Private optimization without constraint violations

Andrés Muñoz Medina
Google Research

Umar Syed
Google Research

Sergei Vassilvitskii
Google Research

Ellen Vitercik
Carnegie Mellon University

Abstract

We study the problem of differentially private optimization with linear constraints when the right-hand-side of the constraints depends on private data. This type of problem appears in many applications, especially resource allocation. Previous research provided solutions that retained privacy but sometimes violated the constraints. In many settings, however, the constraints cannot be violated under any circumstances. To address this hard requirement, we present an algorithm that releases a nearly-optimal solution satisfying the constraints with probability 1. We also prove a lower bound demonstrating that the difference between the objective value of our algorithm's solution and the optimal solution is tight up to logarithmic factors among all differentially private algorithms. We conclude with experiments demonstrating that our algorithm can achieve nearly optimal performance while preserving privacy.

1 Introduction

Differential privacy (Dwork et al., 2006) has emerged as the standard for reasoning about user privacy and private computations. A myriad of practical algorithms exist for a broad range of problems. We can now solve tasks in a private manner ranging from computing simple dataset statistics (Nissim et al., 2007) to modern machine learning (Abadi et al., 2016). In this paper we add to this body of research by tackling a fundamental question of constrained optimization.

Specifically, we study optimization problems with linear constraints and Lipschitz objective functions. This family of optimization problems includes linear programming and quadratic programming with linear constraints, which can be used to formulate diverse problems in computer science, as well as other fields such as engineering, manufacturing, and transportation. Resource allocation is an example of a common problem in this family: given multiple agents competing for limited goods, how should the goods be distributed among the agents? Whether assigning jobs to machines or partitioning network bandwidth among different applications, these problems have convex optimization formulations with linear constraints. Given that the input to these problems may come from private user data, it is imperative that we find solutions that do not leak information about any individual.

Formally, the goal in linearly-constrained optimization is to find a vector $\boldsymbol{x}$ maximizing a function $g(\boldsymbol{x})$ subject to the constraint that $\mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}$. Due in part to the breadth of problems covered by these approaches, the past several decades have seen the development of a variety of optimization algorithms with provable guarantees, as well as fast commercial solvers. The parameters $\mathbf{A}$ and $\boldsymbol{b}$ encode data about the specific problem instance at hand, and it is easy to come up with instances where simply releasing the optimal solution would leak information about this sensitive data.

As a concrete example, suppose there is a hospital with branches located throughout a state, each of which has a number of patients with a certain disease. A specific drug is required to treat the infected patients, which the hospital can obtain from a set of pharmacies. The goal is to determine which pharmacies should supply which hospital branches while minimizing the transportation cost. In Figure 1, we present this problem as a linear program (LP). The LP is defined by sensitive information: the constraint vector reveals the number of patients with the disease at each branch.

We provide tools with provable guarantees for solving linearly-constrained optimization problems while preserving *differential privacy* (DP) (Dwork et al., 2006). Our algorithm applies to the setting where the constraint vector $\boldsymbol{b}$ depends on private data, as is the case in many resource allocation problems, such as the transportation problem above. This problem

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

$$\begin{array}{lll} \text{minimize} & \sum_{i,j} c_{ij} x_{ij} & \\ \text{such that} & \sum_{j=1}^{N} x_{ij} \leq s_i & \forall i \in [M] \\ & \sum_{i=1}^{M} x_{ij} \geq r_j & \forall j \in [N] \\ & x_{ij} \geq 0 & \forall i \in [M], \forall j \in [N]. \end{array}$$

Figure 1: The classic transportation problem formulated as a linear program. There are N hospital branches and M pharmacies. Each branch j requires r_j units of a specific drug. These values are sensitive because they reveal the number of people at each branch with a specific disease. Each pharmacy i has a supply of s_i units. It costs c_{ij} dollars to transport a unit of the drug from pharmacy i to hospital j. We use the notation x_{ij} to denote the units of the drug transported from pharmacy i to hospital j.

falls in the category of private optimization, for which there are multiple algorithms in the unconstrained case (Bassily et al., 2014; Chaudhuri et al., 2011; Kifer et al., 2012). To the best of our knowledge, only Hsu et al. (2014) and Cummings et al. (2015) study differentially private linear programming—a special case of linearly-constrained optimization. Their algorithms are allowed to violate the constraints, which can be unacceptable in many applications. In our transportation example from Figure 1, if the constraints are violated, a hospital will not receive the drugs they require or a pharmacy will be asked to supply more drugs than they have in inventory. The importance of satisfying constraints motivates this paper's central question:

How can we privately solve optimization problems while ensuring that no constraint is violated?

1.1 Results overview

Our goal is to privately solve optimization problems of the form $\max_{\boldsymbol{x} \in \mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\}$, where g is L-Lipschitz and $\boldsymbol{b}(D) \in \mathbb{R}^m$ depends on a private database D. The database is a set of individuals' records, each of which is an element of a domain $\mathcal{X}$.

To solve this problem, our (ϵ, δ)-differentially private algorithm maps $\boldsymbol{b}(D)$ to a nearby vector $\bar{\boldsymbol{b}}(D)$ and releases the vector maximizing $g(\boldsymbol{x})$ such that $\mathbf{A}\boldsymbol{x} \leq \bar{\boldsymbol{b}}(D)$. (We assume that g can be optimized efficiently under linear constraints, which is the case, for example, when g is convex.) We ensure that $\bar{\boldsymbol{b}}(D) \leq \boldsymbol{b}(D)$ coordinate-wise, and therefore our algorithm's output satisfies the constraints. This requirement precludes our use of traditional DP mechanisms: perturbing each component of $\boldsymbol{b}(D)$ using the Laplace, Gaussian, or exponential mechanisms would not result in a vector that is component-wise smaller than $\boldsymbol{b}(D)$. Instead, we extend the truncated Laplace mechanism to a multi-dimensional setting to compute $\bar{\boldsymbol{b}}(D)$.

As our main contribution, we prove that this approach is nearly optimal: we provide upper and lower bounds showing that the difference between the objective value of our algorithm's solution and the optimal solution is tight up to a factor of $O(\ln m)$ among all differentially private algorithms. First, we present an upper bound on the utility of our algorithm. We prove that if $\boldsymbol{x}(D) \in \mathbb{R}^n$ is our algorithm's output and $\boldsymbol{x}^*$ is the optimal solution to the original optimization problem, then $g(\boldsymbol{x}(D))$ is close to $g(\boldsymbol{x}^*)$. Our bound depends on the *sensitivity* Δ of the vector $\boldsymbol{b}(D)$, which equals the maximum ℓ_1-norm between any two vectors $\boldsymbol{b}(D)$ and $\boldsymbol{b}(D')$ when D and D' are neighboring, in the sense that D and D' differ on at most one individual's data. Our bound also depends on the "niceness" of the matrix $\mathbf{A}$, which we quantify using the *condition number* $\alpha(\mathbf{A})$ of the linear system[1] (Li, 1993; Mangasarian, 1981). We summarize our upper bound below (see Theorem 3.4 for the complete statement).

Theorem 1.1 (Simplified upper bound)**.** *With probability 1,*

$$\begin{aligned} & g(\boldsymbol{x}^*) - g(\boldsymbol{x}(D)) \\ \leq \; & \frac{2 \cdot \Delta \cdot L \cdot \alpha(\mathbf{A})}{\epsilon} \ln\left(\frac{m(e^\epsilon - 1)}{\delta} + 1\right). \end{aligned} \tag{1}$$

We provide a lower bound that shows that Equation (1) is tight up to a logarithmic factor.

Theorem 1.2 (Simplified lower bound)**.** *There is an infinite family of matrices* $\mathbf{A} \in \mathbb{R}^{m \times m}$*, a* 1*-Lipschitz function* $g : \mathbb{R}^m \to \mathbb{R}$*, and a mapping from databases* $D \subseteq \mathcal{X}$ *to vectors* $\boldsymbol{b}(D) \in \mathbb{R}^m$ *for any* $\Delta > 0$ *such that:*

1. *The sensitivity of* $\boldsymbol{b}(D)$ *equals* Δ*, and*
2. *For any* $\epsilon > 0$ *and* $\delta \in (0, 1/2]$*, if* $\boldsymbol{\mu}$ *is an* (ϵ, δ)*-differentially private mechanism such that* $\mathbf{A}\boldsymbol{\mu}(D) \leq \boldsymbol{b}(D)$ *with probability 1, then*

$$g(\boldsymbol{x}^*) - \mathbb{E}[g(\boldsymbol{\mu}(D))] \geq \frac{\Delta \cdot \alpha(\mathbf{A})}{4\epsilon} \ln\left(\frac{e^\epsilon - 1}{2\delta} + 1\right).$$

This lower bound matches the upper bound from Equation (1) up to a multiplicative factor of $O(\ln m)$. See Theorem 3.7 for the complete statement.

Pure differential privacy. A natural question is whether we can achieve pure $(\epsilon, 0)$-DP. In Appendix C, we prove that if $S^* := \bigcap_{D \subseteq \mathcal{X}} \{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\}$—the

[1] Here, we use the simplified notation $\alpha(\mathbf{A}) = \inf_{p \geq 1} \{\alpha_{p,q}(\mathbf{A}) \sqrt[p]{m}\}$, where $\alpha_{p,q}(\mathbf{A})$ is defined in Section 3 and $\|\cdot\|_q$ is the ℓ_q-norm under which g is L-Lipschitz.

intersection of the feasible regions across all databases D—is nonempty, then the optimal $(\epsilon, 0)$-differentially private mechanism disregards the database D and outputs $\operatorname{argmax}_{\boldsymbol{x} \in S^*} g(\boldsymbol{x})$ with probability 1. If $S^* = \emptyset$, then no $(\epsilon, 0)$-differentially private mechanism exists. Therefore, any non-trivial private mechanism must allow for a failure probability $\delta > 0$.

Experiments. We empirically evaluate our algorithm in the contexts of financial portfolio optimization and internet advertising. Our experiments show that our algorithm can achieve nearly optimal performance while preserving privacy. We also compare our algorithm to a baseline $(\epsilon, 0)$-differentially private mechanism that is allowed to violate the problem's constraints. Our experiments demonstrate that for small values of the privacy parameter ϵ, using the baseline algorithm yields a large number of violated constraints, while using our algorithm violates no constraints and incurs virtually no loss in revenue.

1.2 Additional related research

Truncated Laplace mechanism. Many papers have employed the truncated Laplace mechanism for various problems (e.g., Zhang et al., 2012; Rinott et al., 2018; Bater et al., 2018; Croft et al., 2019; Holohan et al., 2020; Geng et al., 2020). Our main contribution is not the use of this tool, but rather our proof that the truncated Laplace mechanism is the "right" tool to use for our optimization problem, with upper and lower bounds that match up to logarithmic factors.

Out of all papers employing the truncated Laplace mechanism, the one that is the most closely related to ours is by Geng et al. (2020), who study this mechanism in a one-dimensional setting. Given a query q mapping from databases D to $\mathbb{R}$, they study *query-output independent noise-adding (QIN) algorithms*. Each such algorithm μ is defined by a distribution P over $\mathbb{R}$. It releases the query output $q(D)$ perturbed by additive random noise $X \sim P$, i.e., $\mu(D) = q(D) + X$. They provide upper and lower bounds on the expected noise magnitude $|X|$ of any QIN algorithm, the upper bound equaling the expected noise magnitude of the truncated Laplace mechanism. They show that in the limit as the privacy parameters ϵ and δ converge to zero, these upper and lower bounds converge.

The Laplace mechanism is known to be a nearly optimal, general purpose $(\epsilon, 0)$-DP mechanism. While other task-specific mechanisms can surpass the utility of the Laplace mechanism (Geng et al., 2015), they all induce distributions with exponentially decaying tails. The optimality of these mechanisms comes from the fact that the ratio between the mechanism's output distributions for any two neighboring databases is exactly $\exp(\epsilon)$. Adding less noise would fail to maintain that ratio everywhere, while adding more noise would distort the query output more than necessary. Geng et al. (2020) observe that in the case of (ϵ, δ)-DP mechanisms, adding large magnitude, low probability noise is wasteful, since the DP criteria can instead be satisfied using the δ "budget" rather than maintaining the $\exp(\epsilon)$ ratio everywhere. To solve our private optimization problem, we shift and add noise to the constraints, and in our case adding large magnitude, low probability noise is not only wasteful but will cause the constraints to be violated.

Given their similar characterizations, it is not surprising that our mechanism is closely related to that of Geng et al. (2020)—the mechanisms both draw noise from a truncated Laplace distribution. The proof of our mechanism's optimality, however, is stronger than that of Geng et al.'s in several ways. First, it holds for any differentially private algorithm, not just for the limited class of QIN algorithms. Second, in the one-dimensional setting ($m = 1$)—which is the setting that Geng et al. (2020) analyze—our lower bound matches our algorithm's upper bound up to a constant factor of 8 for any $\epsilon > 0$ and $\delta \in (0, 1/2]$, not only in the limit as ϵ and δ converge to zero.

Private convex optimization. There are multiple algorithmic approaches to differentially private convex optimization. Among others, these approaches include output and objective perturbation (Chaudhuri et al., 2011), the exponential mechanism (Bassily et al., 2014), and private stochastic gradient descent (Bassily et al., 2019). The optimization problems tackled by these papers are either unconstrained, or the constraints are public information (Bassily et al., 2014). By contrast, the problems we show how to solve have private constraints. While Lagrange multipliers can transform a constrained problem into an unconstrained problem, we are not aware of a principled method for selecting Lagrange multipliers that would ensure constraint satisfaction. In fact, to *privately* find the correct multiplier seems to be an equivalent problem to the one we are proposing.

To the best of our knowledge, only Hsu et al. (2014) and Cummings et al. (2015) have studied optimization problems with private constraints. They focus on linear programs where the constraint matrix $\mathbf{A}$, constraint vector $\boldsymbol{b}$, and linear objective function may depend on private data. These papers provide algorithms that are allowed to violate the constraints, but they guarantee that each constraint will not be violated by more than some amount, denoted α, with high probability. Knowing this, an analyst could decrease each constraint by a factor of α, and then be guaranteed that with high probability, the constraints will not be violated. Compared to that approach, our algorithm has several notable advantages. First, it is not *a priori* clear what the loss in the objective value will be using their techniques, whereas we provide a simple approach with upper and lower bounds on the objective value loss that match up to logarithmic factors. Second, that approach only applies to linear programming, whereas we study the more general problem of linearly-constrained optimization. Finally, we guarantee that the constraints will not be violated with probability 1, whereas that approach would only provide a high probability guarantee. In Appendix A, we provide additional comparisons with Hsu et al.'s analysis (namely, the dichotomy between *high-* and *low-sensitivity* linear programs).

Differentially private combinatorial optimization. Several papers have studied differentially private combinatorial optimization (Gupta et al., 2010; Hsu et al., 2016), which is a distinct problem from ours, since most combinatorial optimization problems cannot be formulated only using linear constraints. Hsu et al. (2016) study a private variant of a classic allocation problem: there are n agents and k goods, and the agents' values for all 2^k bundles of the goods are private. The goal is to allocate the goods among the agents in order to maximize social welfare, while maintaining differential privacy. This is similar but distinct from the transportation problem from Figure 1: if we were to follow the formulation from Hsu et al. (2016), the transportation costs would be private, whereas in our setting, the transportation costs are public but the total demand of each hospital is private.

2 Differential privacy definition

To define differential privacy (DP), we first formally introduce the notion of a neighboring database: two databases $D, D' \subseteq \mathcal{X}$ are *neighboring*, denoted $D \sim D'$, if they differ on any one record ($|D \,\Delta\, D'| = 1$). In the hospital and pharmacy example from Figure 1, a dataset is the set of individuals at the hospitals with the particular disease.

We use the notation $\boldsymbol{x}(D) \in \mathbb{R}^n$ to denote the random variable corresponding to the vector that our algorithm releases (non-trivial DP algorithms are, by necessity, randomized). Given privacy parameters $\epsilon \geq 0$ and $\delta \in [0, 1]$, the algorithm satisfies (ϵ, δ)*-differential privacy* (*(ϵ, δ)-DP*) if for any neighboring databases D, D' and any subset $V \subseteq \mathbb{R}^n$,

$$\mathbb{P}[\boldsymbol{x}(D) \in V] \leq e^{\epsilon}\, \mathbb{P}[\boldsymbol{x}(D') \in V] + \delta.$$

Typically, ϵ is chosen to be a moderately small constant and δ is chosen to be $o(1/|D|)$, so it is negligible in the size of the dataset (Dwork et al., 2014).

3 Multi-dimensional optimization

Our goal is to privately solve multi-dimensional optimization problems of the form

$$\max_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D) \right\}, \tag{2}$$

where $\boldsymbol{b}(D) = (b(D)_1, \ldots, b(D)_m)$ is a vector in $\mathbb{R}^m$ and g is an L-Lipschitz function according to an ℓ_q-norm $|| \cdot ||_q$ for $q \geq 1$. Preserving privacy while ensuring the constraints are always satisfied is impossible if the feasible regions change drastically across databases. For example, if D and D' are neighboring databases with disjoint feasible regions, there is no (ϵ, δ)-DP mechanism that always satisfies the constraints with $\delta < 1$ (see Lemma B.3 in Appendix B). To circumvent this impossibility, we assume that the intersection of the feasible regions across databases is nonempty. This is satisfied, for example, if the origin is always feasible. For instance, any private optimization problem that can be modeled as a max-flow problem with private edge capacities satisfies this assumption, since the zero flow is always feasible.

Assumption 3.1. *The set* $S^* := \bigcap_{D \subseteq \mathcal{X}} \left\{ \boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D) \right\}$ *is non-empty.*

In our approach, we map each vector $\boldsymbol{b}(D)$ to a random variable $\bar{\boldsymbol{b}}(D) \in \mathbb{R}^m$ and release

$$\boldsymbol{x}(D) \in \operatorname{argmax}_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \bar{\boldsymbol{b}}(D) \right\}. \tag{3}$$

To formally describe our approach, we use the notation $\Delta = \max_{D \sim D'} \left\| \boldsymbol{b}(D) - \boldsymbol{b}(D') \right\|_1$ to denote the constraint vector's *sensitivity*. (In the example introduced in Figure 1, if a hospital requires r units of the drug to treat one patient, then the sensitivity of the solution will be r. We define the i^{th} component of $\bar{\boldsymbol{b}}(D)$ to be $\bar{b}(D)_i = \max\left\{ b(D)_i - s + \eta_i, b_i^* \right\}$, where $s = \frac{\Delta}{\epsilon} \ln\left(\frac{m(e^{\epsilon} - 1)}{\delta} + 1 \right)$, η_i is drawn from the truncated Laplace distribution with support $[-s, s]$ and scale $\frac{\Delta}{\epsilon}$, and $b_i^* = \inf_D \left\{ b(D)_i \right\}$. In Lemmas B.4 and B.5 in Appendix B, we prove that $S^* = \left\{ \boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq (b_1^*, \ldots, b_m^*) \right\}$, which allows us to prove that Equation (3) is feasible. Algorithm 1 displays the pseudo-code.

First, we prove that our algorithm satisfies differential privacy. We use the notation $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_m)$ to denote a random vector where each component is drawn i.i.d. from the truncated Laplace distribution with support $[-s, s]$ and scale $\frac{\Delta}{\epsilon}$. We also use the notation $\boldsymbol{b}(D) - s + \boldsymbol{\eta} = (b(D)_1 - s + \eta_1, \ldots, b(D)_m - s + \eta_m)$. The proof of the following theorem is in Appendix B.

Algorithm 1 Private linear programming algorithm

Input: Objective $g : \mathbb{R}^n \to \mathbb{R}$, constraint matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, constraint vector $\boldsymbol{b}(D) = (b(D)_1, \dots, b(D)_m) \in \mathbb{R}^m$, privacy parameters $\epsilon > 0$ and $\delta \in (0, 1]$, sensitivity $\Delta \geq 0$, and $b_i^* = \inf_D \{b(D)_i\}$ for all $i \in [m]$.

1: Define $s = \frac{\Delta}{\epsilon} \ln\left(\frac{m(e^\epsilon - 1)}{\delta} + 1\right)$.

2: Let $\bar{\boldsymbol{b}}(D)$ be such that $\bar{b}(D)_i = \max\{b(D)_i - s + \eta_i, b_i^*\}$, where η_i is drawn from the truncated Laplace distribution with support $[-s, s]$ and scale $\frac{\Delta}{\epsilon}$.

Output: $\boldsymbol{x}(D) \in \operatorname{argmax}_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \bar{\boldsymbol{b}}(D)\}$.

Theorem 3.2. *The mapping $D \mapsto \boldsymbol{b}(D) - s + \boldsymbol{\eta}$ preserves (ϵ, δ)-differential privacy.*

Since differential privacy is immune to post-processing (Dwork et al., 2014), Theorem 3.2 implies our algorithm is differentially private.

Corollary 3.3. *The mapping $D \mapsto \boldsymbol{x}(D)$ is (ϵ, δ)-differentially private.*

We next provide a bound on the quality of our algorithm, which measures the difference between the optimal solution $\max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\}$ and the solution our algorithm returns $g(\boldsymbol{x}(D))$. Our bound depends on the "niceness" of the matrix $\mathbf{A}$, as quantified by the linear system's *condition number* (Li, 1993) $\alpha_{p,q}(\mathbf{A})$. Li (1993) proved that this value sharply characterizes the extent to which a change in the vector $\boldsymbol{b}$ causes a change in the feasible region, so it makes sense that it appears in our quality guarantees. Given a norm $||\cdot||_p$ on $\mathbb{R}^m$ where $p \geq 1$, we use the notation $||\cdot||_{p^*}$ to denote the dual norm where $\frac{1}{p} + \frac{1}{p^*} = 1$. The linear system's *condition number* is defined as

$$\alpha_{p,q}(\mathbf{A}) = \sup \left\{ \|\boldsymbol{u}\|_{p^*} : \begin{array}{l} \left\|\mathbf{A}^T\boldsymbol{u}\right\|_{q^*} = 1, \boldsymbol{u} \geq \mathbf{0}, \\ \text{and the rows of } \mathbf{A} \\ \text{corresponding to the} \\ \text{nonzero entries of } \boldsymbol{u} \text{ are} \\ \text{linearly independent} \end{array} \right\}.$$

When $\mathbf{A}$ is nonsingular and $p = q = 2$, $\alpha_{p,q}(\mathbf{A})$ is at most the inverse of the minimum singular value, $\sigma_{\min}(\mathbf{A})^{-1}$. This value $\sigma_{\min}(\mathbf{A})^{-1}$ is closely related to the matrix $\mathbf{A}$'s condition number (which is distinct from $\alpha_{p,q}(\mathbf{A})$, the linear system's condition number), which roughly measures the rate at which the solution to $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ changes with respect to a change in $\boldsymbol{b}$.

We now prove our quality guarantee, which bounds the difference between the optimal solution to the original optimization problem (Equation (2)) and that of the privately transformed problem (Equation (3)).

Theorem 3.4. *Suppose Assumption 3.1 holds and the function $g : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz with respect to an ℓ_q-norm $\|\cdot\|_q$ on $\mathbb{R}^n$. With probability 1,*

$$\begin{aligned} &\max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\} - g(\boldsymbol{x}(D)) \\ \leq\ &\frac{2L\Delta}{\epsilon} \cdot \inf_{p\geq 1} \left\{\alpha_{p,q}(\mathbf{A}) \sqrt[p]{m}\right\} \cdot \ln\left(\frac{m\left(e^\epsilon - 1\right)}{\delta} + 1\right). \end{aligned}$$

Proof. Let $\boldsymbol{b}$ be an arbitrary vector in the support of $\bar{\boldsymbol{b}}(D)$ and let $S = \{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$. Let $\boldsymbol{x}^*$ be an arbitrary point in $\operatorname{argmax}_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\}$ and let $\bar{\boldsymbol{x}}$ be an arbitrary vector in S. We know that

$$\begin{aligned} &\max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\} - \max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\} \\ =\ &g(\boldsymbol{x}^*) - \max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\} \\ =\ &g(\boldsymbol{x}^*) - g(\bar{\boldsymbol{x}}) + g(\bar{\boldsymbol{x}}) - \max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}. \end{aligned}$$

Since $\bar{\boldsymbol{x}} \in S = \{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$, we know that $g(\bar{\boldsymbol{x}}) \leq \max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$. Therefore,

$$\begin{aligned} &\max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\} - \max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\} \\ \leq\ &g(\boldsymbol{x}^*) - g(\bar{\boldsymbol{x}}) \leq L \cdot \|\boldsymbol{x}^* - \bar{\boldsymbol{x}}\|_q. \end{aligned} \tag{4}$$

To simplify notation, let $M = g(\boldsymbol{x}^*) - \max\{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$. Equation (4) shows that for every $\bar{\boldsymbol{x}} \in S$, $\frac{M}{L} \leq \|\boldsymbol{x}^* - \bar{\boldsymbol{x}}\|_q$. Meanwhile, from work by Li (1993), we know that for any ℓ_p-norm $\|\cdot\|_p$,

$$\inf_{\bar{\boldsymbol{x}}\in S} \|\boldsymbol{x}^* - \bar{\boldsymbol{x}}\|_q \leq \alpha_{p,q}(\mathbf{A}) \cdot \|\boldsymbol{b}(D) - \boldsymbol{b}\|_p. \tag{5}$$

By definition of the infimum, this means that $M \leq L \cdot \alpha_{p,q}(\mathbf{A}) \cdot \|\boldsymbol{b}(D) - \boldsymbol{b}\|_p$. This inequality holds for any $\boldsymbol{b}$ in the support of $\bar{\boldsymbol{b}}(D)$ and with probability 1,

$$\left\|\boldsymbol{b}(D) - \bar{\boldsymbol{b}}(D)\right\|_p \leq \frac{2\Delta\sqrt[p]{m}}{\epsilon} \ln\left(\frac{m\left(e^\epsilon - 1\right)}{\delta} + 1\right).$$

Therefore, the theorem holds. □

In the following examples, we instantiate Theorem 3.4 in several specific settings.

Example 3.5 (Nonsingular constraint matrix)**.** When $\mathbf{A}$ is nonsingular, setting $\|\cdot\|_p = \|\cdot\|_q = \|\cdot\|_2$ implies

$$\begin{aligned} &\max_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\} - g(\boldsymbol{x}(D)) \\ \leq\ &\frac{2 \cdot \Delta \cdot \sqrt{m} \cdot L}{\epsilon \cdot \sigma_{\min}(\mathbf{A})} \ln\left(\frac{m\left(e^\epsilon - 1\right)}{\delta} + 1\right). \end{aligned}$$

Example 3.6 (Strongly stable linear inequalities)**.** We can obtain even stronger guarantees when the system of inequalities $\mathbf{A}\boldsymbol{x} < \mathbf{0}$ has a solution. In that case, the set $\{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ is non-empty for any vector $\boldsymbol{b}$ (Mangasarian and Shiau, 1987), so we need not make Assumption 3.1. Moreover, when $\|\cdot\|_q$ and $\|\cdot\|_p$ both equal the ℓ_∞-norm and $\mathbf{A}\boldsymbol{x} < \mathbf{0}$ has a solution, we can replace $\alpha_{p,q}(\mathbf{A})$ in Theorem 3.4 with the following solution to a linear program:

$$\bar{\alpha}(\mathbf{A}) = \max_{(\boldsymbol{u},\boldsymbol{z})\in\mathbb{R}^{m+n}} \left\{ \mathbf{1}\cdot\boldsymbol{u} : \begin{array}{l} -\boldsymbol{z} \leq \boldsymbol{u}^\top \mathbf{A} \leq \boldsymbol{z}, \\ \boldsymbol{u} \geq \mathbf{0}, \text{ and } \mathbf{1}\cdot\boldsymbol{z} = 1 \end{array} \right\}.$$

This is because in the proof of Theorem 3.4, we can replace Equation (5) with $\inf_{\bar{\boldsymbol{x}}\in S} \|\boldsymbol{x}^* - \bar{\boldsymbol{x}}\|_q \leq \bar{\alpha}(\mathbf{A}) \cdot \|\boldsymbol{b}(D) - \boldsymbol{b}\|_p$ (Mangasarian and Shiau, 1987).

We now present our main result. We prove that the quality guarantee from Theorem 3.4 is tight up to a factor of $O(\log m)$.

Theorem 3.7. *Let $\mathbf{A} \in \mathbb{R}^{m\times m}$ be an arbitrary diagonal matrix with positive diagonal entries and let $g : \mathbb{R}^m \to \mathbb{R}$ be the function $g(\boldsymbol{x}) = \langle 1, \boldsymbol{x}\rangle$. For any $\Delta > 0$, there exists a mapping from databases $D \subseteq \mathcal{X}$ to vectors $\boldsymbol{b}(D) \in \mathbb{R}^m$ such that:*

1. *The sensitivity of $\boldsymbol{b}(D)$ equals Δ, and*

2. *For any $\epsilon > 0$ and $\delta \in (0, 1/2]$, if $\boldsymbol{\mu}$ is an (ϵ,δ)-differentially private mechanism such that $\mathbf{A}\boldsymbol{\mu}(D) \leq \boldsymbol{b}(D)$ with probability 1, then*

$$\begin{aligned} &\max\{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}(D)\} - \mathbb{E}[g(\boldsymbol{\mu}(D))] \\ \geq\ & \frac{\Delta}{4\epsilon} \cdot \inf_{p\geq 1} \left\{\alpha_{p,1}(\mathbf{A}) \sqrt[p]{m}\right\} \cdot \ln\left(\frac{e^\epsilon - 1}{2\delta} + 1\right). \end{aligned}$$

Since the objective function g is 1-Lipschitz under the ℓ_1-norm, this lower bound matches the upper bound from Theorem 3.4 up to a factor of $O(\log m)$. The full proof of Theorem 3.7 is in Appendix B.

Proof sketch of Theorem 3.7. For ease of notation, let $t = \frac{1}{\epsilon}\ln\left(\frac{e^\epsilon - 1}{2\delta} + 1\right)$. Notice that $\delta \leq \frac{1}{2}$ implies $t \geq 1$. For each vector $\boldsymbol{d} \in \mathbb{Z}^m$, let $D_{\boldsymbol{d}}$ be a database where for any $\boldsymbol{d}, \boldsymbol{d}' \in \mathbb{Z}^m$, if $\|\boldsymbol{d} - \boldsymbol{d}'\|_1 \leq 1$, then $D_{\boldsymbol{d}}$ and $D_{\boldsymbol{d}'}$ are neighboring. Let $\boldsymbol{b}(D_{\boldsymbol{d}}) = \Delta\boldsymbol{d}$ and let $a_1, \ldots, a_m > 0$ be the diagonal entries of $\mathbf{A}$. Since $\mathbf{A}\boldsymbol{\mu}(D_{\boldsymbol{d}}) \leq \boldsymbol{b}(D_{\boldsymbol{d}})$ with probability 1, $\boldsymbol{\mu}(D_{\boldsymbol{d}})$ must be coordinate-wise smaller than $\Delta\left(\frac{d_1}{a_1}, \ldots, \frac{d_m}{a_m}\right)$.

We begin by partitioning the support of $\boldsymbol{\mu}(D_{\boldsymbol{d}})$ so that we can analyze $\mathbb{E}[g(\boldsymbol{\mu}(D_{\boldsymbol{d}}))]$ using the law of total expectation. We organize this partition using axis-aligned rectangles. Specifically, for each index $i \in [m]$,

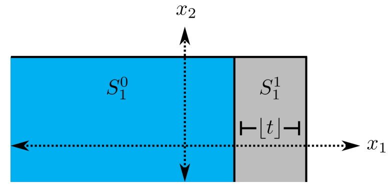

Figure 2: This figure illustrates the partition of $\mathbb{R}^2$ into S_1^0 (the left blue shaded region) and S_1^1 (the right grey shaded region). Assuming $\mathbf{A}$ is the identity matrix, the right vertical edge of S_1^1 lines up with $x_1 = d_1$ and the left vertical edge of S_1^1 lines up with $x_1 = d_1 - \lfloor t\rfloor$. The top horizontal edges of both S_1^0 and S_1^1 line up with $x_2 = d_2$.

let S_i^0 be the set of vectors $\boldsymbol{x} \in \mathbb{R}^m$ whose i^{th} components are smaller than $\frac{\Delta}{a_i}(d_i - \lfloor t\rfloor)$:

$$S_i^0 = \left\{\boldsymbol{x} \in \mathbb{R}^m : x_i \leq \frac{\Delta}{a_i}(d_i - \lfloor t\rfloor)\right\}.$$

Similarly, let

$$S_i^1 = \left\{\boldsymbol{x} \in \mathbb{R}^m : \frac{\Delta}{a_i}(d_i - \lfloor t\rfloor) < x_i \leq \frac{\Delta d_i}{a_i}\right\}.$$

See Figure 2 for an illustration of these regions. For any vector $\boldsymbol{I} \in \{0,1\}^m$, let $S_{\boldsymbol{I}} = \cap_{i=1}^m S_i^{I_i}$. The sets $S_{\boldsymbol{I}}$ partition the support of $\boldsymbol{\mu}(D_{\boldsymbol{d}})$ into rectangles. Therefore, by the law of total expectation,

$$\begin{aligned} &\mathbb{E}[g(\boldsymbol{\mu}(D_{\boldsymbol{d}}))] = \\ &\sum_{\boldsymbol{I}\in\{0,1\}^m} \mathbb{E}[g(\boldsymbol{\mu}(D_{\boldsymbol{d}})) \mid \boldsymbol{\mu}(D_{\boldsymbol{d}}) \in S_{\boldsymbol{I}}]\,\mathbb{P}[\boldsymbol{\mu}(D_{\boldsymbol{d}}) \in S_{\boldsymbol{I}}]. \end{aligned} \tag{6}$$

When we condition on the vector $\boldsymbol{\mu}(D_{\boldsymbol{d}})$ being contained in a rectangle $S_{\boldsymbol{I}}$, our analysis of the expected value of $g(\boldsymbol{\mu}(D_{\boldsymbol{d}}))$ is simplified. Suppose that $\boldsymbol{\mu}(D_{\boldsymbol{d}}) \in S_{\boldsymbol{I}}$ for some $\boldsymbol{I} \in \{0,1\}^m$. If $I_i = 0$, then we know that $\mu(D_{\boldsymbol{d}})_i \leq \frac{\Delta}{a_i}(d_i - \lfloor t\rfloor)$. Meanwhile, if $I_i = 1$, then $\mu(D_{\boldsymbol{d}})_i \leq \frac{\Delta d_i}{a_i}$ since $\mathbf{A}\boldsymbol{\mu}(D_{\boldsymbol{d}}) \leq \boldsymbol{b}(D_{\boldsymbol{d}})$ with probability 1. Since $g(\boldsymbol{x}) = \langle \mathbf{1}, \boldsymbol{x}\rangle$, we have that for each $\boldsymbol{I} \in \{0,1\}^m$,

$$\begin{aligned} &\mathbb{E}[g(\boldsymbol{\mu}(D_{\boldsymbol{d}})) \mid \boldsymbol{\mu}(D_{\boldsymbol{d}}) \in S_{\boldsymbol{I}}] \\ \leq\ & \sum_{i=1}^m \frac{\Delta(d_i - \lfloor t\rfloor)}{a_i}\mathbf{1}_{\{I_i=0\}} + \frac{\Delta d_i}{a_i}\mathbf{1}_{\{I_i=1\}} \\ =\ & \sum_{i=1}^m \frac{\Delta d_i}{a_i} - \frac{\Delta\lfloor t\rfloor}{a_i}\mathbf{1}_{\{I_i=0\}}. \end{aligned}$$

Combining this inequality with Equation (6) and rearranging terms, we are able to prove that

$$\begin{aligned} &\mathbb{E}[g(\boldsymbol{\mu}(D_{\boldsymbol{d}}))] \\ \leq\ & \Delta\sum_{i=1}^m \frac{d_i}{a_i} - \Delta\lfloor t\rfloor \sum_{i=1}^m \frac{1}{a_i}\mathbb{P}\left[\boldsymbol{\mu}(D_{\boldsymbol{d}}) \in S_i^0\right] \end{aligned} \tag{7}$$

(see the full proof in Appendix B for details).

We use the definition of differential privacy to show that for all $i \in [m]$, $\mathbb{P}\left[\boldsymbol{\mu}\left(D_{\boldsymbol{d}}\right) \in S_i^0\right] > \frac{1}{2}$, which allows us to simplify Equation (7). Intuitively this holds since $\boldsymbol{\mu}\left(D_{\boldsymbol{d}}\right)$ cannot have too much probability mass in each set S_i^1, as there are neighboring databases that have zero probability mass in subsets of this region. More precisely, we show that $\mathbb{P}\left[\boldsymbol{\mu}\left(D_{\boldsymbol{d}}\right) \in S_i^0\right] > \delta \sum_{j=0}^{\lfloor t \rfloor - 1} e^{\epsilon j}$. Our choice of t then implies that $\mathbb{P}\left[\boldsymbol{\mu}\left(D_{\boldsymbol{d}}\right) \in S_i^0\right] > \frac{1}{2}$.

This inequality, Equation (7), and the fact that $t \geq 1$ together imply that

$$\mathbb{E}\left[g\left(\boldsymbol{\mu}\left(D_{\boldsymbol{d}}\right)\right)\right] < \Delta \sum_{i=1}^m \frac{d_i}{a_i} - \frac{\Delta t}{4} \sum_{i=1}^m \frac{1}{a_i}.$$

Since $\max\left\{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\left(D_{\boldsymbol{d}}\right)\right\} = \Delta \sum_{i=1}^m \frac{d_i}{a_i}$, we have that

$$\begin{aligned} &\max\left\{g(\boldsymbol{x}) : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\left(D_{\boldsymbol{d}}\right)\right\} - \mathbb{E}\left[g\left(\boldsymbol{\mu}\left(D_{\boldsymbol{d}}\right)\right)\right] \\ \geq\ & \frac{\Delta}{4\epsilon}\left(\sum_{i=1}^m \frac{1}{a_i}\right) \ln\left(\frac{e^\epsilon - 1}{2\delta} + 1\right). \end{aligned}$$

Finally, we prove that $\inf_{p \geq 1} \alpha_{p,1}(\mathbf{A}) \sqrt[p]{m} \leq \alpha_{\infty,1}(\mathbf{A}) = \sum_{i=1}^m \frac{1}{a_i}$, which implies that the theorem statement holds. Since $\mathbf{A}$ is diagonal,

$$\alpha_{\infty,1}(\mathbf{A}) = \sup_{\boldsymbol{u} \geq \mathbf{0}} \left\{\|\boldsymbol{u}\|_1 : u_i a_i \leq 1, \forall i \in [m]\right\} = \sum_{i=1}^m \frac{1}{a_i}.$$

Moreover, since $\alpha_{\infty,1}(\mathbf{A}) \in \{\alpha_{p,1}(\mathbf{A}) \sqrt[p]{m} : p \geq 1\}$, we have that $\inf_{p \geq 1} \alpha_{p,1}(\mathbf{A}) \sqrt[p]{m} \leq \alpha_{\infty,1}(\mathbf{A})$. Therefore, the theorem statement holds. □

This theorem demonstrates that our algorithm's loss (Theorem 3.4) is tight up to a factor of $O(\log m)$ among all differentially private mechanisms.

4 Experiments

In this section, we present empirical evaluations of our algorithm in several settings: financial portfolio optimization and internet advertising.

4.1 Portfolio optimization

Suppose a set of individuals pool their money to invest in a set of n assets over a period of time. The amount contributed by each individual is private, except to the trusted investment manager. Let $b(D)$ be the total amount of money the investors (represented by a database D) contribute. We let x_i denote the amount

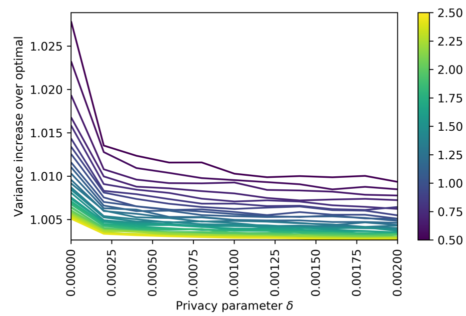

Figure 3: Quality in the portfolio optimization application. The plot shows the multiplicative increase in the objective function value of our algorithm's solution—for various choices of ϵ and δ—over the objective function value of the optimal solution to the original optimization problem (Equation (8)). Darker shading corresponds to lower values of ϵ and hence stronger privacy. See Section 4.1 for details.

of asset i held throughout the period, with x_i in dollars, at the price at the beginning of the period. We adopt the classic Markowitz (1952) portfolio optimization model. The return of each asset is represented by the random vector $\boldsymbol{p} \in \mathbb{R}^n$, which has known mean $\bar{\boldsymbol{p}}$ and covariance Σ. Therefore with portfolio $\boldsymbol{x} \in \mathbb{R}^n$, the return r is a (scalar) random variable with mean $\bar{\boldsymbol{p}} \cdot \boldsymbol{x}$ and variance $\boldsymbol{x}^\top \Sigma \boldsymbol{x}$. The choice of a portfolio $\boldsymbol{x}$ involves a trade-off between the return's mean and variance. Given a minimum return r_{min}, the goal is to solve the following quadratic program while keeping the budget $b(D)$ private:

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{x} \geq 0} & \boldsymbol{x}^\top \Sigma \boldsymbol{x} \\ \text{such that} & \bar{\boldsymbol{p}} \cdot \boldsymbol{x} \geq r_{min} \ \wedge\ \boldsymbol{x} \cdot \mathbf{1} \leq b(D) \end{array} \tag{8}$$

We run experiments using real-world data from stocks included in the Dow Jones Industrial Average, compiled by Bruni et al. (2016). They collected weekly linear returns for 28 stocks over the course of 1363 weeks. The mean vector $\bar{\boldsymbol{p}} \in \mathbb{R}^{28}$ is the average of these weekly returns and the covariance matrix $\Sigma \in \mathbb{R}^{28 \times 28}$ is the covariance of the weekly returns.

In Figure 3, we analyze the quality of our algorithm. First, we set the number of individuals n to be 1000. Then, we define each element of the database (money given by individuals to an investor) as a draw from the uniform distribution between 0 and 1, so $b(D)$ equals the sum of these n random variables. The sensitivity of $b(D)$ is therefore $\Delta = 1$. We set the minimum return r_{min} to be 2.5. We calculate the objective value $v^* \in \mathbb{R}$ of the optimal solution to Equation (8). Then, for $\delta \in \left[\frac{1}{n^2}, 0.002\right]$ and $\epsilon \in [0.5, 2.5]$, we run our algorithm 50 times and calculate the average objective value $\hat{v}_{\epsilon,\delta} \in \mathbb{R}$ of the optimal solutions.

In Figure 3, we plot $\frac{\hat{v}_{\epsilon,\delta}}{v^*}$. We see that even strict values for the privacy parameters do not lead to a significant degradation in the value of the objective function. For example, setting $\epsilon = 0.5$ and $\delta = 2.5 \cdot 10^{-4}$ increases the value of the objective function by about 1%.

In Appendix D, we perform the same experiment with the number n of investors in $\{500, 1000, 1500\}$ and the minimum return r_{min} in the interval $[1, 5]$. We obtain plots that are similar to Figure 3. As we describe in Appendix D, we find that there is a sweet spot for the parameter choices n and r_{min}. If r_{min} is too small, the budget constraint is non-binding with or without privacy, so the variance increase over optimal is always 1. Meanwhile, if r_{min} is too large, then the original quadratic program (Equation (8)) is infeasible.

4.2 Internet advertising

Many internet publishers hire companies called *supply-side platforms (SSPs)* to manage their advertising inventory. A publisher using an SSP partitions its website's pages into M groups, and informs the SSP of the number n_j of impressions (i.e., visiting users) available in each group j. For example, an online newspaper might have a sports section, a fashion section, and so on. The SSP relays the list of inventory groups to N potential advertisers, and each advertiser i responds with the monetary amount $c_{ij} \geq 0$ they are willing to pay per impression from each group j, and also their budget $b(D)_i \geq 0$ for overall spending on the publisher's website, where D represents advertisers' confidential business information, such as their financial health and strategic priorities. The SSP then allocates x_{ij} impressions from each group j to each advertiser i so as to maximize the publisher's revenue while respecting both the impression supply constraints and advertiser budget constraints:

$$\begin{aligned} &\text{maximize } \textstyle\sum_{ij} c_{ij} x_{ij} \\ &\text{such that } \textstyle\sum_{i=1}^{N} x_{ij} \leq n_j \qquad \text{for } j \in [M] \\ &\qquad\qquad \textstyle\sum_{j=1}^{M} c_{ij} x_{ij} \leq b(D)_i \quad \text{for } i \in [N] \\ &\qquad\qquad x_{ij} \geq 0. \end{aligned} \tag{9}$$

This linear program is similar to the transportation problem from Figure 1.

Existing algorithms for private optimization are not guaranteed to output a solution that satisfies all the problem constraints, so we explore how often those algorithms violate the constraints when applied to the advertising problem in Equation (9). The algorithm most closely related to ours is by Hsu et al. (2014), but our settings do not quite match: they require that the optimal solution have constant norm across all possible private database. If this is not the case (and it is not for our advertising problem), Hsu et al. (2014) recommend normalizing the problem parameters by the norm of the optimal non-private solution (which itself is a sensitive value). However, this will necessarily impact the problem's sensitivity parameter Δ, and Hsu et al. (2014) do not provide guidance on how to quantify this impact, though knowing this sensitivity is crucial for running the algorithm.

Therefore, we compare our algorithm with an alternative baseline. We run experiments that use two algorithms to transform each advertiser's budget $b(D)_i$ in Equation (9) to a private budget $\bar{b}(D)_i$. Both algorithms set $\bar{b}(D)_i = \max\{b(D)_i - s + \eta, 0\}$, where $s = \frac{\Delta}{\epsilon} \ln\left(\frac{N(e^\epsilon - 1)}{\delta} + 1\right)$ for privacy parameters ϵ, δ and sensitivity parameter Δ, and η is a random variable. The first algorithm follows our method described in Section 3 and draws η from the truncated Laplace distribution with support $[-s, s]$ and scale $\frac{\Delta}{\epsilon}$. The baseline algorithm instead draws η from the Laplace distribution with scale $\frac{\Delta}{\epsilon}$, and thus is $(\epsilon, 0)$-differentially private. Both algorithms use noise distributions with roughly the same shape, but only our algorithm is guaranteed to satisfy the original constraints.

Our experiments consist of simulations with parameters chosen to resemble real data from an actual SSP. The publisher has $M = 200$ inventory groups, and there are $N = 10$ advertisers who wish to purchase inventory on the publisher's website. The amount c_{ij} each advertiser i is willing to pay per impression from each group j is \$0 with probability 0.2, and drawn uniformly from $[\$0, \$1]$ with probability 0.8. The number of impressions n_j per group j is 10^7, and each advertiser's budget $b(D)_i$ is drawn uniformly from $[\$10^7 - \Delta/2, \$10^7 + \Delta/2]$, where $\Delta = \$10^2$ is also the sensitivity of the budgets with respect to the private database D. The results for various values of the privacy parameter ϵ (with the privacy parameter δ fixed at 10^{-4}) are shown in Figure 4, where every data point on the plot is the average of 400 simulations.

Figure 4 shows that for small values of ϵ, using the baseline algorithm yields a large number of violated constraints, while using our algorithm violates no constraints and incurs virtually no loss in revenue.

5 Conclusions

We presented a differentially private method for solving linearly-constrained optimization problems, where the right-hand side of the constraints $\mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}$ depends on private data, and where the constraints must always be satisfied. We showed that our algorithm is nearly optimal: its loss is tight up to a factor of $O(\log m)$ among all DP algorithms. Empirically, we used real

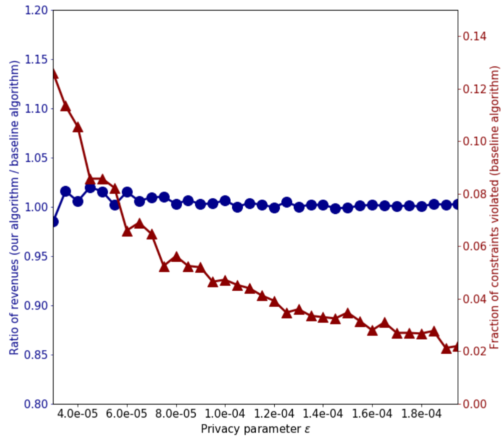

Figure 4: Quality in the advertising application. Ratio of the revenue of our algorithm's solution and that of the baseline algorithm's solution (circle markers, left vertical axis), and fraction of constraints in the original optimization problem (Equation (9)) violated by the baseline algorithm (triangle markers, right vertical axis). See Section 4.2 for details.

and synthetic datasets to show that our algorithm returns nearly optimal solutions in realistic settings. A natural direction for future research would be to allow the matrix $\mathbf{A}$ to also depend on private data.

Acknowledgments

This research was supported in part by a fellowship from Carnegie Mellon University's Center for Machine Learning and Health and an IBM PhD Fellowship to E.V.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the Conference on Computer and Communications Security (CCS)*, 2016.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Proceedings of NeurIPS*, pages 11279–11288, 2019.

Johes Bater, Xi He, William Ehrich, Ashwin Machanavajjhala, and Jennie Rogers. Shrinkwrap: Efficient sql query processing in differentially private data federations. *Proceedings of the VLDB Endowment*, 12(3):307–320, November 2018.

Renato Bruni, Francesco Cesarone, Andrea Scozzari, and Fabio Tardella. Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models. *Data in brief*, 8:858–862, 2016.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

William Lee Croft, Jörg-Rüdiger Sack, and Wei Shi. Differential privacy via a truncated and normalized Laplace mechanism. *arXiv preprint arXiv:1911.00602*, 2019.

Rachel Cummings, Michael Kearns, Aaron Roth, and Zhiwei Steven Wu. Privacy and truthful equilibrium selection for aggregative games. In *International Conference on Web and Internet Economics (WINE)*, 2015.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference (TCC)*, pages 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Quan Geng, Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The staircase mechanism in differential privacy. *IEEE Journal of Selected Topics in Signal Processing*, 9:1176–1184, 2015.

Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. Tight analysis of privacy and utility tradeoff in approximate differential privacy. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1106–1125. Society for Industrial and Applied Mathematics, 2010.

Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pól Mac Aonghusa. The bounded Laplace mechanism in differential privacy. *Journal of Privacy and Confidentiality*, 10(1), 2020.

Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan Ullman. Privately solving linear programs. In *Proceedings of the International Colloquium on*

Automata, Languages and Programming (ICALP), pages 612–624, 2014.

Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. *SIAM Journal on Computing*, 45 (6):1953–1984, 2016.

Daniel Kifer, Adam D. Smith, and Abhradeep Thakurta. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 25.1–25.40, 2012.

Wu Li. The sharp Lipschitz constants for feasible and optimal solutions of a perturbed linear program. *Linear algebra and its applications*, 187:15–40, 1993.

Olvi L Mangasarian. A condition number of linear inequalities and equalities. *Methods of Operations Research*, 43:3–15, 1981.

Olvi L Mangasarian and T-H Shiau. Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems. *SIAM Journal on Control and Optimization*, 25(3):583–595, 1987.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2007.

Yosef Rinott, Christine M O'Keefe, Natalie Shlomo, Chris Skinner, et al. Confidentiality and differential privacy in the dissemination of frequency tables. *Statistical Science*, 33(3):358–385, 2018.

Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, July 2012.