(x))^c) \leq \delta$ for all $x \subset D$.

Remark 9 • When $(Z_x)_{x \subset D}$ is a family of independent variables, identically distributed with respect to the Laplace distribution, then the mechanism $\mathcal{A}$ from the previous definition is the so-called Laplace mechanism.
• As said before, the usual notion of ϵ-differential privacy is equivalent to ϵ-differential privacy in distribution, hence the notion above is a generalization of ϵ-differential privacy. We will see in the next proposition that (ϵ, δ)-differential privacy in distribution implies (ϵ, δ)-differential privacy. We have no counter-example to show that they are not equivalent in general. We let this question open since our interest in this paper is only to simplify proofs of (ϵ, δ)-differential privacy.

In practice for continuous distributions, conditions in Definition 8 are much easier to deal with than the condition of (ϵ, δ)-differential privacy. The following proposition shows that it is indeed sufficient to prove (ϵ, δ)-differential privacy. Its proof is given in appendix 6.1.

Proposition 10 *Let $\mathcal{A}$ be a randomized mechanism. If $\mathcal{A}$ is (ϵ, δ)-differentially private in distribution, then $\mathcal{A}$ is also (ϵ, δ)-differentially private.*

3.2 Gaussian mechanism is (ϵ, δ)-differentially private in distribution

We first show that the Gaussian mechanism satisfies (ϵ, δ)-differential privacy in distribution for a good choice of its parameters. The proof follows the same idea as the existing proofs for traditional (ϵ, δ)-differential privacy. Our contribution is then to prove the (a priori) stronger notion of (ϵ, δ)-privacy in distribution for the Gaussian mechanism, while giving a proof as general as possible so that the same framework can be used in other similar situations, for other continuous distributions. The Definition 8 has been introduced as shown above for this particular reason since it gives the expected general framework. More precisely, one can see that we need to :

1. define the subsets E_x, in order to have ϵ-differential privacy for the restriction of distributions to the subsets E_x;

2. use tail bounds on the distribution to control what is out of $E_x - f(x)$ by the parameter δ.

Regarding the Gaussian mechanism, we obtain the following proposition.

Proposition 11 *Let $f : \mathcal{D}^n \to \mathbb{R}$ be a query, and GS its (global) sensitivity. Let $\mathcal{A}$ be the Gaussian mechanism on f, with parameter λ (see section 2.4). Then $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy in distribution if the following condition holds:*

$$\sqrt{\lambda} = \frac{\sqrt{C^2 + \epsilon} - C}{GS}$$

where $C = C(\delta) = h^{-1}(2\sqrt{\pi}\delta)$ and $h(y) = \frac{e^{-y^2}}{y}$.

Sketch of proof 12 Full details are given in the appendices. For $x \subset \mathcal{D}^n$, we define $E_x = \{\ y \in R \mid \lambda \times (GS^2 + GS \times 2 \times |f(x) - y|) \leq \epsilon\ \}$. Next we show that if the parameter λ is chosen as in the statement, then we have $\mathbb{P}(Z \notin E_x - f(x)) \leq \delta$, using the tail bound of the Gaussian random variable of parameter λ. ■

Remark 13 • One can easily derive a simple expression of λ in terms of the parameters ϵ, δ (see appendix for details). Indeed, for sufficiently small values of δ, we have the estimate

$$\sqrt{\lambda} \sim \frac{1}{GS} \times \frac{\epsilon}{2 \times \sqrt{\ln(\frac{1}{2\times\sqrt{\pi}\times\delta})}}.$$

• In cases where slightly more accuracy is needed, one can find better (i.e. smaller) values for λ: with notations from the proof, the value t' can be improved by approximating the zero of the function $\frac{e^{-t'^2}}{t'} - \sqrt{\pi} \times 2 \times \delta$.

4 Smooth sensitivity and instance-based noise

We now study the case of instance-based noise and smooth sensitivity with our new notion, and give several new results.

4.1 Instance-based noise and (ϵ, δ)-differential privacy in distribution

The first purpose of this section is to show that the instance-based noise technique introduced in [13] can be generalized, and then satisfies privacy in the sense of (ϵ, δ)-differential privacy in distribution.
In fact, the idea from [13] is to consider a mechanism of the form

$$\mathcal{A}(x) = f(x) + \frac{1}{\lambda_x} Z,$$

where Z is a Laplace random variable with parameter 1. The instance-based coefficient λ_x then defines the noise magnitude. With such a mechanism, one can hope to reduce

the error, and then manage larger values of λ_x for subsets $x \subset \mathcal{D}^n$ which are less sensitive.
The following result gives the right condition for such mechanism to be (ϵ, δ)-differentially private, using our new notion. On the one hand, this theorem is our second illustration that the framework of (ϵ, δ)-differential privacy in distribution allows for simple proofs of privacy. On the other hand, it is a generalization of the Smooth Sensitivity technique introduced in [13] (see section 4.2), which gives simple conditions to design instance-based noise differentially private algorithms. We use it in the sequel to give new differentially private algorithms. Before giving our result, we need to introduce the notation

$$\Delta_{x,x'} = \Delta_{x,x'}(f) = |f(x) - f(x')|.$$

Theorem 14 *Let $\mathcal{A}$ be the mechanism defined above, such that the noise magnitude satisfies the 2 following conditions for all $x \subset \mathcal{D}^n$:*

1. $\lambda_x \leq \alpha_x \times \frac{\epsilon}{\Delta_{x,x'}}$ *for all* $x' \sim x \in \mathcal{D}^n$;
2. $|1 - \frac{\lambda_x}{\lambda_{x'}}| \leq (1 - \alpha_x) \times \frac{\epsilon}{\ln(1/\delta)}$ *for all* $x' \sim x \in \mathcal{D}^n$,

and for some values $0 \leq \alpha_x \leq 1$. Then $\mathcal{A}$ is (ϵ, δ)-differentially private in distribution. In particular, $\mathcal{A}$ is (ϵ, δ)-differentially private.

Proof We need to find $E_{x'} \subset R$ such that $\mathbb{P}(\mathcal{A}(x') \in E_{x'}^c) \leq \delta$ and:

$$\lambda_x |f(x) - y| - \lambda_{x'} |f(x') - y| \leq \epsilon \text{ for all } y \in E_{x'}.$$

Moreover, a triangular inequality gives

$$\lambda_x |f(x) - y| - \lambda_{x'} |f(x') - y| \leq \lambda_x \Delta_{x,x'} + |\lambda_{x'} - \lambda_x| \times |f(x') - y|.$$

Hence from the assumptions, it is sufficient to find $E_{x'}$ such that:

$$|\lambda_{x'} - \lambda_x| \times |f(x') - y| \leq (1 - \alpha_x) \times \epsilon \text{ for all } y \in E_{x'}.$$

Now set $E_{x'} = \{ y \mid |\lambda_{x'} - \lambda_x| \times |f(x') - y| \leq (1 - \alpha_x) \times \epsilon \}$. We need to show that $\mathbb{P}(\mathcal{A}(x') \notin E_{x'}) \leq \delta$. This is a straightforward consequence of the tail bound estimation for Laplace distribution:

$$\begin{aligned} \mathbb{P}(\mathcal{A}(x') \notin E_{x'}) &= \mathbb{P}(Z > \frac{(1 - \alpha_x) \times \epsilon \times \lambda_{x'}}{|\lambda_{x'} - \lambda_x|}) \\ &= e^{-((1-\alpha_x)\times\epsilon)\times\frac{1}{|1-\frac{\lambda_x}{\lambda_{x'}}|}}. \end{aligned}$$

■

Remark 15 • One can easily derive analog results for noises with respect to other distributions, such as instance-based Gaussian noise.
• Condition 1 in Theorem 14 is equivalent to the condition

$$\lambda_x \leq \alpha_x \times \frac{\epsilon}{LS(x)},$$

illustrating that the amplitude λ_x is calibrated to local sensitivity (see Section 2.3).
• Condition 2 requires that amplitudes of 2 neighbor instances λ_x and $\lambda_{x'}$ should be close one from each other when $x \sim x'$, where the distance between them is measured by the privacy parameters ϵ and δ.

4.2 Comparison with Smooth Sensitivity calibrated noise

In [13], the authors introduced the notion of Smooth Sensitivity, given in the below definition. Such notion is between the local and the global sensitivity and has been designed to achieve private algorithms with better accuracy than those calibrated with the global sensitivity.

Definition 16 Let $\beta > 0$, and let $f : \mathcal{D}^n \to R$ be a query. A function $S : \mathcal{D}^n \to \mathbb{R}^+$ is a *β-smooth upper bound* on the local sensitivity $LS(f)$ if the following conditions hold for all $x \subset \mathcal{D}^n$:
(i) $S(x) \geq LS(f)(x)$;
(ii) $S(x) \leq e^\beta \times S(x')$ for all $x' \sim x$.

Global sensitivity $GS(f)$ is a (constant) 0-upper bound on $LS(f)$. The β-smooth sensitivity is another example of β-smooth upper bound.

Definition 17 Let $\beta > 0$, and let $f : \mathcal{D}^n \to R$ be a query. The *β-smooth sensitivity* $S^*_{f,\beta}$ of f is defined by

$$S^*_{f,\beta}(x) = \max_{y \subset \mathcal{D}^n} \ (\ LS(f)(y) \times e^{-\beta d(x,y)} \ )$$

where $d(x, y)$ is the number of individuals on which the databases x and y differ.

The authors of [13] show that β-smooth sensitivity of f is the optimal β-smooth upper bound on $LS(f)$. Lemma 2.5 in [13] states that the instance-based noise given in Section 4.1 is (ϵ, δ)-differentially private for $\lambda_x = \frac{\epsilon}{2 \times S(x)}$, where S is a β-smooth upper bound on local sensitivity with $\beta = \frac{\epsilon}{2 \times \ln(1/\delta)}$.
For a suitable choice of α_x, λ_x, the Lemma 2.5 given in [13] is a consequence of Theorem 14. Indeed, on can take $\alpha_x = \frac{1}{2}$ and $\lambda_x = \frac{\epsilon}{2 \times S(x)}$. For $\beta > 0$, we have $1 - e^\beta \leq \beta$. Since S is a β-smooth upper bound on local sensitivity, we have for all $x' \sim x$,

$$\begin{aligned} |1 - \frac{S(x)}{S(x')}| &\leq |1 - e^\beta| \\ &\leq \beta \\ &\leq \frac{\epsilon}{2} \times \frac{1}{\ln(1/\delta)}. \end{aligned}$$

Hence condition 2 in Theorem 14 is satisfied. Moreover, Condition 1 is straightforward since S is an upper bound on local sensitivity LS.

Remark 18 It was shown in [9] that computing Smooth Sensitivity of some graph statistics is NP-hard. In the next sections, we introduce an instance-based noise algorithm that avoids the use of Smooth Sensitivity, and allows efficient computations as long as local sensitivities can be efficiently computed or approximated.

5 Reducing error with instance-based noise

Since the noise error of the mechanism instance-based noise is precisely $\frac{1}{\lambda_x}$ (see above), our goal is to design an algorithm that chooses values of λ_x as large as possible, while satisfying both conditions in Theorem 14 (and not based on the Smooth Sensitivity). In this section, we propose such an algorithm.

5.1 Our instance-base noise algorithm

Let $f : \mathcal{D}^n \to R$ be a query. We consider values $(LS(x))_{x \subset \mathcal{D}^n}$, corresponding to the local sensitivities of f that can be ordered increasingly as follows: $(LS_1, ...LS_r)$, $LS_i \leq LS_{i+1}$. Notice that with such notations, we have $LS_r = GS$. We will denote

$$D_k = \{\ x \mid LS(x) = LS_k\ \},$$

that we call the k-level of sensitivities. Moreover, we will make use of the notation $k \sim l$ when there exist $x \sim x'$ such that $x \in D_k$ and $x' \in D_l$. We also write $x < x'$ if $x \in D_k$, $x' \in D_l$ for some $k < l$.

Ideally, we would like to design our instance-based noise algorithm with noise amplitude at level k equal to $\lambda_k = \frac{\epsilon}{LS_k}$. Unfortunately, it is possible that condition 2 in Theorem 14 is not satisfied, reflecting the fact that local sensitivity is not private in general. Hence we will find $0 < \alpha_k < 1$'s as large as possible such that both $\lambda_k = \alpha_k \times \frac{\epsilon}{\Delta_k}$ and condition 2 hold. For that, we first do an analysis on the situation where only two local sensitivities are at stake, and then we use careful observations on the neighboring relationships to design Algorithm 1.

As explained previously, we will need to give a solution to the following optimization problem, which is the situation of a query whose local sensitivities take only two values. Let $0 < LS_1 < LS_2$, and $0 < t \leq 1$. Let $\lambda_1 = \alpha_1 \times \frac{\epsilon}{LS_1}$, $\lambda_2 = \alpha_2 \times \frac{\epsilon}{LS_2}$, for some $0 \leq \alpha_1, \alpha_2 \leq 1$. Moreover, assume that we want the following constraints to hold:

$$|1 - \frac{\lambda_1}{\lambda_2}| \leq (1 - \alpha_1) \times t \quad (1)$$
$$|1 - \frac{\lambda_2}{\lambda_1}| \leq (1 - \alpha_2) \times t \quad (2)\ .$$

Claim 19 *The following condition is a sufficient condition to satisfy (1) and (2) in the situation where $LS_1 \leq LS_2$, and the values α_1, α_2 are chosen less than $\frac{1}{2}$:*

$$\lambda_2 \leq \lambda_1 \leq \min(\ \lambda_2 \times (1 + \frac{1}{2}t)\ ,\ \frac{\epsilon}{2 \times LS_1}\ ).\quad (*)$$

Proof of Claim 19 is given in the appendices. Now the idea of our algorithm goes as follows. We build the sequence $(\lambda_k)_k$ by descending induction, starting from $\lambda_r = \frac{\epsilon}{2LS_r}$. When λ_{k+1} $(\leq \frac{\epsilon}{2LS_{k+1}})$ is given, we try to find λ_k $(\leq \frac{\epsilon}{2LS_k})$ greater than λ_{k+1}, and such that the privacy conditions $(*)$ hold with all previously constructed λ_{k+m} (these conditions were kept in memory in previous steps). During step k also, we compute the privacy conditions $(*)$ between λ_k and the lower neighbors λ_{k-m}, to be kept in memory until we reach the lowest of their levels $\tau_k = k - m_0$. Later in the algorithm, at step

$k - m_0 - 1$, we are free to delete the privacy conditions associated to λ_k, in order to maximize as much as possible further values of $(\lambda_j)_{j \leq k-m_0-1}$. In the proof, we will see that in order to choose $\lambda_k \geq \lambda_{k+1}$ at step k, we need that LS_k and LS_{k+1} are not too close one from each other. More precisely, we need the condition $(1 + \frac{1}{2} \times t) \geq \frac{LS_k}{LS_{k+1}}$, for $t = \frac{\epsilon}{\ln(1/\delta)}$.
We now use the previous discussion to design the following instance-based noise Algorithm 1. If the algorithm for the computation of the local sensitivities is efficient, then our Algorithm 1 is efficient as well.

Algorithm 1: Instance-Based Noise Algorithm
Input: data set $y \subset \mathcal{D}^n$, query f, privacy parameters ϵ, δ
Output: private value for $f(y)$
. set $t = \frac{\epsilon}{\ln(1/\delta)}$
. compute LS_r and set $\lambda_r = \frac{\epsilon}{2LS_r}$
. **if** $y \in D_r$, **return** $f(y) + \mathrm{Lap}(\lambda_r)$
. **end if**
. **for** $l \sim r, l < r$, compute LS_l
. **end for**
. compute $\tau_r = \min_{l<r, l\sim r} l$
. **for** k from r-1 to 1, **do**
. **for** $l \sim k, l < k$, compute LS_l
. **end for**
. compute $\tau_k = \min_{l<k, l\sim k} l$
. **if** $\frac{LS_{k+1}}{LS_k} < (1 + \frac{1}{2} \times t)$, set $\lambda_k = \lambda_{k+1}$
. **else** compute $l_k = \max_{\tau_l \leq k} l$
. set $\lambda_k = \min(\frac{\epsilon}{2LS_k}, \lambda_{l_k} \times (1 + \frac{1}{2} \times t))$
. **end if**
. **if** $y \in D_k$, **return** $f(y) + \mathrm{Lap}(\lambda_k)$
. **end if**
. **end for**

Proposition 20 *Algorithm 1 is (ϵ, δ)-differentially private in distribution. It follows that Algorithm 1 is (ϵ, δ)-differentially private.*

First, we notice that Algorithm 1 is well-defined since the induction process is valid. As already assumed, $\mathcal{D}^n$ is connected with respect to our neighboring relation. In particular, there exists $l > k$ (when $k \neq r$) such that $\tau_l \leq k$. It follows that $l_k > k$, and then λ_k is constructed from the sequence $(\lambda_l)_{l>k}$. A complete proof of Proposition 20 is given in appendix 6.2. Now we emphasize on some nice features of Algorithm 1.

Remark 21 • Algorithm 1 does not require to find a β-smooth upper bound, nor to compute the Smooth Sensitivity $S^*_{f,\beta}$ (which requires the knowledge of all local sensitivities for the computation of one single value $S^*_{f,\beta}(x)$). Given some $x \subset \mathcal{D}^n$, $x \in D_k$, we only need to know the list $(LS_i)_{i \geq k}$ and the subjacent relationships for $y, y' \subset \cup_{l \geq k} D_l$.
• Compared to standard Laplace mechanism, Algorithm 1 is particularly attractive when local sensitivities are far apart from each other, and when repeated queries on

different data sets in $\mathcal{D}^n$ are asked. In such a situation, the trusted server can keep in memory the pairs (λ_k, LS_k) already computed in the past, in order to avoid repeating computations in the future. Even with reduced memory storage, one can keep in memory a *sparse* well-chosen subset of these pairs.
• Sometimes it is easier to compute upper bounds $\tilde{LS}_k$ on local sensitivities LS_k. Algorithm 1 is still (ϵ, δ)-differentially private if we replace LS_k by its approximation $\tilde{LS}_k$.
• We believe that refined algorithms can be designed relying on Theorem 14, in particular when a trusted server knows that some query $f(x)$ is more likely to be asked, and $LS(x)$ is a small value from the list $(LS_i)_i$.

5.2 Algorithm 1 working on examples, and discussion about utility

Here we give some insights about when Algorithm 1 should be preferred to other variants: roughly speaking, Algorithm 1 gives significantly better accuracy when sufficiently many LS_k's are far apart from each other.

Example 22 To illustrate a typical situation where Algorithm 1 gives significantly better bounds, let us assume that we have $LS_1 < ... < LS_r$, and that for all i, a dataset $x \in D_i$ can have neighbors only in D_{i+1}, D_i or D_{i-1}. Assume also that local sensitivities are not too close one from each other, that is $\frac{LS_{i+1}}{LS_i} \geq 1 + \frac{1}{2} \times t$ for $t = \frac{\epsilon}{\ln(1/\delta)}$.
Here we decide to compare mean error of Laplace mechanism to that of mechanism from Algorithm 1 over r datasets x_i, each one from a distinct D_i. For standard Laplace noise, this error is given by the following formula:

$$\text{err}_{Laplace} = r \times \frac{LS_r}{\epsilon}.$$

Now assume that $t = f(r) \to 0$ as r becomes larger. Then the mean error over r datasets chosen as above can be approximated (as r goes to infinity) by:

$$\text{err}_{Algo1} \sim_{r>>1} \frac{2 \times LS_r}{\epsilon} \times (1 + \frac{2}{t}) \times (1 - e^{-(1/2)\times rt}).$$

From these computations (see the appendix for details), it is clear that if one chooses parameters ϵ, δ such that $\lim_{r\to\infty} rt = \infty$, then the error err_{Algo1} is equivalent to $\frac{2\times LS_r}{\epsilon} \times (1 + \frac{2}{t})$, which is negligible compared to $\text{err}_{Laplace}$.
For instance, take $t = \frac{1}{\ln(r)}$. Then we have $\text{err}_{Algo1} \sim 4 \times \ln(r) \times \frac{LS_r}{\epsilon}$. Such a value of $t = \frac{\epsilon}{\ln(1/\delta)}$ is reasonable from a privacy point of view: indeed, it fits with the values $\epsilon = 1$, $\delta = \epsilon^2/r$, which are admissible values of the privacy parameters (see [5], [8]).
But when $rt << 1$, it is easy to see that the error err_{Algo1} is equivalent to $\text{err}_{Laplace}$ up to some constant independent of r. Hence, there is no gain in utility with Algorithm 1 if parameters ϵ, δ are chosen too small (which is anyway not reasonable in practice).

Example 23 In [9] (in their Theorem A.1), it is shown that computing Smooth Sensitivity of some statistics on graphs is a NP hard problem. More precisely, let $\mathcal{G}_N$ be the set of (undirected) graphs with N vertices, and consider that two graphs are neighbors if and only if they differ from each other by exactly one edge (hence $\mathcal{D}^n = \mathcal{G}_N$, n being the number of possible edges for graphs in $\mathcal{G}_N$). Let $f_{2\Delta}$ the number of 2-triangles (a

2-triangle is given by two triangles sharing a common edge). Then it is NP hard to compute $S^*_{f,\beta}(f_{2\Delta})$.
Moreover, authors of [9] give an alternative method to Smooth Sensitivity technique in order to design an instance-based noise (ϵ, δ)-differentially private algorithm. They use a second order local sensitivity, that is local sensitivity of local sensitivity itself. Unfortunately, their result gives reasonable accuracy only if one wants to release privately $f_{2\Delta}(G)$ for some sufficiently connected graph G. The precise assumption they made is the existence of a pair of vertices in G which have a number of common neighbors significantly greater than $\frac{\ln(1/\delta)}{\epsilon}$. Our algorithm does not make any assumption on the graphs we want to release and can be used even when all graphs do not have many connections. We now explain how it can be performed.
For a graph $H \in \mathcal{G}_N$, and an edge $e \notin H$, denote by $H + e$ the graph obtained from H by adding the edge e. For an edge $e \in G$ and $H \in \mathcal{G}_N$, denote by a_e^H the number of triangles in H involving edge e. Set $a_H = \max_{e \in H+e', e' \notin H} a_e^{H+e'}$, and $\overline{LS}(H) = \frac{5}{2} \times a_H$. Using [9], it is easy to show that $\overline{LS}$ is an upper bound approximation on LS (see details in the appendices). Hence Algorithm 1 can be used with $\overline{LS}$ in place of LS.
Moreover, one can simplify the design of Algorithm 1 by making a further approximation on $\overline{LS}$. Indeed, define the k-layer L_k to be the subset of $\mathcal{G}_N$ of graphs with k egdes. In particular, two graphs are neighbors if and only if they lie in successive layers L_k, L_{k+1}. Set $\tilde{LS}_k = \max_{H \in L_k} \overline{LS}(H)$. Then $(\tilde{LS}_k)_k$ is ordered increasingly, and one can use Algorithm 1 with $\tilde{LS}_k$ in place of LS_k, and L_k in place of D_k. In particular, the level τ_k is always equal to $k-1$ in such a situation.

Remark 24 • Since we are looking for values of t which are not too large, the condition $\frac{\overline{LS}_{k+1}}{\overline{LS}_k} \geq 1 + \frac{1}{2}t$ should hold in most cases, and especially when the values a_H are not too large (this is in contrast with the assumption of [7] where it is required a_H to be much larger than $1/t$).
• For the first $f(x)$ ($x \in D_l$) query asked, computation of the sequence $(\overline{LS}_k)_{k \geq l}$ is required, and this can be computed in time $O(d_{max} m^2)$ where m is the number of possible edges in the overall graph G considered, and d_{max} a bound on the degree of vertices. For next computations, the lists $(\overline{LS}_k)_{k \geq l}$ already computed can be used, and many less computations are required. The computation time is more likely to be $O(d_{max} m)$ after a few queries.
• This latter example illustrates that one can obtain a more simple version of Algorithm 1 if it is possible to find an ordering on data subsets (the layers L_k) which is coherent with the ordering of local sensitivities (the approximations $\tilde{LS}_k$).

5.3 A few remarks about privacy in Instance-Based Noise Algorithm

Algorithm 1 always gives better bounds than standard Laplace noise. Nevertheless, privacy guarantees offered by pure ϵ-differential privacy are stronger than that of (ϵ, δ)-differential privacy.
Our methods in the design of Algorithm 1 show an interesting feature of databases privacy. Even if we want to avoid calibrating noise to global sensitivity, we are in some sense forced to do it: we can make significantly less noise only for databases far away (with respect to the neighboring relation we are looking at) from the sensitive databases.

References

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

[2] [DKMMN] C. Dwork, K. Kenthapadi, F. Mc Sherry, I. Mironov and M. Naor. Our data, ourselves: Privacy via distributed noise generation. *Advances in Cryptology-EUROCRYPT*, p 486-503, 2006.

[3] [DMNS] 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.

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

[5] [GKS] S. R. Ganta, S. P. Kasiviswanathan and A. Smith. Composition attacks and auxiliary information in data privacy. *Proceedings of the 14th ACM SIGKDD international on Knowledge and data mining*, p 265-273, 2008.

[6] [KKMM] K. Kenthapadi, A. Korolova, I. Mironov and N. Mishra. Privacy via the Johnson-Lindenstrauss transform. *arXiv preprint*:1204.2606. 2012.

[7] [KNRS] 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.

[8] [KS] S. P. Kasivisiwanathan and A. Smith. On the 'Semantics' of Differential Privacy: A Bayesian Formulation. *Journal of Privacy and Confidentiality*, 6, Number 1, 1-16, 2014.

[9] [KRSY] 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.

[10] [M] F. D. Mc Sherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, p 19-30, 2009.

[11] [MW] 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.

[12] [NTZ] A. Nikolov, K. Talwar and L. Zhang. The geometry of differential privacy: the sparse and approximate cases. *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, p 351-360, 2013.

[13] [NRS] 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.

[14] [WW] 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.

6 Appendix

Now we show the missing proofs from this paper.

6.1 Proofs for differential privacy in distribution

We prove the equivalence between ϵ-differential privacy and ϵ-differential privacy in distribution.

Proof of Proposition 5 We denote by 1_S the indicator function of a subset $S \subset R$. The result is a consequence of the following formulae

$$\mathbb{P}(\mathcal{A}(x) \in S) = \int_S g_x(y)dy \text{ for all } S \subset R.$$

From ϵ-differential privacy in distribution, one obtains ϵ-differential privacy by integrating the relations $g_x(y) \leq e^{\epsilon} g_{x'}(y)$.
Now, if $\mathcal{A}$ is ϵ-differentially private, then we have

$$1_S(y)(g_x(y) - e^{\epsilon} g_{x'}(y) \leq 0 \text{ for all } S.$$

Then ϵ-differential privacy in distribution follows by considering $S = \{\ y \mid g_x(y) - e^{\epsilon} g_{x'}(y) \geq 0\ \}$. ■

We show that (ϵ, δ)-differential privacy in distribution implies (ϵ, δ)-differential privacy.

Proof of Proposition 10 This follows from the formula $\mathbb{P}(\mathcal{A}(x) \in S) = \int_S g_x(y)dy$, and the following inequalities:

$$\begin{aligned}
\mathbb{P}(\mathcal{A}(x) \in S) &= \mathbb{P}(\mathcal{A}(x) \in S \cap E_x) + \mathbb{P}(\mathcal{A}(x) \in S \cap E_x^c) \\
&\leq \mathbb{P}(\mathcal{A}(x) \in S \cap E_x) + \delta \\
&\leq e^{\epsilon}\mathbb{P}(\mathcal{A}(x') \in S \cap E_x) + \delta \\
&\leq e^{\epsilon}\mathbb{P}(\mathcal{A}(x') \in S) + \delta,
\end{aligned}$$

using the equality

$$\mathbb{P}(\mathcal{A}(x) \in S \cap E_x) = \int_{S \cap E_x} g_x(y)\ dy = \int_{S \cap E_x} \frac{g_x(y)}{g_{x'}(y)} \times g_{x'}(y)\ dy.$$

■

Here is the proof that Gaussian mechanism is a (ϵ, δ)-differentially private mechanism, for a suitable choice of parameters ϵ, δ.

Proof of Proposition 11 Notice that condition 1 in Definition 8 writes as follows for the Gaussian mechanism:

$$\lambda \times |(f(x) - y)^2 - (f(x') - y)^2| \leq \epsilon \text{ for all } x \subset \mathcal{D}^n, x \sim x', y \in E_x.$$

Denote by Z the Gaussian random variable of parameter λ. A sufficient condition to prove the proposition is to find sets E_x such that $\mathbb{P}(Z \in (E_x - f(x))^c) \leq \delta$ and:

$$\lambda \times (GS^2 + GS \times 2|f(x) - y|) \leq \epsilon \text{ for all } x \subset \mathcal{D}^n.$$

Hence we naturally define, for $x \subset \mathcal{D}^n$,

$$E_x = \{\ y \in \mathbb{R} \mid \lambda \times (GS^2 + GS \times 2|f(x) - y|) \leq \epsilon\ \}.$$

Hence we have

$$\begin{aligned}
\mathbb{P}(Z \in (E_x - f(x))^c) &= \mathbb{P}(\ |Z| \geq \frac{\epsilon}{2\lambda \times GS} - \frac{GS}{2}\ ) \\
&= 2 \times \mathbb{P}(Z \geq \frac{\epsilon}{2\lambda \times GS} - \frac{GS}{2}) \\
&\leq \frac{\mu}{\lambda \times t} e^{-\lambda \times t^2} \\
&= \frac{1}{2\sqrt{\pi} \times t'} e^{-t'^2}
\end{aligned}$$

where $t' = \sqrt{\lambda} \times t = \sqrt{\lambda} \times (\frac{\epsilon}{2\lambda \times GS} - \frac{GS}{2})$. Notice that in order to have $\mathbb{P}(\ |Z| \geq \frac{\epsilon}{2\lambda \times GS} - \frac{GS}{2}\ )$ small (at least for $\delta \leq 1/2$), it is necessary to have $\frac{\epsilon}{2\lambda \times GS} \geq \frac{GS}{2}$.
Then we have

$$\begin{aligned}
&\frac{1}{2\sqrt{\pi} \times t'} e^{-t'^2} \leq \delta \\
&\Leftrightarrow \frac{e^{-t'^2}}{t'} \leq \sqrt{\pi} \times 2 \times \delta \\
&\Leftrightarrow t' \geq h^{-1}(2\sqrt{\pi} \times \delta)
\end{aligned}$$

where h is the decreasing function defined by $h(y) = \frac{e^{-y^2}}{y}$. We denote by $C = C(\delta) = h^{-1}(2\sqrt{\pi} \times \delta)$.
Now we have

$$\begin{aligned}
t' \geq C &\Leftrightarrow \frac{\epsilon}{2GS} - \frac{\lambda \times GS}{2} - C \times \sqrt{\lambda} \geq 0 \\
&\Leftrightarrow \frac{\epsilon}{2GS} - C \times y - \frac{GS}{2} \times y^2 \geq 0
\end{aligned}$$

The polynomial above has two roots, namely

$$y_+ = \frac{C - \sqrt{C^2 + \epsilon}}{-GS} \text{ and } y_- = \frac{C + \sqrt{C^2 + \epsilon}}{-GS}.$$

Hence y satisfies the inequality if and only if $y_- \leq y \leq y_+$. Then we obtain that $t' \geq C$ if and only if $0 \leq \sqrt{\lambda} \leq y_+$.
Since we want to maximize $\sqrt{\lambda}$ in order to reduce noise, the optimal bound (with respect to the tail bound of the distribution we used) is

$$\sqrt{\lambda} = \frac{\sqrt{C^2 + \epsilon} - C}{GS}.$$

■

Notice that if δ is small enough in the above proof, then we have $t' \geq 1$. Indeed, for $\delta \leq 0.1$ (δ should be even smaller for applications), one can check that values t' such that $h(t') \leq 2 \times \sqrt{\pi} \times 0.1$ satisfy $t' \geq 1$. Then the following condition is sufficient to get (ϵ, δ)-differential privacy:

$$e^{-t'^2} \leq 2 \times \sqrt{\pi} \times \delta.$$

As in the previous proof, we can take $C = \sqrt{\ln(\frac{1}{2 \times \sqrt{\pi} \times \delta})}$. Hence, for small values of δ, ϵ, we have the estimate

$$\sqrt{\lambda} \sim \frac{1}{GS} \times \frac{\epsilon}{2 \times \sqrt{\ln(\frac{1}{2 \times \sqrt{\pi} \times \delta})}}.$$

6.2 Proofs relative to Algorithm 1

Our design of Algorithm 1 relies on Claim 19, which is the situation when the query has only two distinct local sensitivities $LS_1 < LS_2$.

Proof of Claim 19 Our goal is the following: $\alpha_2 \neq 1$ being fixed, we want to find α_1 as large as possible in order to maximize λ_1 (in particular we want $\lambda_1 \geq \lambda_2$), and such that the above conditions (1) and (2) hold. We have $\frac{\lambda_1}{\lambda_2} = \frac{\alpha_1}{\alpha_2} \times \frac{LS_2}{LS_1} = \frac{\alpha_1}{\alpha_2} \times s$, where $s = \frac{LS_2}{LS_1} > 1$.

• Step 1: We look for some $\lambda \geq 1$ such that $\frac{\lambda_1}{\lambda_2} = \lambda$, and rewrites equations (1) and (2).

Write $\lambda = 1 + \gamma$, where $\gamma > 0$. We have:

$$\begin{aligned}(1) &\Leftrightarrow \lambda - 1 \leq (1 - \alpha_1) \times t \\ &\Leftrightarrow \gamma \leq (1 - \alpha_1) \times t,\end{aligned}$$

and

$$\begin{aligned}(2) &\Leftrightarrow 1 - \frac{1}{\lambda} \leq (1 - \alpha_2) \times t \\ &\Leftrightarrow \lambda - 1 \leq \lambda \times (1 - \alpha_2) \times t \\ &\Leftrightarrow \gamma \leq \lambda \times (1 - \alpha_2) \times t.\end{aligned}$$

In particular, notice that it is sufficient to have $\gamma \leq (1 - \alpha_2) \times t$ for (2) to hold.

• Step 2: Give the admissible value of α_1, depending on a common threashold α for α_1, α_2, that is $\alpha_1 \leq \alpha$ and $\alpha_2 \leq \alpha$. When the threashold α is determined (for us, it is determined by the value α_2), we have $\gamma \leq \alpha \times t \Rightarrow (1) + (2)$.

In the paper, we use exclusively the threashold $\alpha = 1/2$. Then $\gamma \leq (1/2) \times t \Rightarrow (1) + (2)$, which gives the following admissible values for α_1:

$$\alpha_1 \leq \min(\ \frac{\alpha_2}{s} \times (1 + (1/2) \times t)\ ,\ 1/2\ ).$$

■

Now we prove the privacy statement concerning Algorithm 1.

Proof of Proposition 20 We need to show that the mechanism constructed in Algorithm 1 satsifies condition 2 from Theorem 14, that is

$$|1 - \frac{\lambda_x}{\lambda_{x'}}| \leq (1 - \alpha_x) \times \frac{\epsilon}{\ln(1/\delta)} \text{ for all } x \in \mathcal{D}^n, x' \sim x$$

where $\lambda_x = \alpha_x \times \frac{\epsilon}{LS(x)}$. From the discussion above restricted to the situation with two sensitivities (with $t = \frac{\epsilon}{\ln(1/\delta)}$), we deduce that it is sufficient to find $0 < \alpha_x \leq \frac{1}{2}$ such that, for all $x' \sim x$, $x' < x$, the following inequalities hold:

$$\lambda_x \leq \lambda_{x'} \leq \min(\ \lambda_x \times (1 + \frac{1}{2}t)\ ,\ \frac{\epsilon}{2 \times LS(x')}\ ). \quad (*)$$

Inequalities on the left in $(*)$ is simply requiring that the sequence $(\lambda_k)_k$ is decreasing. We only need to prove the monoticity in the case $\frac{LS_{k+1}}{LS_k} \geq (1 + \frac{1}{2} \times t)$. For that, we consider the following two cases:

- Case 1 : $\lambda_k = \frac{\epsilon}{2LS_k}$. Obviously, we have $\frac{\epsilon}{2LS_k} \geq \frac{\epsilon}{2LS_{k+1}}$. Moreover, we have $\lambda_{l_{k+1}} \leq \lambda_{k+1}$ (by monoticity in the previous steps), and then $\lambda_{l_{k+1}} \leq \frac{\epsilon}{2LS_{k+1}}$. Hence, in order to have $\lambda_k \geq \lambda_{k+1}$, it is sufficient that the following holds:

  $$\frac{\epsilon}{2LS_k} \geq \frac{\epsilon}{2LS_{k+1}} \times (1 + \frac{1}{2} \times t).$$

  This is the case since we are in the situation where $\frac{LS_{k+1}}{LS_k} \geq (1 + \frac{1}{2} \times t)$.

- Case 2 : $\lambda_k = \lambda_{l_k} \times (1 + \frac{1}{2} \times t)$. This is clear that level l_k decreases as k itself decreases. Hence by monoticity in previous steps, we have $\lambda_{l_k} \geq \lambda_{l_{k+1}}$. It follows

  $$\lambda_k \geq \min(\ \lambda_{l_{k+1}} \times (1 + \frac{1}{2} \times t)\ ,\ \frac{\epsilon}{2LS_{k+1}}\ )$$

  as required.

To finish the proof, we need to show inequalities on the right in $(*)$. This is a straightforward consequence of the choice of λ_k in Algorithm 1. Indeed, λ_k is calibrated to $\lambda_{l_k} \times (1 + \frac{1}{2} \times t)$ which is, by definition of l_k, the lowest possible value of $\lambda_l \times (1 + \frac{1}{2} \times t)$ among possible upper neighbors $l \sim k$, $l > k$. ■

6.3 Proofs for Examples 22 and 23

We explain computations occuring in Example 22.

Details for Example 22 For standard Laplace mechanism, we have all amplitude parameters λ_{x_i} equal to $\lambda = \frac{\epsilon}{LS_r}$. Hence we have

$$\begin{aligned}\mathrm{err}_{Laplace} &= \sum_i \mathbb{E}(\mathrm{Lap}(\lambda_{x_i})) \\ &= \sum_i \frac{1}{\lambda_{x_i}} \\ &= r \times \frac{\epsilon}{LS_r}.\end{aligned}$$

On the other hand, we made the assumption that neighbors of a database lie in the neighboring levels of sensitivities, and that local sensitivities are not too close one from each other. It follows that $\lambda_{x_i} = \frac{\epsilon}{LS_r} \times (1 + \frac{1}{2}t)^{r-i}$, for $1 \leq i \leq r$ and $t = \frac{\epsilon}{\ln(1/\delta)}$. Then the error for instance-based mechanism from Algorithm 1 is as follows:

$$\begin{aligned}\mathrm{err}_{Algo1} &= \sum_{i=1}^{r} \frac{1}{\lambda_{x_i}} \\ &= \frac{2 \times LS_r}{\epsilon} \times (\sum_{i=0}^{r-1} (\frac{1}{1 + 1/2 \times t})^i) \\ &= \frac{2 \times LS_r}{\epsilon} \times \frac{1 - \frac{1}{(1+1/2\times t)^r}}{1 - \frac{1}{1+1/2\times t}} \\ &= \frac{2 \times LS_r}{\epsilon} \times (1 + \frac{2}{t}) \times (1 - \frac{1}{(1 + 1/2 \times t)^r}).\end{aligned}$$

Now assume that t is chosen such that $t = f(r) \to 0$ as $r \to \infty$. Then we have

$$\mathrm{err}_{Algo1} \sim \frac{2 \times LS_r}{\epsilon} \times (1 + \frac{2}{t}) \times (1 - e^{-1/2\times rt}).$$

In particular, we have $\mathrm{err}_{Algo1} << \mathrm{err}_{Laplace}$ when $rt >> 1$. Indeed, we have $1/t << r$, and $\mathrm{err}_{Algo1} \sim \frac{2\times LS_r}{\epsilon} \times (1 + \frac{2}{t}) << r \times \frac{LS_r}{\epsilon}$. ■

Now we give some explanations concerning Example 23.

Details for Example 23 First, we recall notations from [9]. Denote by $(x_{ij})_{i,j\in[N]}$ the adjacency matrix, and then $a_{ij} = \sum_{l\in[N]} x_{il}x_{lj}$ the number of triangles involving the edge (ij). Hence we have $a_{ij} = |N_{ij}|$ where $N_{ij} = \{\ l \in [N] \mid x_{il}x_{lj} = 1\ \}$. We write a_e, N_e for the edge $e = (ij)$. Notice that these notations depend on the graph $H \in \mathcal{G}_N$ considered. For shorthand, we write LS for $LS(f_{2\Delta})$. Moreover, we recall that $a_H = \max_{e\in H+e', e'\notin H} a_e^{H+e'}$ for all graph $H \in \mathcal{G}_N$.

Lemma 25 *For any graph $H \in \mathcal{G}_N$, we have the following upper bound estimate:*

$$LS(H) \leq \frac{5}{2} \times a_H^2.$$

Proof By Lemma 4.1 in [9], we have $LS(H) = \max_{e' \notin H} \max_{e \in H+e'} LS_e(H)$, where

$$LS_{ij}(H) = \binom{a_{ij}}{2} + \sum_{l \in N_{ij}} a_{il} + a_{lj} - 2x_{ij}.$$

The result follows since we have $\binom{a_{ij}}{2} \leq \frac{1}{2} \times a_H^2$ and $\sum_{l \in N_{ij}} a_{il} + a_{lj} - 2x_{ij} \leq 2 \times a_H^2$, for all $(ij) = e \in H + e'$, and all edge e'.

Hence formulae $\overline{LS}(H) = \frac{5}{2} \times a_H$ define an upper bound $\overline{LS}$ on local sensitivity LS. Moreover, when we have an inclusion of graphs $H \subset H'$, we have $\overline{LS}(H) \leq \overline{LS}(H')$ as well, since more edges (thus more triangles) are considered to compute $a_{H'}$. Recall that we defined the following approximation on $\overline{LS}$: $\tilde{LS}_k = \max_{H \in L_k} \overline{LS}(H)$, where $L_k \subset \mathcal{G}_N$ is the subset of graphs with k edges. In particular, it follows that $\tilde{LS}_k \leq \tilde{LS}_{k+1}$ since any graph in layer L_k is included in a graph in layer L_{k+1}. ■