Oblivious Mechanisms in Differential Privacy
Experiments, Conjectures, and Open Questions

Chien-Lun Chen, Ranjan Pal, Leana Golubchik
Department of Computer Science
University of Southern California
Los Angeles, USA
Email: {chienlun, rpal, leana}@usc.edu

***Abstract*—Differential privacy (DP) is a framework to quantify to what extent individual privacy in a statistical database is preserved while releasing useful aggregate information about the database. In this work, we aim an exploratory study to understand questions related to the optimality of noise generation mechanisms (NGMs) in differential privacy by taking into consideration the (i) query sensitivity, (ii) query side information, and (iii) the presence of longitudinal and collusion attacks. The results/observations from our study serve three important purposes: (i) provide us with conjectures on appropriate (in the sense of privacy-utility tradeoffs) oblivious NGM selection for scalar queries in both non-Bayesian as well as Bayesian user settings, (ii) provide supporting evidence and counterexamples to existing theory results on the optimality of NGMs when they are tested on a relaxed assumption set, and (ii) lead to a string of interesting open questions for the theory community in relation to the design and analysis of *provably optimal* oblivious differential privacy mechanisms.**

***Index Terms*—Privacy, Noise Generation Mechanisms, Utility**

1. Introduction

Organizations such as the Census Bureau, hospitals, and Internet companies have long maintained databases of personal information. The census bureau may, for instance, publish the result of a statistical query such as "How many individuals have incomes that exceed $100,000?" An implicit hope here is that the released aggregate information is sufficiently anonymous so as not to breach the privacy of any individual. Unfortunately, publication schemes initially thought to be "private" have succumbed to privacy attacks [1], highlighting the urgent need for mechanisms that are provably private.

Differential Privacy (DP) is a formal framework to quantify to what extent individual privacy in a statistical database is preserved while releasing useful aggregate information about the database. It provides strong privacy guarantees by requiring the indistinguishability of whether an individual is in the dataset or not based on the released information. The key idea of differential privacy is that the presence or absence of any individual data in the database should not affect the final released statistical information significantly, and thus it can give strong privacy guarantees against an adversary with arbitrary auxiliary information. Since its introduction in [2] by Dwork et. al., differential privacy has spawned a large body of research in differentially private data-releasing mechanism design and performance analysis in various settings, e.g., statistical query processing, machine learning, pricing, etc. Differential privacy is a privacy-preserving constraint imposed on the query output releasing mechanisms, and to make use of the released information, it is important to understand the fundamental tradeoff between utility(accuracy) and privacy.

Research Motivation - When answering a scalar (single-dimensional numeric) query, differential privacy is usually achieved by adding some noise to the result of the query on a given dataset via a noise generation mechanism (NGM) [see Section 2.5]. It is evident that adding random noise to a correct query output in order to preserve privacy will generate an error for the query generator (QG), and will result in its loss. The important question at hand here is, *for a given query, what is the best NGM(s) that for a given desired level of privacy, guarantees a low loss to the QG?* Quite a few researchers (see 'Related Work') have focussed on this question in theory, and either (i) proposed provably universally optimal NGMs for *specific* query types (e.g., count) under general QG loss functions, (ii) prove that for general query functions, there exists no universally optimal NGM that minimize QG loss, or (iii) prove the optimality of a given NGM for specific queries under restricted conditions related to the query sensitivity and side information. *However, for DP to be prevalently used in practice, we need to find an answer to the question of what NGM(s) provide good (if not optimal) privacy-utility tradeoffs (see Section 2.4) for general query types when QGs possess general loss functions, and in scenarios unrestricted by assumptions related to query sensitivity and side information.*

Goal - Our *primary* goal in this paper is to conduct an exploratory study to understand questions and challenges related to the design and analysis of optimal oblivious noise generation mechanisms (NGMs) in differential privacy (see Section 4.1).

Contributions - In regard to our goal, for the *general* utility-maximization framework in DP, we use experiments to understand the privacy-utility impact of adding various oblivious noise generation mechanisms (NGMs) to the output of single real-valued (scalar) query functions having arbitrary sensitivity. More specifically, for a given scalar query function of a particular output domain (continuous or discrete), we investigate the existence and design of high utility preserving oblivious NGMs for a given privacy regime (high or low) in a Bayesian setting. Our study takes into consideration (i) different privacy regimes (levels of privacy strength), (ii) continous and discrete query output domains, (iii) varied levels of query sensitivity, (iv) query side information, and (v) the presence of collusion and longitudinal attacks on a query (see Sections 4.1 and 5.3). Our experiments help provide supporting evidence and counterexamples to existing theory results on the optimality of NGMs when they are tested on a relaxed assumption set. The experimental results (see Section 5) also provide us with conjectures on appropriate (in the sense of privacy-utility tradeoffs) oblivious NGM selection for scalar queries with side information in Bayesian user settings, for which a general theory is yet to be developed. Following our experimental results, as a secondary goal, we propose interesting and important open questions for the theory community in relation to the design and analysis of *provably optimal* oblivious DP mechanisms.

2. Background

We briefly review the essential components of the differential privacy framework as applicable to this paper, viz., the privacy mechanism, query sensitivity, QG utility, the privacy-utility tradeoff, and popular NGMs in existing literature. Most of the material in this section is based on the paper by Ghosh, Roughgarden, and Sundararajan [3] and is intended as background material, including terminology and notation used in the remainder of the paper.

2.1. Differentially Private Mechanism

Consider a database with n rows drawn from a finite set D^n. Each row corresponds to data of an individual entity. The Hamming distance $d_H(D_1, D_2)$ between two datasets D_1, D_2 is the number of entries on which D_1 and D_2 differ. Two datasets D_1, D_2 are neighbors if and only if $d_H(D_1, D_2)$ = 1. A query q takes a database $D \in D^n$ as input and outputs the result $q(D) \in L$ in the set L of legitimate query results.

A differentially private mechanism X (also a noise generation mechanism (NGM)), is a probabilistic function from L to some range R, continuous or discrete, that adds a random noise to the true query output of $q(D)$. Typical examples of range R of answers to query q are the set of real numbers, integers, and natural numbers. Let x_{ir} denote the probability that a mechanism X outputs $r \in R$ for input (which is query output) $i = q(D) \in L$. For such a mechanism X and a parameter $\alpha = \mathrm{e}^{-\varepsilon} \in [0, 1]$, X is ε-*differentially private* if and only if $\frac{x_{D_1,r}}{x_{D_2,r}}$ lies in the interval $[\mathrm{e}^{-\varepsilon}, \mathrm{e}^{\varepsilon}]$ for every possible output $r \in R$ and pair D_1, D_2 of neighboring databases. A mechanism is *oblivious* if, for all $r \in R$, $x_{D_1 r} = x_{D_2 r}$ whenever $q(D_1) = q(D_2)$ - if the distribution of the noisy query output depends only on the true query result. We discuss examples of probabilistic noise generation mechanisms used in the existing DP literature in Section 2.5.

Intuitively, providing differential privacy implies that the probability of every response of the privacy mechanism, and hence the probability of a successful privacy attack following an interaction with the mechanism, is, up to a controllable ε factor, independent of whether a given entity "opts in" or "opts out" of the database.

2.2. Query Sensitivity

Two types of query sensitivity are typically considered when determining a proper noise mechanism - global sensitivity and smooth sensitivity. Specifically, *global sensitivity* (GS) for a query function q over the entire database domain D^n is defined as:

$$\Delta_{GS} = \max_{D_1, D_2 \in D^n : d_H(D_1, D_2) = 1} \|q(D_1) - q(D_2)\| . \quad (1)$$

That is, for a given query, GS denotes the largest difference of query outputs possible from all dataset pair combinations having a Hamming distance of one. Since the power of the noise to be added to a query output is proportional to query sensitivity values, global sensitivity is the safest way of adding noise to prevent de-anonymity. However, frequently an unnecessarily large amount of noise is added, when GS is used, thus resulting in significant difference between the observed query output and the true query output.

One approach to adding appropriate noise levels to a query output is to adopt the use of *smooth sensitivity* instead. Before we can introduce smooth sensitivity, we need to define *local sensitivity* (LS) as follows:

$$\Delta_{LS}(D_1) = \max_{D_2 \in D^n : d_H(D_1, D_2) = 1} \|q(D_1) - q(D_2)\| . \quad (2)$$

However, we cannot use LS directly to add noise to a query output, as LS is dataset D_1 dependent (i.e., sensitive to the values in the dataset), and does not preserve ε-DP. [4]. Smooth sensitivity (SS) [4] is a function of LS, and is "in between" global and local sensitivities. The definition of smooth sensitivity is given as

$$\Delta_{SS}(D_1) = \max_{k=0,1,\cdots,n} e^{-k\varepsilon} \left(\max_{D_2 \in D^n : d_H(D_1, D_2) = k} \Delta_{LS}(D_2) \right) . \quad (3)$$

Like LS, SS is dataset dependent; however, it enables the addition of an appropriate amount of noise (greater than that due to LS) to a query output, and most importantly preserves DP [4], primarily due to the appropriately extra noise added compared to that in the LS case.

2.3. Query Generator Utility

Utility Without Side Information: For a given query q, the utility to a query generator (QG) is its measure of the usefulness of the output of a differentially private mechanism X for q. One of the goals of X (in theory) is to guarantee optimal utility to every potential QG (user), independent of its side information and preferences. The notion of usefulness, however, is conceptually intuitive but intractable for quantification. *Since, in most cases the output of X will deviate from the true value of query q, a more convenient way for researchers to quantify utility is to measure its expected loss/deviation.* In the absence of any side information available to QG, let $l(i, j)$ be QG's loss function when the true answer to q is i while QG believes it to be j. In general, a loss function is likely to possess the properties of *symmetry* and *monotonicity*, i.e., the loss function would depend only on i and $|j - i|$, and would be non-decreasing in $|j - i|$. Typical examples of such functions include $l(i, j) = |j - i|$, $l(i, j) = (j - i)^2$, and the binary loss function $l_{bin}(i, j)$, defined as 0 if $i = j$, and 1 otherwise. *As in [3], here we will measure utility of QG as its expected loss (as detailed below).*

The Presence of Side Information: A QG potentially has side information pertaining to a query q, which might stem from other information sources, previous interactions with mechanism X, introspection, or common sense. For example, if q requires the count of the number of adults in Los Angeles contracting flu in December 2015, an estimated lower bound to the true query output could be the number of people buying flu drugs from a drug company in that month. An upper bound to the true query output is the total adult population of Los Angeles in the month of December. Information such as the upper bound and lower bound of the true query output serve as potential side information to the QG. One of the ways to model this side information is via a *prior probability distribution* [3]. For a given q, this prior distribution represents the belief of the QG (user) on the query output of q. Note that the use of priors as model parameters does not in any way affect the preservation or non-preservation of differential privacy; it only influences the utility of a QG to discuss the utility of a (differentially private) mechanism to a potential user.

The Net Utility Function: The net utility function for a QG is a function of both the side information he has (in terms of a prior distribution) and his loss function (user's preference).

For a query q, consider a Bayesian user with a prior p and loss function l that interacts with a differentially private mechanism X with range R. Since the range R of X need not coincide with the set L of legitimate query results (which includes the side information set), a QG, in general, must first reinterpret an output $r \in R$ of the mechanism X as a query result $j \in L$. For example, a user that observes the output "-2" from the α-geometric mechanism (Example

2.1 in [3]) might guess that the actual query result is most likely to be 0 (since the range of the true query output is non-negative). In such a case, there needs to be a remap of mechanism X with range R, which is a probabilistic function Y from R to L, with y_{rj} denoting the probability that a user reinterprets the mechanism response $r \in R$ as as a query result $j \in L$. A mechanism X and a remap Y together induce a new probabilistic mechanism $Z = Y \circ X$ with $z_{ij} = (Y \circ X)_{ij} = \sum_{r \in R} x_{ir} y_{rj}$. We define the net utility function of a QG with prior p as its expected loss with respect to a mechanism X and a remap Y, for a query $q(D)$ whose true result is i, and denote it as $EU(p, q, D, i)$ that is expressed as

$$EU(p, q, D, i) = \sum_{i \in L} p_i \sum_{j \in L} z_{ij} l(i, j), \ \sum_{i \in L} p_i = 1. \quad (4)$$

On a similar note, the net utility function of a non-Bayesian (Risk-Averse) QG that does not take into account prior information but accounts for the worst case expected loss [5], is expressed as

$$EU(q, D) = \max_{i \in L} \sum_{j \in L} z_{ij} l(i, j). \quad (5)$$

2.4. Utility-Privacy Tradeoffs

We assume that the query generator is a rational entity and would thus want to minimize its expected loss. On the other hand, the differentially private framework will need to ensure that its privacy requirements are met and that entity anonymity is preserved. For differential private frameworks, we express this conflict/tradeoff between utility and privacy for countable ranges R as two optimization problems, OPT1 and OPT2, when the QG does and does not account for its prior, respectively.

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in L} p_i \sum_{j \in L} z_{ij} l(i, j) \\ \text{subject to} \quad & \text{privacy constraint set on } \alpha, z'_{ij}s \text{ specific to } X, \\ & \sum_{j \in L} z_{ij} = 1, \ \forall i \in L, \\ & z_{ij} \geq 0, \ \forall i \in L, \forall j \in L. \end{aligned} \quad \text{(OPT1,Bayesian)}$$

$$\begin{aligned} \text{minimize} \quad & \max_{i \in L} \sum_{j \in L} z_{ij} l(i, j) \\ \text{subject to} \quad & \text{privacy constraint set on } \alpha, z'_{ij}s \text{ specific to } X, \\ & \sum_{j \in L} z_{ij} = 1, \ \forall i \in L, \\ & z_{ij} \geq 0, \ \forall i \in L, \forall j \in L. \end{aligned} \quad \text{(OPT2,Risk-Averse)}$$

In both OPT1 and OPT2, the objective function reflects the minimization of the expected loss of the QG, and the constraints, apart from the validity of problem variables, reflect the ensuring of privacy constraints specific to mechanism X. Given a query q, if user's priori (side information) p_i and preference of the loss function l are known, an optimal mechanism Z can be derived by minimizing the expected loss (Bayesian model) or the worst case loss (Risk-Averse model) subject to differential privacy.

2.5. Popular NGMs in Literature

As representative examples of NGMs, we consider three popular oblivious noise-adding mechanisms in existing literature: (i) the Laplacian [2], (ii) the Geometric [3], [5], and (iii) the Staircase [6] mechanisms, defined as follows:

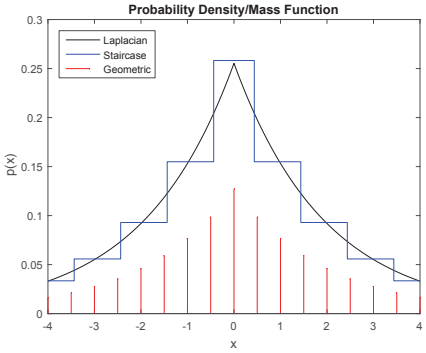

Figure 1. Laplacian, Staircase, and Geometric mechanisms.

$$\text{Laplacian } :p(x|\varepsilon, \Delta) = \frac{\varepsilon}{2\Delta} e^{-\varepsilon \frac{|x|}{\Delta}}, \forall x \in \mathbb{R} \quad (6)$$

$$\text{Geometric } :p(x|\alpha) = \frac{1+\alpha}{1-\alpha} \alpha^{|x|}, \forall x \in \mathbb{Z} \quad (7)$$

$$\text{Staircase } :p_\gamma(x|\varepsilon, \Delta) = \frac{1-\alpha}{2\Delta\sqrt{\alpha}} e^{-\varepsilon(k+[x]_\gamma)}, \forall x \in \mathbb{R} \quad (8)$$

where $0 \leq \alpha = e^{-\varepsilon} \leq 1$ is the privacy level, and Δ is the sensitivity level. In the Staircase mechanism, the rounding function $[x]_\gamma$ is defined as:

$$[x]_\gamma = \begin{cases} 0, & |x| \in [k\Delta, (k+\gamma)\Delta) \\ 1, & |x| \in [(k+\gamma)\Delta, (k+1)\Delta], \end{cases} \quad (9)$$

where $k \in \mathbb{Z}$; $0 \leq \gamma \leq 1$ controls the shape of the staircase and is set to $\frac{\sqrt{\alpha}}{1+\sqrt{\alpha}}$ in the one-dimensional case, in order to minimize the expectation of the noise amplitude. Note that the Geometric mechanism can be applied to quantized numeric query outputs using the following generalization:

$$\text{Geometric } :p(x|\alpha, \Delta, d) = d \left(\frac{1+\alpha^{\frac{d}{\Delta}}}{1-\alpha^{\frac{d}{\Delta}}} \right) \alpha^{\frac{|x|}{\Delta}}, \quad (10)$$

for all $x \in 0, \pm d, \pm 2d, \cdots$, where $d \leq \Delta$ is the quantization level of the output query, with $\frac{\Delta}{d} \in \mathbb{N}$. The conventional Geometric mechanism is a special case when $d = \Delta = 1$. A depiction of the three mechanisms is given in Fig. 1. The quantization level (resolution) here is set to 0.5 (and therefore the probability mass is one-half of the density).

3. Related Work

In this section, we briefly describe the state-of-the-art optimal oblivious DP NGM design for scalar queries. We focus on this literature to explore the drawbacks of existing research with respect to determining optimal NGMs for queries without any constraints on the availability of side information and query sensitivity, which in turn would shape directions for our ongoing and future research.

Ghosh, Roughgarden, and Sundararajan [3], [5] show that for a single count query, with special property $\Delta_{GS} = \Delta_{LS} = 1$, for a general class of utility functions, the *universally optimal* mechanism (see definition below) for preserving differential privacy is the Geometric (noise) mechanism. In [5], the authors propose mechanism analysis similar to that in [3], with similarities and differences as follows: both [5] and [3] study a count query where the query output is integer-valued, bounded, with unit sensitivity. The cost

Level 6: Universally Optimal DP Mechanism (unknown queries)
Level 5: Universally Optimal DP Mechanism (count $\Delta = 1$) :Geometric
Level 4: Optimal DP Mechanism (Bayesian)
Level 3: Optimal DP Mechanism (Bayesian, $\Delta GS = \Delta LS$)
Level 2: Optimal DP Mechanism (Risk-Averse)
Level 1: Optimal DP Mechanism (Risk-Averse, $\Delta GS = \Delta LS$) :Staircase

Figure 2. Problem Difficulty Levels (low to high) w.r.t. Optimal NGM Design

function only depends on the additive noise magnitude and is an increasing function of the noise magnitude. [5] is based on a non-Bayesian *risk-averse model*, however, where [3] is based on a Bayesian model. In [5], the authors show that although there is no optimal solution to the minimax optimization problem (to optimize privacy-utility tradeoff) for a general class of cost functions, each solution corresponding to a specific cost function instance can be derived from the same Geometric mechanism by randomly remapping. From [3] and [5], it easily follows that the Geometric mechanism is *universally optimal* (see definition below) for every count query under both, the Bayesian and risk-averse models.

Moreover, [7] states the following definition of universal optimality of a differentially private mechanism.

Definition 1. Given a query and a privacy level ε, a ε-differential private mechanism X is universally optimal if and only if every user/QG u derives as much utility from X as from the mechanism X_u which is optimally tailored to u, no matter what u's side information and preferences are.

The above definition of universal optimality reflects an extremely strong utility guarantee, regardless of the side information and preferences of QG, for Bayesian and risk-averse QG models. *Unfortunately, such a strong guarantee does not hold for general queries under either the Bayesian or risk-averse model.* Brenner et al. showed the impossibility of universally optimal oblivious mechanism for histograms, generalizations of count queries, and several other queries satisfying certain properties, for Bayesian and risk-averse users [7]. Geng et al. recently proposed a *Staircase* mechanism [6], [8] and proved optimality (not universal) *only under the risk-averse model* for general real-valued query functions where the query output can take any real value. *However, the optimality of the Staircase mechanism holds only under the following assumptions stated in [6], [8]:*

- The query output domain is the entire real domain ranging from -∞ to +∞.
- There is no side information available to a QG about the output of the query function is known.
- Local sensitivity equals global sensitivity, or the sensitivity should remain constant over all possible query outputs, so that the optimal NGM is in the family of NGMs that are query-output independent.

However, the first and last assumptions do not hold true for many query functions in practice, e.g., quite a few queries whose outputs are scalar non-integers.

Nissim, Raskhodnikova, and Smith [4] show that for certain nonlinear query functions, one can improve the query output utility by adding data-dependent noise calibrated to the smooth sensitivity of the query function, which is based on the local sensitivity of the query function. *In the model in [6], the authors use only global sensitivity of the query function to prove the optimality of the Staircase mechanism for nonlinear query functions, and assume that the local sensitivity is the same as the global sensitivity.* A summary illustration stating the difficulty (low to high) level of open problems is shown in Figure 2. Level 1 has already been tackled by researchers, whereas w.r.t. Level 5, the Geometric mechanism, as of yet, is the only proven instance of a universally optimal mechanism for the count query.

4. Challenges, Opportunities, and Our Approach

We begin with possible open research directions, followed by corresponding challenges and our approach to making progress on these open questions.

4.1. Problem Statement

Based on the above survey of the state of the art in optimal differential private mechanisms, we first state the following interesting open research directions. Each question is then mapped to a corresponding level in figure 2.

- **R1:** Do there exist universally optimal oblivious mechanisms for queries other than those stated in [7]? (*Level 6.*)
- **R2:** The authors in [6] show that the optimal mechanism for a single real-valued query under the risk-averse model is the Staircase mechanism. This optimality holds under the assumption of constant sensitivity over all query outputs, which holds in general only if the worst case global sensitivity is considered (i.e., "optimal in the sense of very large noise); but, in general, this is not true for under a better/tighter sensitivity metric such as smooth sensitivity. Thus, a natural question here is: what would the optimal mechanism be if we relax the assumption of constant noise over query outputs? (*From Level 1 to Level 2.*)
- **R3:** More ambitiously, given user preferences and side information, is there an optimal mechanism for a scalar query in the Bayesian model? This question can be treated as a relaxed version of the harder question of finding a universally optimal mechanism, as a mechanism of the latter type is optimal in the sense of arbitrary user preferences and side information. (*From Level 1 to Level 3.*)

Challenges- R1 is challenging due to the difficulty of (i) determining general or specific properties of such queries that are necessary and/or sufficient criteria for any NGM to be universally optimal for those queries, and (ii) determining the necessary and/or sufficient conditions (e.g., mathematical properties of the noise distributions) for the existence of universally optimal mechanisms for such queries. In the case of R2, relaxing the assumption of constant sensitivity leads to loss of linearity in the original (linear programming) optimization problem in [6] used to model the privacy-utility tradeoff, which is quite difficult to solve for general loss functions. In the case of R3 (i.e., given side information), determining a side-information specific optimal mechanism for general or a specific scalar query is a non-trivial task, the challenge being similar to those in (i) and (ii), but in settings when QG's have prior information on the query output.

Given these challenges, in this paper, we would like to address the following simplified but related questions and answer them via experiments assisted by some analysis:

1) "*Utility-Privacy Tradeoff of Existing Mechanisms*": With reference to R2, the Staircase mechanism is known to perform better in theory than the Laplacian mechanism (known to be the best mechanism for real-query outputs prior to the work by [6]) in the low-medium privacy regime for real query output [6]. We are interested in investigating the extent of this improvement. To this end, we will study this under arbitrary sensitivity and ε-differential privacy settings.
2) "*Presence of side information*": With reference to R3, given user preferences and partial user side information, can we figure out a heuristic differentially private mechanism that takes advantage of partial side information and performs better than

the risk-averse (non Bayesian) Staircase mechanism? If so, what would such a heuristic mechanism look like? What would be the performance gap region of the privacy-utility tradeoff?

Studying the problem of side-information specific optimal differential privacy mechanism is non-trivial. Many of the query outputs are expected to be distributed in a certain manner. For example, consider a query asking for the mean of a certain attribute of a large database. The Central Limit Theorem tells us that (assuming independent and identically distributed entries), the mean (query output) should be Gaussian distributed, no matter how the original entries are distributed. A similar idea applies to other queries such as maximum query, where we can reasonably expect a high probability of large numbers and a low probability of small numbers in a large database. More specifically, if the entries are independent and uniformly distributed, the max query output will be beta distributed over the query output domain (scaled and shifted).

Moreover, based on the open question in [5], it is still not clear whether collusion-resistance and simultaneous utility maximization hold for other types of queries (i.e., other than the count query considered in [5]). *This inspires another interesting question:* for queries other than count, how would the utility function behave (e.g., as a function of the number of QGs) when QGs interact and can potentially share information of a particular query output?

Approach - We focus on experimentally addressing the above-mentioned questions in the rest of this paper, i.e., respecting the intricacies of finding the answers to our questions in theory, instead of analytically modeling arbitrary sensitivity and side information and resolving the questions via mathematical rigor, we will run experiments for certain query functions on sampled values in the DP parameter space and the prior distribution space. Based on our observations, we will come up with conjectures whose proofs/disproofs would be open problems for the theory community working on DP. *To the best of our knowledge, ours is the first work touching upon an experimental performance evaluation of oblivious noise generating mechanisms (NGMs) for differential privacy (DP).*

4.2. Experimental Methodology

In this section, we propose our methodology to run experiments whose outcome would lead us to major conjectures about the optimality of NGMs in the presence of query side information.

4.2.1. Dataset Domains and Query Functions. Recall that here, we focus only on numeric queries and differential private (DP) mechanisms which are oblivious. Our goal is to study the utility-privacy tradeoff of three popular oblivious mechanisms, and investigate the (simplified) open questions posed above. We note that, given our goal, there is no need to perform experiments on large-scale real datasets to obtain our results, for the following reasons.

Given a database and a query result, there are three major components that DP outcomes depend on: (i) the true query output, (ii) the query sensitivity metric, and (iii) corresponding DP noise generating mechanism. However, oblivious mechanisms are not database dependent conditioned on the unperturbed query output. If two databases have same true query output, then the oblivious mechanisms apply noise to the query output in exactly the same manner, oblivious of the database. This implies that the DP mechanism output depends on the true query output and the query sensitivity. The latter again depends on the database if we consider local sensitivity, and does not depend on the database if we consider global sensitivity.

However, LS cannot be used to generate noise for a true query output because it does not preserve DP. Researchers usually consider global sensitivity or smooth sensitivity instead for adding noise. The bad news here is that, to experimentally obtain these two sensitivities, by definition, we need to investigate *all* possible databases over the entire database space, which is in practice infeasible. On the other hand, the good news is that, we can remove the need to compute sensitivity values by simply normalizing the performance metric (i.e., expected loss) by global sensitivity.

To illustrate this idea, recall that the utility (measured by expected loss) of Laplacian and Staircase mechanisms from [8] is as follows:

$$\text{Laplacian} : EL(\alpha, \Delta) = \frac{\Delta}{-\log \alpha} \tag{11}$$

$$\text{Staircase} : EL(\alpha, \Delta) = \frac{\Delta\sqrt{\alpha}}{1-\alpha}. \tag{12}$$

If we normalize above loss functions by the global sensitivity Δ, the loss function no longer depends on Δ. Particularly, the loss function of the Geometric mechanism for count query does not depend on Δ either, due to the fact that $\Delta_{GS} = \Delta_{LS} = 1$, which is

$$\text{Geometric} : EL(\alpha) = \frac{2\alpha}{1-\alpha^2}, \tag{13}$$

Therefore, without loss of generality, given the performance metric *normalized utility*, which we use in our experiments, these experiments need not be done in a database-specific way. We only need to specify the query output domain L, which can be continuous or discrete. For scalar queries, we consider mean, maximum, and count queries in our experiments.

4.2.2. Deployed (Noise-adding) Mechanisms. The popular noise-adding mechanisms we explore here are the Laplacian, Staircase, and Geometric mechanisms. The details of these mechanisms are given in section 2.5.

4.2.3. Interaction (Remap) Mechanisms. The remap function, Y, is an optimal mapping mechanism from the noisy query output, r, to the estimated result, j, in the true query output domain.

If the true query output domain is real, then Y is nothing but an impulse/identity function, since the noise we add has the highest density at the true answer with no bias. If the true query output domain is discrete, then Y should be a round() function which rounds the noisy results to the nearest legitimate discrete value.

4.2.4. Collusion in Query Results. As explained in section 4.1, we would like to understand the drop rate of expected loss corresponding to the number of cooperating customers. In our experiments, we make the following assumptions: each customer can send the same query only once, but they can attempt to extract useful information by sharing their query answers with other users asking the same query. Based on these assumptions, we study the utility-privacy tradeoffs for collusion attacks. We note here that the case for longitudinal attacks is exactly the same as that of collusion attacks because the privacy harm caused due to the same QG asking k questions on a query is the same as that caused by k QGs asking one question each on the same query and then colluding with each other on the perturbed query outputs [5].

5. Experimental Results and Analysis

In this section, we focus on addressing the simplified questions posed in Section 4.1 using experiments aided by analysis.

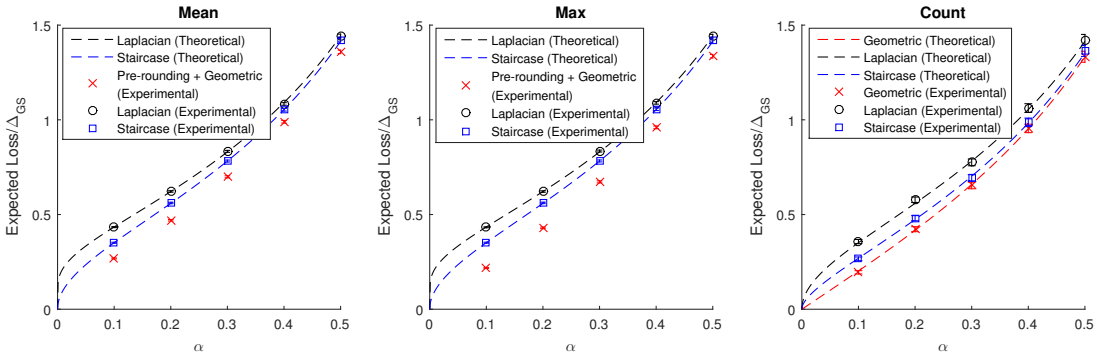

Figure 3. Utility-Privacy tradeoff and optimality of three mechanisms. Experimental results are compared to theoretical analysis.

5.1. Utility-Privacy Tradeoff of Existing Mechanisms

5.1.1. Analysis. We assume every entry in the database is a real number. In the cases of the mean and max queries, the query outputs are also real numbers. The corresponding expected disutilities were provided by previous efforts [8] and has been mentioned in (11) and (12). Note that for continuous query output domain L (e.g., the entire or bounded real domain), the Geometric mechanism cannot be applied since the perturbed query output does not cover entire (all possible query outputs in) L and thus will not satisfy DP by definition.

However, in the case of the count query, the query output is an integer. Consequently, for continuous NGMs, we have to remap the perturbed query output to integers. We derive the normalized expected loss based on its definition in section 2.3, particularly for the count query ($\Delta = 1$), as follows:

$$\text{Laplacian}: EL(\alpha) = \frac{\sqrt{\alpha}}{1-\alpha} \tag{14}$$

$$\text{Staircase}: EL(\alpha) = (1 - \frac{(1-\sqrt{\alpha})^2}{2})\frac{\sqrt{\alpha}}{1-\alpha} \tag{15}$$

Note that we do not need to revisit this for Geometric, since it has also been shown in (13).

Fig. 3 depicts our analysis results (the "Theoretical" curves) of the utility-privacy tradeoff for existing popular NGMs. Since utility is measured by expected loss, lower curves stand for better performance. We note that the Staircase mechanism outperforms the Laplacian mechanisms for the mean and maximum queries under low-medium privacy regions ($\alpha < 0.5$). Under medium-high privacy regions ($\alpha \geq 0.5$), there is essentially no statistical difference between the mechanisms, and therefore we only capture curves under low-medium privacy regions (so gaps would be more clear). Although Staircase outperforms Laplacian mechanism, the performance improvement does not seem very significant. For the count query, we find that the Geometric mechanism performs the best in our experiments. This is expected, since the Geometric mechanism is universally optimal for the count query.

5.2. Presence of Side Information

Given the setting of having side information, in this experiment, we show that even without knowing the exact distribution a priori, we could still do better than applying the risk-averse optimal mechanism blindly.

5.2.1. Scenario and Experiment Settings. In many real scenarios, we have only very limited information (if any) of the exact underlying prior distribution of query output. This is because given a query and a particular database, the corresponding query output is only one sample of the underlying prior distribution. For many cases like medical records, it is impractical to estimation the underlying prior distribution by gathering tremendous samples from tremendous databases. To understand whether we can still do better with very limited side information, in this section, we are going to design an experiment for this purpose. Even if a simple toy example showing non-trivial improvement with very limited side information is representative enough to claim the optimization problem is worthy.

Considering numeric query functions mean and maximum. Recall that for i.i.d. entries in a database, from probability theories we know that the mean and maximum will be Gaussian and Beta distributed. In our toy experiment, we define the size of database $n = 100$, the global sensitivity to be 10, and the query output domain L to be a bounded real interval $[-10, 10]$. (This is just for convenience, so that the query output of maximum query would not go to infinity). The prior of their unperturbed query output are set to be a truncated Gaussian $N(0, 1)$ and a scaled-and-shifted $Beta(n = 100, 1)$ distribution for mean and maximum query, respectively. However, in the experiment we pretend we have very obscure information (worst-case scenario under the presence of side information) about the parameters, i.e., we only know rough shapes of the distributions. More specifically, we assume that we only know that they behave like $N(\mu \in [-2, 2], \sigma)$ and $Beta(n, 1)$, with μ unclear but in a certain range, σ small and n large due to large database ($n \geq 100$).

5.2.2. Proposed Heuristic Mechanism. We propose a heuristic DP mechanism in what follows. This heuristic mechanism has two stages. The first stage is a pre-processing stage which simply rounds all numbers (the true query outputs) in $[-10, -5)$ to -10, all numbers in $[-5, 5)$ to 0, and all numbers in $[5, 10]$ to 10, i.e., there are only three possible outputs $\{-10, 0, 10\}$ after preprocessing. The second stage adds generalized Geometric noise presented in (10) with $d = \Delta = 10$ (according to the set up of this experiment) to the pre-processed output of the first stage. The true query output is then perturbed twice in our heuristic mechanism.

The idea our heuristic design is that the pre-processing is actually designed based on the side information we have, which is assumed to be Gaussian in $N(\mu \in [-2, 2], \sigma)$ and Beta in $Beta(n, 1)$ shape with small σ and large n. In other words, from side information, we know that the true result of the mean query has a high probability (due to small σ) of being in [-2,2], which is centered around 0, and the true result of the max query has a high probability (due

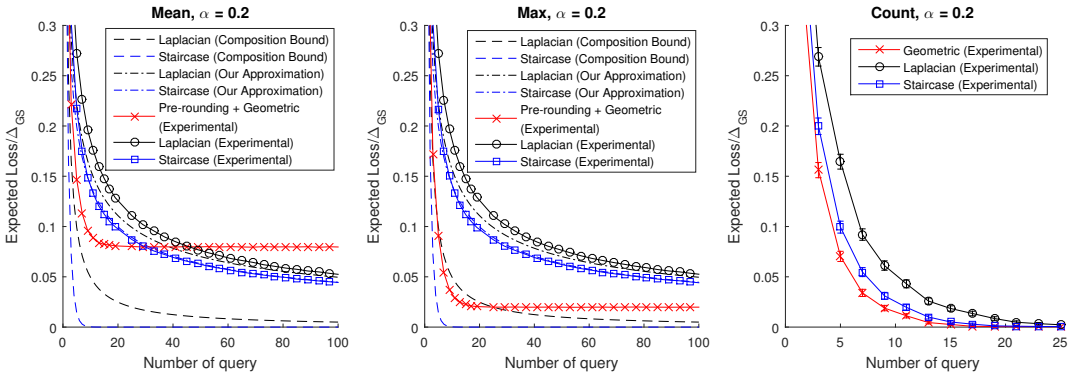

Figure 4. Drop rates of expected loss due to collusion. Experimental results are compared to theoretical approximations.

to large n) of being a large number around 10. Therefore, by discretizing the query output domain (introducing small loss first), we can then apply Geometric mechanism which is known to perform better for discrete output domain (gaining much then).

5.2.3. Experimental Results. We compare the utility-privacy tradeoff of the proposed heuristic mechanism for mean and maximum queries under different privacy levels (α). The experiments are run 10^6 times for each point and then averaged to compute the expected loss. All results are reported with 95% confidence intervals. Our experimental results are depicted in Fig. 3. The performance curve of our heuristic mechanism is marked as red crosses. We note that for both mean and max queries, with simple preprocessing and application of the Geometric mechanism, we can obtain significant performance improvement in the low-medium privacy regime ($\alpha < 0.5$). Indeed, our heuristic design is not optimal and not general in any sense, but this is not our goal in this paper. Rather, we want to show here, through experiments, the following interesting observations:

1) The pre-processing function should be designed based on available side information. We can design a meaningful pre-processing function which aids performance significantly without knowing the actual prior.
2) By simply pre-rounding and applying generalized Geometric mechanism, we can improve the performance significantly without designing anything new here. This suggests that designing a side-information specific optimal DP mechanism is non-trivial.
3) Designing a side-information specific pre-processing improves performance significantly without knowing the actual prior. However, it is not clear if the side-information specific pre-processing is indispensable. In other words, it is not clear if a side-information specific optimal DP mechanism can be designed in just a single-stage.
4) In this heuristic design, we note that the pre-rounding stage prevents the expected loss from converging to zero under collusion attacks, due to its irreversibility (see Fig. 4). This suggests interesting directions for the design of a collusion prevention mechanism.

5.3. Collusion in Query Results

Here we assume that each user can only pose the same query once (in a given time period). Let k be the number of users that cooperate with each other by sharing their perturbed query results. Dwork [9] shows that the composition of k queries, each of which is (ε,δ)-differentially private, is at least ($k\varepsilon$,$k\delta$)-differentially private. However, this bound is known to be loose [10]. Here, we derive our approximation for the trend of expected loss drop for large k. The approximation will be compared with experimental results as well as the composition bound, to validate the accuracy of the approximation and to show how loose the bound is.

5.3.1. Analysis. For k users sharing their perturbed query results, the first question we would like to ask is: According to user preference, what would be the best strategy of utilizing their results? We propose it in the following lemma.

Lemma 1. For Laplacian, Staircase and Geometric mechanisms, if user preference (loss function) is defined/known as $l(i,j) = |i-j|$, the maximum likelihood estimation (MLE) strategy for collusion in query results is to use the corresponding sample medians.

Proof: Please see Appendix for details of the proof. The definitions of l, i and j can be found in section 2.3. □

Using Lemma 1 and applying an approximation [11] of real-valued sample median distribution for large k, we then derive the normalized expected loss of the optimal collusion results for queries with continuous outputs in the following:

$$\text{Laplacian}: EL(\alpha,k) = \sqrt{\frac{2}{\pi k}}\frac{1}{(-\log\alpha)} \tag{16}$$

$$\text{Staircase}: EL(\alpha,k) = \sqrt{\frac{2}{\pi k}}\frac{\sqrt{\alpha}}{1-\alpha} \tag{17}$$

$$\text{Geometric}: EU(\alpha,k) = \frac{1}{\sqrt{2\pi k}}\frac{1+\alpha}{1-\alpha} \tag{18}$$

Not surprisingly, the expected loss drops when the number of users (k) increases. However, the drop rate is inversely proportional to $\sqrt{k}$. From above analysis, the curator can re-define a new privacy level according to the (expected) number of users (k) and the original privacy level. This approximation is expected to approach the expected loss of experimental results for large k and is thus particularly useful for estimating the trend of utility loss and re-defining new privacy levels. For the count query, however, the drop rate of expected loss is difficult to analyze (and is part of future efforts). We use experimental results to understand the trend.

5.3.2. Experimental Results. Fig. 4 illustrates how the expected loss drops as a function of the number of cooperating users (k). The experimental results of each mechanism for mean and maximum queries are compared with the corresponding approximations and composition bounds. As we can see from this figure, for uncountable numeric query

output domain (such as mean and maximum), the expected loss drops roughly inversely proportionally to $\sqrt{k}$, not much difference with our approximation. For countable numeric query output domain (such as count), our experimental results indicate that the expected loss drops much faster. This indicates that count query is particularly vulnerable to collusion attacks. That is, cooperating users could narrow down the target information fairly quickly. To prevent collusion attacks, a service provider should consider adding correlated noise between cooperating users [5], as they would not be able to remove the correlation, resulting in better privacy protection of sensitive data.

6. Conclusions and Future Work

This paper is focused on the problem of optimal DP mechanism design for one-dimensional numeric queries. We consider possible levels of optimality, consider the current state-of-the-art work in the context of these levels, and state several open questions that have not been investigated or answered by current differential privacy research. Moreover, we consider the utility-privacy tradeoff performance of existing (popular) mechanisms. In the presence of side information, a heuristic DP mechanism is proposed, largely to illustrate the non-triviality of optimal design. We also consider the effect of collusion in query results. Theoretical bounds under k-fold adaptive composition are compared with our experimental results, where collusion is based on the maximal likelihood estimation (MLE) of k query results. As a main result, we conjecture that a heuristic DP mechanism betters the Staircase and Laplace mechanism for scalar output queries in the presence of side information.

As part of future work, we plan to address the problem of determining optimal noise generation mechanisms when both side information is present and sensitivity constraints are relaxed. We also plan to tackle the problem of finding specific queries for which universally optimal mechanisms can be designed. Finally, we want to explore the design and existence of optimal NGMs (both oblivious and non-oblivious) in the multi-dimensional query output scenario under the presence or absence of side-information.

Acknowledgement

We would like to thank Dr. Jaap-Henk Hoepman and the anonymous reviewers for their valuable comments and suggestions that helped improve this paper. This work was supported in part by the Zumberge award.

References

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 111–125.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*. Springer, 2006, pp. 265–284.

[3] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1673–1693, 2012.

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

[5] M. Gupte and M. Sundararajan, "Universally optimal privacy mechanisms for minimax agents," in *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2010, pp. 135–146.

[6] Q. Geng and P. Viswanath, "The optimal mechanism in differential privacy," in *Information Theory (ISIT), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 2371–2375.

[7] H. Brenner and K. Nissim, "Impossibility of differentially private universally optimal mechanisms," *SIAM Journal on Computing*, vol. 43, no. 5, pp. 1513–1540, 2014.

[8] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1176–1184, 2015.

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology-EUROCRYPT 2006*. Springer, 2006, pp. 486–503.

[10] S. Oh and P. Viswanath, "The composition theorem for differential privacy," Technical Report, Tech. Rep., 2013.

[11] S. M. Stigler, "Studies in the history of probability and statistics. xxxii laplace, fisher, and the discovery of the concept of sufficiency," *Biometrika*, vol. 60, no. 3, pp. 439–445, 1973.

Appendix

In this section, we provide the proof to Lemma 1.

Proof of Lemma 1. Since Geometric mechanism can be treated as the sampled version of Laplacian mechanism, it is sufficient for us to prove Lemma 1 just for Laplacian and Staircase mechanisms.

Recall there are k users colluding in their perturbed query outputs $m_1, \cdots, m_k$. The unperturbed query output is their target called θ. The noise-adding mechanism X, with probability density/mass function $p_X(x)$, is applied to θ so that

$$m_i = \theta + x_i, \forall i = 1, \cdots, k. \tag{19}$$

The maximum likelihood estimation (MLE) for θ would be

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{k} \log p_X(x_i = m_i - \theta). \tag{20}$$

Recall for Laplacian, the probability density function is shown in (6). Based on (20), the MLE $\hat{\theta}$ for Laplacian mechanism can be derived as

$$\begin{aligned} \hat{\theta}_{Lap} &= \arg\max_{\theta} \left\{ k \log(\frac{\varepsilon}{2\Delta}) - \frac{\varepsilon}{\Delta} \sum_{i=1}^{k} |m_i - \theta| \right\} \\ &= \arg\min_{\theta} \left\{ \sum_{i=1}^{k} |m_i - \theta| \right\}, \end{aligned} \tag{21}$$

which is the $\hat{\theta}$ that minimizes the above k-dimensional Manhattan distance. Therefore, θ_{Lap} is the median of perturbed query outputs $m_1, \cdots, m_k$.

For staircase mechanism, the probability density function is in (8). One can easily verify that it is equivalent to the expression in the following

$$\text{Staircase} : p_\gamma(x|\varepsilon, \Delta) = \frac{1-\alpha}{2\Delta\sqrt{\alpha}} e^{-\varepsilon\left(ceil\left\{\frac{|x|}{\Delta} - \gamma\right\}\right)} \tag{22}$$

where $ceil()$ is the ceiling function. By applying (20),

$$\begin{aligned} \hat{\theta}_{\text{SC}} &= \arg\max_{\theta} \left\{ k \log(\frac{1-\alpha}{2\Delta\sqrt{\alpha}}) - \frac{\varepsilon}{\Delta} \sum_{i=1}^{k} ceil\left\{|m_i - \theta| - \gamma\Delta\right\} \right\} \\ &= \arg\min_{\theta} \left\{ \sum_{i=1}^{k} ceil\left\{|m_i - \theta| - \gamma\Delta\right\} \right\}. \end{aligned} \tag{23}$$

However, since the ceiling function $ceil(t)$ is a non-decreasing function of t, any $\hat{\theta}$ that minimize the t will also minimize $ceil(t)$ (not vice versa). Therefore, it is equivalent to say

$$\arg\min_{\theta} \left\{ \sum_{i=1}^{k} \left\{|m_i - \theta| - \gamma\Delta\right\} \right\} \subseteq \hat{\theta}_{\text{SC}} \tag{24}$$

Note that $\gamma\Delta$ is a constant and can be further removed in summation. Therefore, the median of perturbed query outputs $m_1, \cdots, m_k$ is a valid subset of the MLE $\hat{\theta}_{\text{SC}}$.