

Privately Solving Linear Programs

Justin Hsu* Aaron Roth† Tim Roughgarden‡ Jonathan Ullman§

May 9, 2014

Abstract

In this paper, we initiate the systematic study of solving linear programs under differential privacy. The first step is simply to define the problem: to this end, we introduce several natural classes of *private linear programs* that capture different ways sensitive data can be incorporated into a linear program. For each class of linear programs we give an efficient, differentially private solver based on the multiplicative weights framework, or we give an impossibility result.

arXiv:1402.3631v2 [cs.DS] 8 May 2014

*Department of Computer and Information Science, University of Pennsylvania. Supported in part by NSF Grant CNS-1065060.

†Department of Computer and Information Science, University of Pennsylvania. Supported in part by an NSF CAREER award, NSF Grants CCF-1101389 and CNS-1065060, and a Google Focused Research Award. Email: aaroth@cis.upenn.edu.

‡Department of Computer Science, Stanford University. Supported in part by NSF Awards CCF-1016885 and CCF-1215965, and an ONR PECASE Award.

§School of Engineering and Applied Sciences and Center for Research on Computation and Society, Harvard University. Supported by NSF grant CNS-1237235. Email: jullman@seas.harvard.edu.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 1.1 | Our Results and Techniques | 3 |
| 1.2 | Related Work | 5 |
| 2 | Differential Privacy Preliminaries | 6 |
| 3 | Constraint Private LPs | 8 |
| 3.1 | Solving LPs with Dense Multiplicative Weights | 9 |
| 3.2 | Achieving Constraint Privacy | 11 |
| 3.3 | Private Fractional Set Cover | 13 |
| 4 | Low-Sensitivity LPs | 15 |
| 4.1 | Solving LPs with Multiplicative Weights | 15 |
| 4.2 | Scalar-Private LPs | 16 |
| 4.3 | Row/Matrix-Private LPs | 18 |
| 4.4 | Column Private LPs | 21 |
| 4.5 | Objective Private LPs | 23 |
| 5 | Lower Bounds | 24 |
| 5.1 | High-Sensitivity Scalars | 25 |
| 5.2 | High-Sensitivity Objective | 26 |
| 5.3 | High-Sensitivity Constraints/Columns | 26 |
| 6 | Discussion | 27 |

1 Introduction

Linear programming is one of the most fundamental and powerful tools in algorithmic design. It is used ubiquitously throughout computer science: applications include maximum matching, maximum and minimum cost flow, and fractional packing and covering problems. Linear programming relaxations of NP-complete problems also underlie countless efficient approximation algorithms.

At the same time, differential privacy is a field where efficient algorithms have been difficult to find. For many problems in differential privacy, the initial focus was on understanding the *information-theoretic complexity*—the extent to which solving the problem, efficiently or not, is compatible with differential privacy. As a result, there are many central problems that are known to be privately solvable, but for which computationally efficient algorithms are not known. For example, Kasiviswanathan et al. [19] show how to privately PAC learn any PAC learnable concept class (without privacy) with only a small increase in the sample complexity, but via an exponential time algorithm. It remains open whether a computationally efficient algorithm can do this in general. Similarly, Blum et al. [4] show how to privately release a summary of a private database that approximately preserves the answers to rich families of linear queries, again via an exponential time algorithm. In fact, under standard cryptographic assumptions, it is not possible to efficiently and privately answer large collections of general linear queries [10, 28, 27].

The two preceding examples are among the many algorithms that use the extremely general *exponential mechanism* of McSherry and Talwar [23] to achieve near optimal error. However, the exponential mechanism is not efficient in general: it requires running time linear in the size of its output range, which can be extremely large. In contrast, general tools for designing *efficient* differentially private algorithms are harder to come by (although not non-existent, e.g., the sample and aggregate framework [24] and output/objective perturbation for unconstrained convex optimization [6, 21]).

Our work contributes to the toolbox of general algorithmic techniques for designing computationally efficient and differentially private algorithms; specifically, we give tools to privately and efficiently solve linear programs (LPs) of various types. An initial problem is to simply *define* what it means to solve a linear program privately. Differential privacy is defined in terms of *neighboring databases*. A database is a collection of records from some domain and two databases are neighboring if they differ in a single record. Differential privacy requires the output distribution of an algorithm to be nearly identical when run on either of a pair of neighboring databases. If linear programs can depend on private databases, we naturally have a notion of *neighboring linear programs*, and we want an algorithm for solving these linear programs that is differentially private with respect to this notion of neighboring inputs.

The way in which the linear program is derived from the database gives rise to several distinct notions of neighboring linear programs. For instance, consider an LP with objective $c^\top x$ and constraints $Ax \leq b$, where moving to a neighboring LP neighboring database leaves c and A unchanged but perturbs b by only a small amount in each coordinate. Solving this kind of linear programming privately is similar to the well-studied *linear query release* problem in differential privacy, and techniques for linear query release—such as the private multiplicative weights algorithm of Hardt and Rothblum [15] (and its offline variants [14, 16])—can be adapted with minor changes. (This result may even be considered folklore.) On the other hand, the situation is qualitatively different if moving to a neighboring LP can change either the constraint matrix A or the objective vector c . Some of these private LPs can still be solved; others are provably impossible to solve to nontrivial accuracy under differential privacy.

In this paper, we develop a taxonomy of private LPs. For each class, we either present an efficient and accurate differentially private solver, or prove that general LPs of this type cannot be accurately solved while preserving privacy.

1.1 Our Results and Techniques

We consider linear programs $LP(D)$ defined by a database D , with form

$$\begin{aligned} & \max_{x \in \mathbb{R}_+^d} c^\top x \\ & \text{s.t. } Ax \leq b. \end{aligned}$$

Here, the vector x represents the variables of the linear program, and $c = c(D)$, $A = A(D)$, and $b = b(D)$ may each depend on the private database D . Our goal is to find an approximate solution to $LP(D)$ in a sense to be defined, while ensuring differential privacy for the underlying database D .

We classify private LPs along two dimensions: *which part* of the LP depends on the database and *how sensitive* the LP is to changes in the database. Along the second axis, we will consider: 1)

low-sensitivity LPs, where changing one record of the database induces a small difference between coefficients that vanishes as the size of the database n grows and 2) *high-sensitivity* LPs, where changing one record of the database can induce a potentially large change in some coefficient. Low-sensitivity LPs are natural when the coefficients of the LP represent some kind of average over the database, whereas high-sensitivity LPs are natural when the coefficients represent specific records of the database.

Furthermore, we consider four parts of the LP that might depend on the database: 1) *the rows of A* , 2) *the scalars b* , 3) *the columns of A* , and 4) *the objective c* . These four parts of the LP, combined with the two notions of sensitivity, lead to the following eight notions of private linear programming:

1. **The constraints:** For these linear programs, moving to a neighboring database can affect at most one row of A and the corresponding entry of b , which corresponds to changing one constraint of the LP.
 - (a) **High-sensitivity:** For *high-sensitivity constraint private LPs* (Section 3), moving to a neighboring database can change a single constraint arbitrarily. That is, for every pair of neighboring databases D, D' , there exists a row i such that for every row $j \neq i$, $A(D)_j = A(D')_j$ and $b(D)_j = b(D')_j$. This kind of linear program arises, for example, in covering LPs in which each record of the database represents an individual that needs to be covered. We cannot hope to approximately satisfy *every* constraint while ensuring privacy,¹ but we show that by using a variant of multiplicative weights that operates only over a restricted set of distributions, we can still find solutions to such LPs that approximately satisfy *most* of the constraints. As an example of our technique, we solve a private version of the fractional set cover problem.
 - (b) **Low-sensitivity:** For *low-sensitivity constraint private LPs* (Section 4.3), moving to a neighboring database can change a single row of A by a small amount in each entry—for some row i , $\|A_i(D) - A_i(D')\|_\infty \leq 1/n$. We show how to solve these LPs using multiplicative weights; our techniques work equally well if the entire constraint matrix can change on neighboring problems (we will sometimes call these *low-sensitivity matrix* or *row private LPs*).
2. **The scalars:** For these linear programs, c and A are fixed and moving to a neighboring database only affects $b = b(D)$.
 - (a) **High-sensitivity:** For *high-sensitivity scalar private LPs*, for every neighboring D, D' , there is a row i such that for every $j \neq i$, $b(D)_j = b(D')_j$. We show that in general, such LPs cannot be solved privately.
 - (b) **Low-sensitivity:** For *low-sensitivity scalar private LPs*, moving to a neighboring database can change every entry in b slightly, such that $\|b(D) - b(D')\|_\infty \leq 1/n$. These LPs capture the *private linear query release* problem, so we will sometimes refer to them as *query release LPs*. In this problem, the database is viewed as a histogram $D \in \mathbb{N}_+^d$ and

¹For example, given a solution x to $LP(D)$, we can always derive a new constraint (A_i, b_i) that is far from being satisfied by x . If we introduce this new constraint in a neighboring linear program $LP(D')$, by the differential privacy condition, this new constraint must also be far from being satisfied in $LP(D')$ with high probability.

the objective is to find a *synthetic database* $x \in \mathbb{R}_+^d$ such that for every linear query q in some family, $\langle q, x \rangle \approx \langle q, D \rangle$. We show how to adapt existing techniques for this problem and derive resulting accurate solvers for LPs of this form (Section 4.2).

3. **One column in A :** For these linear programs, moving to a neighboring database can affect at most one column of A .

These LPs arise literally as the dual linear programs of row private LPs. For example, in a LP where variables represent different tasks, the private coefficients corresponding to a single variable may represent the amount of resources needed for that task. Then, a packing LP seeks to maximize some objective subject to resource constraints.

- (a) **High-sensitivity:** For *high-sensitivity column private LPs*, for every neighboring D, D' , the matrices $A(D), A(D')$ are arbitrarily different in a single column, and identical in all other columns. We show that in general, such LPs cannot be solved privately (Section 5.3).
 - (b) **Low-sensitivity:** For *low-sensitivity column private LPs*, moving to a neighboring database can change every entry in a single column of A by a small amount. More generally, if A_i is the i th row of A , then $\|A(D)_i - A(D')_i\|_1 \leq 1/n$ for each i . We show how to use these LPs using multiplicative weights (Section 4.4).
4. **The objective:** For these linear programs, moving to a neighboring database can affect the objective c . The scalars b and constraints A remain unchanged.
 - (a) **High-sensitivity:** For *high-sensitivity objective private LPs*, for every neighboring D, D' , a single entry of the objective $c(D), c(D')$ can change arbitrarily. We show that in general, such LPs cannot be solved privately (Section 5.2).
 - (b) **Low-sensitivity:** For *low-sensitivity objective private LPs*, for every neighboring D, D' , the objective vectors $c(D), c(D')$ satisfy $\|c(D) - c(D')\|_1 \leq 1/n$. This kind of linear program can be solved inefficiently to high accuracy by selecting from the set of vertices of the feasible polytope with the exponential mechanism; we show that linear programs in this class can also be solved efficiently and accurately, by directly using randomized response (Section 4.5).

This taxonomy is summarized in Table 1. We will formally define accuracy, but roughly speaking, an accurate solution satisfies each constraint to within additive α , and has objective within additive α of optimal (when there is an objective). The exception is constraint privacy (indicated by the asterisk), where our algorithm finds a solution that satisfies only *most* of the constraints to within additive α , and may violate the other constraints arbitrarily.

1.2 Related Work

Differential privacy emerged from a line of work initiated by Dinur and Nissim [7], was defined by Dwork et al. [9], and is now a standard definition of privacy in computer science. Below, we discuss relevant results in differential privacy; the survey by Dwork [8] is an excellent source for a more comprehensive overview.

| Location of change | High sensitivity | Low sensitivity |
|--------------------|------------------|-----------------------------|
| Objective c | No (Section 5) | Yes (Section 4.5) |
| Scalar b | No (Section 5) | Yes (Folklore, Section 4.2) |
| Row/All of A | Yes* (Section 3) | Yes (Section 4.4) |
| Column of A | No (Section 5) | Yes (Section 4.4) |

Table 1: Efficient and accurate solvability

Private optimization has been studied since the work of Blum et al. [3] and Kasiviswanathan et al. [19], who considered how to choose an optimal classifier privately. Blum et al. [3] give an efficient reduction from SQ learning to private SQ learning, and Kasiviswanathan et al. [19] give a very general but inefficient reduction from PAC learning to private PAC learning using the exponential mechanism of McSherry and Talwar [23]. Private learning was placed explicitly into an optimization framework by Chaudhuri et al. [6], who give two techniques for privately solving certain *unconstrained* convex optimization problems. Gupta et al. [12] give several algorithms for problems in private *combinatorial optimization*, but these were specialized combinatorial algorithms for specific problems.

In parallel, a line of work initiated by Blum et al. [4] and continuing with Dwork et al. [10], Roth and Roughgarden [26], Dwork et al. [11], Hardt and Rothblum [15], Gupta et al. [14], Hardt et al. [16] study the problem of privately producing *synthetic data* consistent with some private database on many *linear queries*. (Of particular note is the private multiplicative weights mechanism of Hardt and Rothblum [15], which achieves the optimal accuracy and running time bounds [27, 5].) This problem can be represented as a linear program with queries defining constraints, and indeed, the private multiplicative weights algorithm of Hardt and Rothblum [15] can be directly applied to solve this kind of linear program. This observation motivates our current investigation.

Our algorithms are mostly based on different variants of the *multiplicative weights* method of solving linear programs, which was introduced by Plotkin et al. [25] (see the excellent survey by Arora et al. [2] for more details). Whereas Plotkin et al. [25] maintain a distribution over the dual variables with multiplicative weights, depending on the kind of linear program we are solving, we either maintain a distribution over the dual variables or the primal variables. To solve *constraint private* LPs, we use a combination of the multiplicative weights update method and *Bregman projections* [2]—Hsu et al. [18] use a similar version of this technique in designing *analyst private* mechanisms.

2 Differential Privacy Preliminaries

Differential privacy is a strong notion of privacy, first introduced by Dwork et al. [9]. In the typical setting, we consider a *database* as a multiset of records, each belonging to a single individual. Then, a randomized function from databases to an output range satisfies differential privacy if, for any change in a single record of the input database, the distribution on outputs remains roughly the same. More formally, we have the following definition.

Definition 1 (Dwork et al. [9]). Let $\epsilon > 0$ and $0 \leq \delta < 1$ be given. A randomized function $M : \mathcal{D} \rightarrow \mathcal{R}$ mapping databases to an output range is (ϵ, δ) -*differentially private* if for every subset

$S \subseteq \mathcal{R}$ and for every pair of database D, D' that differ in a single record,

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] + \delta.$$

When $\delta = 0$, we will say that M is ϵ -differentially private.

We will use two basic mechanisms from differential privacy: the Laplace mechanism and the exponential mechanism. The Laplace mechanism privately releases a number by adding noise drawn from the Laplace distribution.

Definition 2 (Dwork et al. [9]). Let $\epsilon > 0$ be given. A function $f : \mathcal{D} \rightarrow \mathbb{R}$ is Δ -sensitive if for every pair of database D, D' that differ in a single record,

$$|f(D) - f(D')| \leq \Delta.$$

The *Laplace mechanism* applied to a Δ -sensitive function releases

$$f(D) + \nu,$$

where ν is a draw from the *Laplace distribution* with parameter ϵ/Δ ; that is, with probability density function

$$F(\nu) = \frac{\epsilon}{2\Delta} \exp\left(-\frac{|\nu|\epsilon}{\Delta}\right).$$

The Laplace mechanism is ϵ -differentially private and satisfies the following tail bound, which is also an accuracy guarantee for the Laplace mechanism.

Lemma 3. Let $\beta \in (0, 1)$ be given, and let ν be drawn from the Laplace distribution with scale b . Then,

$$\Pr[|\nu| \geq T] \leq \beta \quad \text{for } T = \frac{1}{b} \log(1/\beta).$$

We will also use the exponential mechanism [23], which can privately produce a non-numeric or discrete output. The exponential mechanism is defined in terms of a *quality score* that maps a database and an element of the range to a real valued score. For a given a database, the exponential mechanism privately outputs an element of the range that approximately maximizes the quality score.

Definition 4 (McSherry and Talwar [23]). Let $\epsilon > 0$ be given, and suppose the *quality score* $Q : \mathcal{R} \times \mathcal{D} \rightarrow \mathbb{R}$ is Δ -sensitive in the database. On database D , the ϵ -private exponential mechanism with quality score Q outputs $r \in \mathcal{R}$ with probability proportional to

$$\exp\left(\frac{\epsilon}{2\Delta} \cdot Q(r, D)\right).$$

The exponential mechanism is ϵ -differentially private, and satisfies the following accuracy guarantee.

Theorem 5 (McSherry and Talwar [23]). Let $\beta \in (0, 1)$ and the database D be given. Suppose that the maximum value of the quality score Q on database D is OPT . Then, the ϵ -private exponential mechanism with quality score Q on D outputs $r \in \mathcal{R}$ such that

$$\Pr\left[Q(r, D) \geq \text{OPT} - \frac{2\Delta}{\epsilon} \log\left(\frac{|\mathcal{R}|}{\beta}\right)\right] \geq 1 - \beta.$$

To combine these mechanisms, we will use standard composition theorems.

Theorem 6 (Dwork et al. [11]). *For any $\delta \in (0, 1)$, the composition of k (adaptively chosen) ϵ' -private mechanisms is (ϵ, δ) -differentially private, for*

$$\epsilon' = \frac{\epsilon}{\sqrt{8k \log(1/\delta)}}.$$

3 Constraint Private LPs

Let us begin by considering *constraint private LPs*, with the general form

$$\begin{aligned} & \max_{x \in \mathcal{K}} c^\top x \\ & \text{s.t. } Ax \leq b, \end{aligned}$$

where $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^d$, and $\mathcal{K} \subseteq \mathbb{R}^d$. We think of \mathcal{K} as the *easy* constraints, those that are independent of the database, like non-negativity.

Let $\mathcal{K}_{\text{OPT}} = \mathcal{K} \cap \{x \in \mathbb{R}^d \mid c^\top x = \text{OPT}\}$. Then, the original LP can be solved approximately by repeatedly solving the feasibility problem

$$\begin{aligned} & \text{find } x \in \mathcal{K}_{\text{OPT}} \\ & \text{s.t. } Ax \leq b, \end{aligned}$$

binary searching on the optimal objective value OPT .² Thus, unless we specify otherwise, we will restrict our attention to feasibility LPs. Furthermore, since a linear program has a convex feasible region, \mathcal{K} (and hence \mathcal{K}_{OPT}) are convex. From now on, we will write \mathcal{K} for \mathcal{K}_{OPT} .

Roughly, a private database D defines a linear program with objective vector $c(D)$, constraint matrix $A(D)$, and vector $b(D)$, which we will call the *scalars*. In a constraint private LP, the objective $c(D) = c(D')$ is independent of the data and for every two neighboring datasets D, D' , the matrices $A(D)$ and $A(D')$ are exactly the same, except one matrix has an additional row that the other does not. The vectors $b(D)$ and $b(D')$ are also identical, except there is an entry corresponding to the different constraint in A in one b , but not the other. Then, we want a LP solver that satisfies the differential privacy guarantee with respect to this notion of adjacency. Formally:

Definition 7. A randomized algorithm \mathcal{M} with inputs $m \in \mathbb{N}$, vector $b \in \mathbb{R}^m$, and matrix $A \in \mathbb{R}^{m \times d}$ and outputting a vector in \mathbb{R}^d is (ϵ, δ) -*high sensitivity constraint private* if for any A, A' such that A' is equal to A with an additional row appended, and b, b' such that b' is equal to b with an additional entry,

$$\Pr[\mathcal{M}(m, b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(m + 1, b', A') \in S] + \delta$$

for any set $S \subseteq \mathbb{R}^d$.

²Binary search will incur an additional overhead in privacy, but in some situations may not be necessary: for instance, if a bound on the sensitivity of the optimal objective is known, we can solve the LP non-privately and estimate OPT with the Laplace mechanism.

3.1 Solving LPs with Dense Multiplicative Weights

A standard approach to solving LPs is via *no-regret* algorithms. For a brief summary, these algorithms operate over a series of timesteps, selecting a single action at each step. Once the action is selected, the loss for each action is revealed (perhaps adversarially); the no-regret algorithm then adjusts to favor actions with less loss. While LPs can be solved using any no-regret algorithm, for concreteness we use the multiplicative weights update algorithm.

Throughout, we will use calligraphic letters (\mathcal{A}) to denote sets of actions, Roman letters (A) to denote measures on those actions $A : \mathcal{A} \rightarrow [0, 1]$, and letters with tildes (\tilde{A}) to denote a probability distributions over actions. We will write $|A|$ to mean the *density* of measure A , defined to be $\sum_{a \in \mathcal{A}} A_a$.

We will use a variant of the standard multiplicative weights algorithm that maintains a *dense distribution* over the set of constraints, i.e., a distribution that doesn't place too much probability on any action. We will call this algorithm, due to Herbster and Warmuth [17], the *dense multiplicative weights algorithm* (Algorithm 1). Roughly, the algorithm projects the MW distribution on actions into the set of dense distributions at each step. The loss at each step will be defined by a point that approximately satisfies the average constraint weighted by the MW distribution—by capping the probability on any constraint, we ensure that this point can be selected privately even when a single constraint can change arbitrarily on neighboring instances.

We first define this projection step, also known as a *Bregman projection*.

Definition 8. Let $s > 0$. Given a measure A such that $|A| \leq s$, let $\Gamma_s A$ be the (*Bregman projection of A into the set of $1/s$ -dense distributions*), defined by $\Gamma_s A_a = \frac{1}{s} \cdot \min\{1, cA_a\}$ for every $a \in \mathcal{A}$, where $c \geq 0$ is such that $s = \sum_{a \in \mathcal{A}} \min\{1, cA_a\}$.

Then, we can define the Dense Multiplicative Weights algorithm, which uses the standard multiplicative weights update rule combined with a Bregman projection into the set of dense distributions after each step.

Algorithm 1 The Dense Multiplicative Weights algorithm, $DMW_{s,\eta}$

Let A_1 be the uniform measure on \mathcal{A}
 For $t = 1, 2, \dots, T$:
 Let $\tilde{B}^t = \Gamma_s A^t$
 Receive loss vector ℓ^t (may depend on B^1, \dots, B^t)
Update: For each $a \in \mathcal{A}$:
 Update $A_a^{t+1} = e^{-\eta \ell_a^t} A_a^t$

Then, dense multiplicative weights satisfies the following (regret) guarantee.³

Theorem 9 (Herbster and Warmuth [17]). *Let A_1 be the uniform measure of density 1 and let $\{\tilde{B}^t\}$ be the sequence of projected distributions obtained by $DMW_{s,\eta}$ with arbitrary losses $\{\ell^t\}$ satisfying $\|\ell^t\|_\infty \leq 1$ and $\eta \leq 1/2$. Let \tilde{B}^* be the uniform distribution on some subset $S^* \subseteq \mathcal{A}$ of*

³Note that the regret guarantee is only with respect to dense distributions, rather than arbitrary distributions. This is a result of projecting the MW distribution to a dense distribution—the algorithm may not be able to compete with non-dense distributions.

size s . Then,

$$\frac{1}{T} \sum_{i=1}^T \langle \ell^t, \tilde{B}^t \rangle \leq \frac{1}{T} \sum_{i=1}^T \langle \ell^t, \tilde{B}^* \rangle + \eta + \frac{\log |\mathcal{A}|}{\eta T}.$$

Recall we can assume that we know the optimal value OPT , so the objective can be represented as the constraint $c^\top x = \text{OPT}$. Hence, let $\mathcal{K} = \{x \in \mathbb{R}_+^d \mid c^\top x = \text{OPT}\}$ be the public feasible set. We will assume that there is a known, data-independent upper bound ρ such that

$$\rho \geq \max_D \max_{x \in \mathcal{K}} \|A(D)x - b(D)\|_\infty,$$

which we call the *width* of the LP.

We will define our algorithm in terms of an approximate oracle for solving a linear minimization problem. (For a concrete example of such an oracle in the context of fractional set cover, see the next section.)

Definition 10. An (α, β) -approximate, ρ -bounded oracle, given a distribution $y \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times d}$, with probability at least $1 - \beta$ finds $x^* \in \mathbb{R}^d$ with

$$\sum_{i=1}^m y_i(A_i \cdot x^*) \leq \min_{x \in \mathcal{K}} \sum_{i=1}^m y_i(A_i \cdot x) + \alpha$$

and $\|Ax^* - b\|_\infty \leq \rho$.

To solve linear programs, we use the dense multiplicative weights algorithm to maintain a distribution over the constraints, and pick points $x^t \in \mathcal{K}$ that best satisfy the weighted combination of constraints at each step. Intuitively, the losses will lead to more weight on violated constraints, leading to points that are more feasible. Taking the average of the points x^t will yield an approximately feasible point, if it exists. See Algorithm 2 for the full algorithm.

We note that similar techniques for solving linear programs using multiplicative weights have been known since at least Plotkin et al. [25]; the novelty in our approach is that we use multiplicative weights paired with a projection onto the set of dense distributions, and show that the solution approximately satisfies *most* of the constraints. As we will see, the projection step is needed for privacy.

Theorem 11. Let $0 < \alpha \leq 9\rho$, and let $\beta \in (0, 1)$. Suppose there is a feasible solution of the linear program. Then with probability at least $1 - \beta$, Algorithm 2 with density parameter s run with an $(\alpha/3, \beta/T)$ -approximate, ρ -bounded oracle finds a point x^* in \mathcal{K} such that there is a set of constraints S of size at most $|S| < s$, with $A_i x^* \leq b_i + \alpha$ for every $i \notin S$.

Proof. By a union bound over T steps, the oracle succeeds on all steps with probability at least $1 - \beta$; condition on this event.

Let $\mathcal{K}_s = \{y \in \mathbb{R}^m \mid \mathbf{1}^\top y, \|y\|_\infty \leq 1/s\}$ be the set of $1/s$ -dense distributions. Then, $y^\top Ax^* \leq y^\top b$ for any $y \in \mathcal{K}_s$, so in particular the oracle finds x^t with $y^\top Ax^t < y^\top b + \alpha/3$.

Thus, the loss vectors $\ell^t = (1/2\rho)(b - Ax^t) + 1/2$ satisfy $\ell^t \cdot y^t \geq 1/2 - \alpha/6\rho$, which is at least -1 if $\alpha \leq 9\rho$. Since the oracle is ρ -bounded, $\ell^t \cdot y^t \leq 1$. So, Theorem 9 applies; for p any point in

Algorithm 2 Solving for LP feasibility with dense multiplicative weights

Input $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$.

Let \tilde{y}^1 be the uniform distribution in \mathbb{R}^m , $\rho \geq \max_{x \in \mathcal{K}} \|Ax - b\|_\infty$ be the *width* of the LP, $s \in \mathbb{N}$ be the *density parameter*, and $\alpha > 0$ be the desired accuracy. Let *Oracle* be an (α, β) -accurate, ρ -bounded oracle, and set

$$\eta = \sqrt{\frac{\log m}{T}}, \quad T = \frac{36\rho^2 \log m}{\alpha^2}.$$

For $t = 1, \dots, T$:

Find $x^t = \text{Oracle}(\tilde{y}^t, A)$

Compute losses $\ell_i^t := (1/2\rho)(b_i - A_i \cdot x^t) + 1/2$.

Update \tilde{y}^{t+1} from \tilde{y}^t and ℓ^t via dense multiplicative weights with density s .

Output $\bar{x} = (1/T) \sum_{t=1}^T x^t$.

\mathcal{K}_s , we have the following bound:

$$\begin{aligned} \frac{1}{2} - \frac{\alpha}{6\rho} &\leq \frac{1}{T} \sum_{t=1}^T (\ell^t \cdot p) + \eta + \frac{\log m}{\eta T} \\ &= \frac{1}{T} \sum_{t=1}^T \left(\frac{1}{2\rho} (b - Ax^t) + \frac{1}{2} \right) \cdot p + \eta + \frac{\log m}{\eta T} \end{aligned}$$

Thus,

$$-\frac{\alpha}{6\rho} \leq \frac{1}{T} \sum_{t=1}^T \frac{1}{2\rho} (b - Ax^t) \cdot p + \eta + \frac{\log m}{\eta T}.$$

Define $x = (1/T) \sum_{t=1}^T x^t$, and rearrange:

$$p^\top Ax \leq p^\top b + 2\rho\eta + \frac{2\rho \log m}{\eta T} + \frac{\alpha}{3}.$$

By our choice of η and T , we get

$$p^\top Ax \leq p^\top b + \alpha.$$

Since this holds for any $p \in \mathcal{K}_s$, x satisfies all but $s - 1$ constraints with error α —if it didn't, letting p be the uniform distribution on the s violated constraints would give a contradiction. \square

3.2 Achieving Constraint Privacy

Now, we will see how to make Algorithm 2 constraint private. First, the output point depends on the private data (the constraints A) only through the minimization step. Thus, if we can make the minimization private (in a certain sense), then each x^t (and hence the final point \bar{x}) will satisfy constraint privacy. Note that if the oracle privately minimizes over \mathcal{K} , the final point \bar{x} will automatically be in \mathcal{K} since \mathcal{K} is convex. Hence, we can also think of \mathcal{K} as the *public* constraints, the ones that are always satisfied.

Theorem 12. Let $\epsilon, \delta, T > 0$, and let

$$\epsilon' = \frac{\epsilon}{\sqrt{8T \log(1/\delta)}}.$$

with density parameter $s \in \mathbb{N}$. Suppose the oracle is ϵ' -private, where on neighboring instances the inputs (distributions) \tilde{y}, \tilde{y}' satisfy

$$\|\tilde{y}\|_\infty \leq 1/s, \quad \|\tilde{y}'\|_\infty \leq 1/s, \quad \|\tilde{y} - \tilde{y}'\|_1 \leq 2/s,$$

and the matrices A, A' are exactly the same except one has an additional row, and the vectors b, b' except one has a corresponding additional entry. Then, Algorithm 2 with density s is (ϵ, δ) -high sensitivity constraint private.

Proof. If the oracle is ϵ' -differentially private, then (ϵ, δ) -constraint privacy for the whole algorithm follows directly by composition (Theorem 6).

To show that the oracle is private when adding or removing a constraint from the LP, we know that A, A' are exactly the same except one has an extra row, and we know that $\|\tilde{y}\|_\infty \leq 1/s$ since we have projected into the set \mathcal{K}_s . Hence, it only remains to check that neighboring \tilde{y}, \tilde{y}' satisfy $\|\tilde{y} - \tilde{y}'\|_1 \leq 2/s$ for each timestep t . We use a result about the sensitivity of Bregman projections from Hsu et al. [18]; we reproduce the proof for completeness.

Lemma 13 (Hsu et al. [18]). Let $s > 0$ be given. Suppose A, A' be measures on sets $\mathcal{A}, \mathcal{A} \cup a'$ respectively, and identical on \mathcal{A} . If \tilde{A}, \tilde{A}' are the respective Bregman projections into the set of $1/s$ -dense distributions, then

$$\|\tilde{A} - \tilde{A}'\|_1 \leq 2/s.$$

Here and below, we treat A, A' as supported on the same set with $A_{a'}$ fixed at 0.

Proof. From the definition of the projection (Definition 8), it's clear that $s\tilde{A}_a \geq s\tilde{A}'_a$ for all $a \neq a'$. We then have the following:

$$\begin{aligned} \sum_{a \in \mathcal{A} \cup a'} |s\tilde{A}_a - s\tilde{A}'_a| &= |s\tilde{A}_{a'} - s\tilde{A}'_{a'}| + \sum_{a \neq a'} |s\tilde{A}_a - s\tilde{A}'_a| \\ &\leq 1 + \sum_{a \neq a'} |s\tilde{A}_a - s\tilde{A}'_a| \\ &= 1 + \sum_{a \neq a'} s\tilde{A}_a - s\tilde{A}'_a \\ &= 1 + s|\tilde{A}'| - s(|\tilde{A}'| - \tilde{A}'_{a'}) \\ &\leq 1 + s - (s - 1) = 2 \end{aligned}$$

Dividing through by s , we are done. □

Since y, y' are identical except for the weight corresponding to the differing constraint, we are done by the lemma. □

Now that we have presented our algorithm for solving LPs under constraint privacy, we give an example of how to instantiate the oracle and apply Theorem 11.

3.3 Private Fractional Set Cover

We will consider the example of the *fractional set cover* LP, though our arguments extend to constraint private LPs with a private oracle that has low width. (For example, many covering and packing LPs satisfy this property.)

Suppose there are d sets, each covering some subset of m people. Each set has a cost c_S , and we wish to select the cheapest collection of sets that covers every person. We will consider the fractional relaxation of this problem, where instead of selecting whole sets for the cover, we can decide to select a fraction of each set, i.e., each set can be chosen to some non-negative degree, and the cost for set S is the degree to which it is open times c_S . We again want the cheapest fractional collection of sets, such that at least weight 1 covers each person.⁴

To formulate this as a linear program, let the variables be $x \in \mathbb{R}_+^d$; variable x_S will be the degree that we choose set S in the cover. For the constraints, let $A_i \in \{0, 1\}^m$ such that A_{iS} is 1 exactly when set S covers i , otherwise 0.

We will assume that the optimal value OPT is known, and the goal is to compute an approximate fractional set covering x^* corresponding to OPT . This is equivalent to solving the following linear program:

$$\begin{aligned} \text{find: } & x \in \mathcal{K} \\ \text{s.t. } & A_i \cdot x \geq 1 \quad \text{for each } i \end{aligned}$$

where $\mathcal{K} = \{x \in \mathbb{R}_+^d \mid c \cdot x = \text{OPT}\}$ is the feasible region.

We wish to achieve constraint privacy: if each individual corresponds to a covering constraint, then we want an approximate solution that hides whether a person i needs to be covered or not. This is not always possible—if each set contains just one person, then the presence of a set in any valid covering will reveal information about the people that need to be covered. Thus, we will find a solution violating a few constraints, so only covering *most* people.

To use our constraint private LP solver, we first define a private oracle solving the minimization problem

$$O(y) = \operatorname{argmin}_{x \in \mathcal{K}} \sum_i y_i (A_i \cdot x).$$

Since the oracle is minimizing a linear function, the optimal point lies at a vertex of \mathcal{K} and is of the form

$$x^* = \frac{\text{OPT}}{c_i} e_i$$

for some i , where e_i is the i 'th standard basis vector, i.e., all zeros except for a 1 in the i 'th coordinate. We can use the exponential mechanism to privately select this vertex.

Lemma 14. *Let $\gamma \in (0, 1)$ be given. Suppose $\|y\|_\infty \leq 1/s$, and suppose that $\|y - y'\|_\infty \leq 2/s$ on adjacent inputs. Let $O(y)$ be the ϵ -private exponential mechanism over the vertices of \mathcal{K} with quality score*

$$Q(j, y) = \sum_i y_i \left(A_i \cdot \frac{\text{OPT}}{c_j} e_j - 1 \right) = \frac{\text{OPT}}{c_j} \sum_i y_i a_{ij} - 1.$$

⁴To highlight the constraint private LP, we will only consider the fractional version. It is also possible to round the fractional solution to an integral solution (with slightly worse cost), since randomized rounding is independent of the private data.

Then O is an (α, γ) -approximate, ρ -bounded oracle, with

$$\rho = \frac{\text{OPT}}{c_{\min}} - 1 \quad \text{and} \quad \alpha = \frac{6 \text{OPT} \log d \log(1/\gamma)}{c_{\min} \cdot s \cdot \epsilon}.$$

Proof. The width of the oracle is clear: when returning a point $x = \frac{\text{OPT}}{c_i} e_i$,

$$A_i \cdot x - 1 \leq \frac{\text{OPT}}{c_{\min}} - 1.$$

For accuracy, note that the quality score Q has sensitivity at most

$$\Delta = \frac{3 \text{OPT}}{c_{\min} \cdot s}.$$

Why? On neighboring databases, there are two possible changes: first we may have $|y_i - y'_i| \leq 2/s$, and second we may have an extra term in the sum on one neighbor (since the sum is taken over all constraints, and one neighboring instance has an extra constraint). The first source contributes sensitivity $2 \text{OPT} / (c_{\min} s)$, and since $\|y\|_\infty, \|y'\|_\infty \leq 1/s$, the second source contributes sensitivity $\text{OPT} / (c_{\min} s)$.

Now, since there are d possible outputs, the accuracy guarantee for the exponential mechanism (Theorem 5) shows that O selects a point with additive error at most

$$\alpha = \frac{2\Delta}{\epsilon} \log d \log(1/\gamma) = \frac{6 \text{OPT}}{c_{\min} s \epsilon} \log d \log(1/\gamma)$$

with probability at least $1 - \gamma$. Hence, we are done. \square

Now, it follows that Algorithm 2 solves the private fractional set cover problem with the following accuracy guarantee.

Theorem 15. *Let $\beta \in (0, 1)$. With probability at least $1 - \beta$, Algorithm 2 with the exponential mechanism as an oracle (Lemma 14)—where ρ is the width of the oracle and $\alpha \leq 9\rho$ —finds a point x^* such that $A_i x^* \geq 1 - \alpha$ except for at most s constraints i , where*

$$s = \tilde{O} \left(\frac{\text{OPT}^2 \log d \log^{1/2} m \log(1/\beta) \log^{1/2}(1/\delta)}{c^2 \cdot \alpha^2 \cdot \epsilon} \right).$$

Algorithm 2 is also ϵ -high sensitivity constraint private.

Proof. Let ϵ' be as in Theorem 12, and let $\gamma = \beta/T$ with T as in Algorithm 2. Unfolding the definition of ϵ' and ρ and applying Lemma 14, the oracle gives accuracy

$$\frac{6 \text{OPT}}{c_{\min} s \epsilon'} \log d \log(1/\gamma) = \frac{96\sqrt{2} \text{OPT}^2 \log d \log^{1/2} m \log(1/\gamma) \log^{1/2}(1/\delta)}{c_{\min}^2 \epsilon s \alpha}$$

with probability at least $1 - \gamma$. Set this equal to $\alpha/3$. By assumption $\alpha \leq 9\rho$, so Theorem 11 applies: with probability at least $1 - \beta$, there is a set S of at most s constraints such $A_i x^* \geq 1 - \alpha$ for every $i \notin S$, where

$$s = O \left(\frac{\text{OPT}^2 \log d \log^{1/2} m \log(1/\gamma) \log^{1/2}(1/\delta)}{c^2 \cdot \alpha^2 \cdot \epsilon} \right).$$

and $\gamma = \beta/T$. \square

Remark 16. A variant of the efficient private set cover problem has been investigated by Gupta et al. [12]. Our techniques are more general, but the solution we provide here has an incomparable accuracy guarantee. We include this example to demonstrate how to use Algorithm 2 and Theorem 11. On the one hand, we may fail to satisfy some of the coverage constraints, and if we imagine that each uncovered element can be covered at a cost of 1, our approximation guarantee now depends on OPT unlike the guarantee of Gupta et al. [12].

On the other hand, we output an explicit solution whereas the algorithm of Gupta et al. [12] outputs an implicit solution, a “set of instructions” that describes a set cover when paired with the private data. (Their approach can also be interpreted as satisfying the weaker guarantee of *joint differential privacy* [20] rather than standard differential privacy.) Finally, our techniques apply to general constraint-private linear programs, not just set cover.

4 Low-Sensitivity LPs

Let us now turn to low-sensitivity LPs. Recall that for these LPs, the distance between adjacent inputs decreases as the size of the database (i.e., the number of individuals) grows. First, a few simplifying assumptions. Like above, we will continue to solve feasibility LPs of the following form:

$$\begin{aligned} &\text{find } x \in \mathbb{R}_+^d \\ &\text{s.t. } Ax \leq b \end{aligned}$$

Unlike the case for general constraint private LPs, we require that the feasible solution is a distribution, i.e., is non-negative and has ℓ_1 norm 1. Note that if the optimal solution has ℓ_1 norm L , then the rescaled LP

$$\begin{aligned} &\text{find } x \in \mathbb{R}_+^d \\ &\text{s.t. } Ax \leq b/L \end{aligned}$$

has a distribution as a solution. Our algorithms will find a point x^* such that $Ax^* \leq b/L + \alpha \cdot \mathbf{1}$, so if we set $\alpha = \alpha'/L$, then $A(Lx^*) \leq b + \alpha'$ gives an approximate solution to the original, unscaled LP.

4.1 Solving LPs with Multiplicative Weights

Before getting into specific kinds of low-sensitivity LPs, we first review another standard method for solving LPs via the standard multiplicative weights algorithm presented in Algorithm 3.

Algorithm 3 The Multiplicative Weights Algorithm, MW_η

Let \tilde{A}^1 be the uniform distribution on \mathcal{A}
 For $t = 1, 2, \dots, T$:
 Receive loss vector ℓ^t (may depend on A^1, \dots, A^t)
 For each $a \in \mathcal{A}$:
 Update $A_a^{t+1} = e^{-\eta \ell_a^t} \tilde{A}_a^t$ for every $a \in \mathcal{A}$
 Normalize $\tilde{A}^{t+1} = A^{t+1} / |A_{t+1}|$

Unlike the dense multiplicative weights approach presented earlier (Algorithm 2), we use multiplicative weights to maintain a distribution over the *variables* rather than the constraints.⁵ This distribution will be the candidate solution, and we define losses by the maximum constraint violation of this candidate solution at each step. It will be useful to first define an oracle for linear maximizations.

Definition 17. For $\epsilon > 0, \gamma > 0$ an (α, γ) -dual oracle, given A, b, x as input, finds a constraint $i \in [m]$ such that

$$A_i x - b_i \geq \max_j A_j x - b_j - \alpha,$$

with probability at least $1 - \gamma$.

We now give the full algorithm in Algorithm 4.

Algorithm 4 Solving for LP feasibility with primal multiplicative weights

Input $A \in \mathbb{R}^{m \times d}, b \in \mathbb{R}^m$.

Let \tilde{x}^1 be the uniform distribution in \mathbb{R}^d , $\rho = \max_{i,j} |A_{ij}|$ be the *width* of the LP, $\alpha > 0$ be the desired accuracy. Let *Oracle* be a (α, γ) -dual oracle, and set

$$\eta = \sqrt{\frac{\log d}{T}}, \quad T = \frac{9\rho^2 \log d}{\alpha^2}.$$

For $t = 1, \dots, T$:

Find $p^t = \text{Oracle}(A, b, \tilde{x}^t)$

Compute losses $\ell_i^t := (1/\rho)A_{p^t i}$

Update \tilde{x}^{t+1} from \tilde{x}^t and ℓ^t via multiplicative weights.

Output $\bar{x} = (1/T) \sum_{t=1}^T \tilde{x}^t$

Then, the following accuracy guarantee is known.

Theorem 18 (Plotkin et al. [25]). *Suppose there is a feasible distribution solution of the linear program $Ax \leq b$. Then, running Algorithm 4 with an $(\alpha/3, \gamma)$ -dual oracle finds a point x such that $Ax \leq b + \alpha \cdot \mathbf{1}$ with probability at least $1 - T\gamma$.*

4.2 Scalar-Private LPs

First, we consider linear programs where the objective and constraint coefficients are public data, but the right hand side in the constraints may contain private data. Roughly, a private database D maps to an objective vector $c(D)$, a constraint matrix $A(D)$, and a vector $b(D)$. For every pair of neighboring databases D, D' , we have $c(D) = c(D')$ and $A(D) = A(D')$ independent of the data, and $\|b(D) - b(D')\|_\infty \leq \Delta_\infty$. We will think of Δ_∞ as decreasing in n ; our accuracy guarantees will be trivial if this is not true. As usual, we will assume the LP is in feasibility form, and leave the objective c implicit. Formally:

⁵Readers familiar with game theory may notice that we are solving LPs by finding the equilibrium of a two player, zero-sum game. Then, Algorithm 2 is solving the game with MW over the constraints and best response over the variables, while the approach we present in this section swaps the two roles.

Definition 19. A randomized algorithm \mathcal{M} with inputs vector $b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times d}$, and outputting a vector in \mathbb{R}^d is (ϵ, δ) -low sensitivity scalar private with sensitivity Δ_∞ if for any b, b' such that $\|b - b'\|_\infty \leq \Delta_\infty$,

$$\Pr[\mathcal{M}(b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(b', A) \in S] + \delta$$

for any set $S \subseteq \mathbb{R}^d$.

The algorithm we use is a slight generalization of the offline private multiplicative weights algorithm [13, 16] (building on the influential work of Hardt and Rothblum [15], who introduced the “online” variant). In our framework, we will express the algorithm as a differentially private variant of Algorithm 4 to solve these linear programs while preserving differential privacy.

Throughout, we assume that the vector b is private data. On neighboring databases, b can change by at most Δ_∞ in ℓ_∞ norm. Looking at Algorithm 4, we see that the only place we touch the private data is in the dual oracle. Accordingly, if the dual oracle is private in b , then the whole algorithm is private.

Theorem 20. Let ϵ, δ, T be as in Algorithm 4, and let

$$\epsilon' = \frac{\epsilon}{\sqrt{8T \log(1/\delta)}}.$$

Algorithm 4, run with an ϵ' -private dual oracle is (ϵ, δ) -differentially private.

Proof. Direct from composition (Theorem 6). □

Just like in private multiplicative weights for private query release, the exponential mechanism gives an appropriate dual oracle.

Lemma 21. Let $\epsilon, \gamma > 0$ be given, and suppose the vector b can differ by at most Δ_∞ in ℓ_∞ norm on neighboring instances. Then, the ϵ -private exponential mechanism with quality score

$$Q(i, b) = A_i x - b_i$$

is an (α, γ) -dual oracle, for

$$\alpha = \frac{2\Delta_\infty}{\epsilon} \cdot \log\left(\frac{m}{\gamma}\right).$$

Proof. This is ϵ -private by definition, and the accuracy follows from the accuracy of the exponential mechanism (Theorem 5)—the quality score is Δ_∞ -sensitive in b , and the output ranges over the constraints, so has size m . □

Combining the MW with the oracle, our private low-sensitivity scalar-private LP solver Algorithm 4 satisfies the following accuracy guarantee.

Theorem 22. Let $\alpha, \beta \in (0, 1)$ be given. Suppose the linear program $Ax \leq b$ has a distribution as a feasible solution. Algorithm 4, run with the exponential mechanism as a dual oracle (Lemma 21), is (ϵ, δ) -low sensitivity scalar private with sensitivity Δ_∞ , and finds x^* satisfying $Ax^* \leq b + \alpha \cdot \mathbf{1}$, with probability at least $1 - \beta$, where

$$\alpha = \tilde{O}\left(\frac{\rho^{1/2} \Delta_\infty^{1/2}}{\epsilon^{1/2}} \cdot \log^{1/4} d \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m\right).$$

Proof. Let ϵ' be as in Theorem 20, and let $\gamma = \beta/T$ with T from Algorithm 4. By Lemma 21, the ϵ' -private exponential mechanism with quality score

$$Q(i, b) = A_i x - b_i$$

is an $(\alpha/3, \gamma)$ -dual oracle for

$$\alpha = \frac{6\Delta_\infty \sqrt{8T \log(1/\delta)}}{\epsilon} \cdot \log\left(\frac{mT}{\beta}\right) = \frac{18\rho\Delta_\infty \sqrt{8 \log d \log(1/\delta)}}{\alpha\epsilon} \cdot \log\left(\frac{9\rho^2(\log d)m}{\alpha^2\beta}\right).$$

Solving,

$$\alpha = \tilde{O}\left(\frac{\rho^{1/2}\Delta_\infty^{1/2}}{\epsilon^{1/2}} \cdot \log^{1/4} d \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m\right)$$

as desired. \square

Remark 23. This bound generalizes the guarantee for the private multiplicative weights algorithm when privately generating synthetic data for linear queries [15]. In that setting, there is one variable for each element in some underlying data universe \mathcal{X} (and so $d = |\mathcal{X}|$), and there is one equality constraint for each of k linear queries (and so $m = k$).

Now, let us consider the low-sensitivity version of constraint privacy: neighboring instances have constraint matrices that differ to a small degree. We distinguish two further subcases: either every coefficient in each constraint can differ, or only a few coefficients in each constraint can differ.

4.3 Row/Matrix-Private LPs

Suppose we have the feasibility problem

$$\begin{aligned} &\text{find } x \\ &\text{s.t. } Ax \leq b, \end{aligned}$$

where some entries in A may change by at most Δ_∞ on a neighboring instance.

Roughly, a private database D maps to an objective vector $c(D)$, a constraint matrix $A(D)$, and a vector $b(D)$. For every pair of neighboring databases D, D' , we have $c(D) = c(D')$ and $b(D) = b(D')$ independent of the data, and $\|A(D) - A(D')\|_\infty \leq \Delta_\infty$. Again, we will think of Δ_∞ as decreasing in n ; our accuracy guarantees will be trivial if this is not true. Our techniques work equally well whether only a single row of A or the entire matrix A can differ, so we will assume the latter. We will also assume that the LP is in feasibility form, and leave the objective c implicit. Formally:

Definition 24. A randomized algorithm \mathcal{M} with inputs vector $b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times d}$, and outputting a vector in \mathbb{R}^d is (ϵ, δ) -low sensitivity row private with sensitivity Δ_∞ if for any A, A' such that $\|A - A'\|_\infty \leq \Delta_\infty$,

$$\Pr[\mathcal{M}(b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(b', A') \in S] + \delta$$

for any set $S \subseteq \mathbb{R}^d$.

Algorithm 5 Row/matrix private LP solver

Input $A \in [-1, 1]^{m \times d}$, $b \in \mathbb{R}^m$.

Let \tilde{x}^1 be the uniform distribution in \mathbb{R}^d , $\alpha > 0$ be the desired accuracy, and Δ_∞ be the sensitivity.

Let *Oracle* be a (α, γ) -dual oracle, and set

$$T = \frac{144 \log d}{\alpha^2}, \quad \epsilon' = \frac{\epsilon}{4\sqrt{dT \log(1/\delta)}}, \quad \eta = \sqrt{\frac{\log d}{T}}.$$

For $t = 1, \dots, T$:

Find $p^t = \text{Oracle}(A, b, \tilde{x}^t)$.

Compute private losses $\hat{\ell}_i^t := \frac{A_{p^t i} + \text{Lap}(\frac{\Delta_\infty}{\epsilon'})}{2}$.

For each i , update $x_i^{t+1} = e^{-\eta \hat{\ell}_i^t} \cdot \tilde{x}_i^t$.

Normalize $\tilde{x}^{t+1} = x^{t+1} / |x^{t+1}|$.

Output $\bar{x} = (1/T) \sum_{t=1}^T \tilde{x}^t$.

We will normalize the problem so each entry in A is in $[-1, 1]$.

The basic idea is to use multiplicative weights over the primal variables, with a dual oracle selecting the most violated constraint—since the losses fed into the multiplicative weights algorithm now depend on private data (the matrix A), we add Laplace noise to the loss vectors as they are selected. The full algorithm is given in Algorithm 5.

We can now show privacy and accuracy for Algorithm 5.

Theorem 25. *Let $\epsilon, \epsilon', \delta, \Delta_\infty$ be as in Algorithm 5. Algorithm 5 run with an ϵ' -private dual oracle is (ϵ, δ) -low sensitivity row private with sensitivity Δ_∞ .*

Proof. Algorithm 5 performs dT Laplace operations, and T oracle operations. Each operation is ϵ' -private, so this is at most $2dT$ ϵ' -private operations. By our choice of ϵ' and Theorem 6, the whole algorithm is (ϵ, δ) -differentially private. \square

We can show that the exponential mechanism is a private dual oracle.

Lemma 26. *Let $\epsilon, \gamma > 0$ be given, and suppose the matrix A can differ by at most Δ_∞ in ℓ_∞ norm on neighboring instances. Let x be any distribution. Then, the ϵ -private exponential mechanism with quality score*

$$Q(i, b) = A_i x - b_i$$

is an (α, γ) -dual oracle, for

$$\alpha = \frac{2\Delta_\infty}{\epsilon} \cdot \log\left(\frac{m}{\gamma}\right).$$

Proof. Since x is a distribution, the quality score is Δ_∞ -sensitive and accuracy follows from accuracy of the exponential mechanism (Theorem 5). \square

While previously our accuracy theorems followed from standard accuracy results for solving LPs using multiplicative weights, the proof for Algorithm 5 does not. Since the constraint matrix is private, Algorithm 4 perturbs the losses and requires a more custom analysis. So, we will need a standard regret bound for multiplicative weights.

Theorem 27 (Littlestone and Warmuth [22]). Let $\{\tilde{A}^t\}$ be the distributions obtained by MW_η with arbitrary losses $\{\ell^t\}$ satisfying $\|\ell^t\|_\infty \leq 1$. Suppose that $\eta \leq 1/2$. Let $A^* = \mathbf{1}_{a=a^*}$, for some $a^* \in \mathcal{A}$. Then,

$$\sum_{t=1}^T \langle \ell^t, \tilde{A}^t \rangle \leq \sum_{t=1}^T \langle \ell^t, A^* \rangle + \eta + \frac{\log |\mathcal{A}|}{\eta T}.$$

Then, we have the following accuracy guarantee.

Theorem 28. Let $\beta > 0$. Suppose the program has a distribution as a feasible solution. Then, with probability at least $1 - \beta$, Algorithm 5 run with the exponential mechanism as a dual oracle (Lemma 26) finds a solution x^* such that $Ax^* \leq b + \alpha \cdot \mathbf{1}$, where

$$\alpha = \tilde{O} \left(\frac{\Delta_\infty^{1/2} d^{1/4}}{\epsilon^{1/2}} \cdot \text{polylog} \left(d, m, \frac{1}{\beta}, \frac{1}{\delta} \right) \right).$$

Proof. Let ϵ' be as in Theorem 25, T be from Algorithm 5, and $\gamma = \beta/2dT$. By Lemma 26, with probability at least $1/\gamma$, the oracle's choices satisfy

$$(A_i \cdot \tilde{x}^t - b_i) - (A_{p^t} \cdot \tilde{x}^t - b_{p^t}) \leq \frac{2\Delta_\infty}{\epsilon'} \cdot \log \left(\frac{m}{\gamma} \right)$$

for all constraints i . Note that the left hand side is equal to $(A_i \cdot \tilde{x}^t - b_i) - (\ell^t \cdot \tilde{x}^t - b_{p^t})$. Taking a union bound over all T steps, this is true for all p^t with probability at least $1 - \beta/2d \geq 1 - \beta/2$; condition on this event.

By a tail bound on the Laplace mechanism (Lemma 3), with probability at least $1 - \gamma$, a noisy loss $\hat{\ell}_i^t$ satisfies

$$\left| \hat{\ell}_i^t - (1/2)\ell_i^t \right| \leq \frac{\Delta_\infty}{\epsilon'} \cdot \log \left(\frac{1}{\gamma} \right).$$

Taking a union bound over all T steps and d losses, this is true for all losses with probability at least $1 - \beta/2$; condition on this event.

We first show that if these errors are small, then the theorem holds. Assume

$$\frac{2\Delta_\infty}{\epsilon'} \cdot \log \left(\frac{m}{\gamma} \right) \leq \frac{\alpha}{6}, \tag{1}$$

so that both right hand sides above are at most $\alpha/6$. Since $\alpha < 1$, this implies that every Laplace noise is at most 1, so the noisy losses are bounded: $|\hat{\ell}_i^t| \leq 1$.

Let x^* be an exactly feasible point, with ℓ_1 norm 1. By the regret guarantee for multiplicative weights (Theorem 27),

$$\begin{aligned} \frac{1}{T} \sum_t \hat{\ell}^t \cdot \tilde{x}^t &\leq \frac{1}{T} \sum_t \hat{\ell}^t \cdot x^* + \eta + \frac{\log d}{\eta T} \\ \frac{1}{T} \sum_t \left(\frac{A_{p^t}}{2} + \frac{\nu}{2} \right) \cdot \tilde{x}^t - \frac{b_{p^t}}{2} &= \frac{1}{T} \sum_t \hat{\ell}^t \cdot \tilde{x}^t - \frac{b_{p^t}}{2} \leq \frac{1}{T} \sum_t \hat{\ell}^t \cdot x^* - \frac{b_{p^t}}{2} + \eta + \frac{\log d}{\eta T}, \end{aligned}$$

where ν is a vector of independent draws from $\text{Lap}(\Delta_\infty/\epsilon')$. Let i be any constraint. Since we assumed the error of the exponential mechanism to be small (Equation (1)) and \tilde{x}^t is a distribution,

$$\frac{1}{T} \sum_t \left(\frac{1}{2} A_i \cdot \tilde{x}^t - \frac{b_i}{2} \right) + \frac{1}{2} \nu \leq \frac{1}{T} \sum_t \hat{\ell}^t \cdot x^* - \frac{b_{p^t}}{2} + \eta + \frac{\log d}{\eta T} + \frac{\alpha}{6}.$$

By assumption (Equation (1)) $|\nu| \leq \alpha/6$, so

$$\frac{1}{T} \sum_t A_i \cdot \tilde{x}^t - b_i \leq \frac{1}{T} \sum_t 2\hat{\ell}^t \cdot x^* - b_{p^t} + \alpha/3 + \alpha/6 + \eta + \frac{\log d}{\eta T}.$$

Since x^* is a feasible point, $A_i \cdot x^* - b_i \leq 0$ for all i . By assumption (Equation (1)), $|A_i - 2\hat{\ell}_i^t| \leq \alpha/3$. By our choice of η and T ,

$$\begin{aligned} A_i \cdot \bar{x} - b_i &\leq \frac{1}{T} \sum_t A_i \cdot x^* - b_i + \alpha/3 + \alpha/3 + \alpha/6 + \eta + \frac{\log d}{\eta T} \\ &\leq \frac{5\alpha}{6} + \eta + \frac{\log d}{\eta T} \leq \alpha, \end{aligned}$$

as desired.

Now, it only remains to show Equation (1). By unfolding definitions like before, it suffices to take

$$\alpha \geq \frac{12\Delta_\infty^{1/2} d^{1/4} (\log d)^{1/4} (\log(1/\delta))^{1/4}}{\epsilon^{1/2}} \cdot \left(\log \frac{288d \log dm}{\alpha^2 \beta} \right)^{1/2}.$$

□

4.4 Column Private LPs

Rather than an entire row changing, neighboring LPs may differ in a *column*; that is, they differ in coefficients corresponding to a single variable. Roughly, a private database D maps to an objective vector $c(D)$, a constraint matrix $A(D)$, and a vector $b(D)$. For every pair of neighboring databases D, D' , we have $c(D) = c(D')$ and $b(D) = b(D')$ independent of the data, and $\|A(D)_i - A(D')_i\|_\infty \leq \Delta_1$ for every row i of the constraint matrix. Again, we will think of Δ_1 as decreasing in n ; our accuracy guarantees will be trivial if this is not true. Formally:

Definition 29. A randomized algorithm \mathcal{M} with inputs vector $b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times d}$, and outputting a vector in \mathbb{R}^d is (ϵ, δ) -*low sensitivity column private with sensitivity* Δ_1 if for any A, A' such that $\|A_i - A'_i\|_\infty \leq \Delta_1$ for each row $i \in [m]$,

$$\Pr[\mathcal{M}(b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(b', A') \in S] + \delta$$

for any set $S \subseteq \mathbb{R}^d$.

We can use a very slight modification of Algorithm 5 to solve these LPs privately; the algorithm is given in Algorithm 6.

Like before, we can show that the exponential mechanism can be used as a private dual oracle.

Algorithm 6 Column private LP solver

Input $A \in [-1, 1]^{m \times d}$, $b \in \mathbb{R}^m$.

Let \tilde{x}^1 be the uniform distribution in \mathbb{R}^d , $\alpha > 0$ be the desired accuracy, and Δ_1 be the sensitivity.

Let *Oracle* be an (α, γ) -dual oracle, and set

$$T = \frac{144 \log d}{\alpha^2}, \quad \epsilon' = \frac{\epsilon}{4\sqrt{T \log(1/\delta)}}, \quad \eta = \sqrt{\frac{\log d}{T}}.$$

For $t = 1, \dots, T$:

Find $p^t = \text{Oracle}(A, b, \tilde{x}^t)$.

Compute private losses $\hat{\ell}_i^t := \frac{A_{p^t i} + \text{Lap}\left(\frac{\Delta_1}{\epsilon'}\right)}{2}$.

For each i , update $x_i^{t+1} = e^{-\eta \hat{\ell}_i^t} \cdot \tilde{x}_i^t$.

Normalize $\tilde{x}^{t+1} = x^{t+1} / |x^{t+1}|$.

Output $\bar{x} = (1/T) \sum_{t=1}^T \tilde{x}^t$.

Lemma 30. *Let $\epsilon, \gamma > 0$ be given, and suppose neighboring matrices A, A' satisfy $\|A_i - A'_i\|_1 \leq \Delta_1$ for every row i . Let x be any distribution. Then, the ϵ -private exponential mechanism with quality score*

$$Q(i, b) = A_i x - b_i$$

is an (α, γ) -dual oracle, for

$$\alpha = \frac{2\Delta_1}{\epsilon} \cdot \log\left(\frac{m}{\gamma}\right).$$

Proof. Since x is a distribution, the quality score is Δ_1 -sensitive and accuracy follows from accuracy of the exponential mechanism (Theorem 5). \square

Theorem 31. *Let $\epsilon, \epsilon', \delta, \Delta_1$ be as in Algorithm 6. Algorithm 6 run with an ϵ' -private dual oracle is (ϵ, δ) -low sensitivity column private with sensitivity Δ_1 .*

Proof. Since the loss vector ℓ^t can differ by at most Δ_1 in ℓ_1 norm, adding Laplace noise with scale Δ_1/ϵ suffices for ϵ -differential privacy. Thus, there are T Laplace and oracle mechanism steps, each ϵ' -private. By choice of ϵ' and composition, Algorithm 6 is (ϵ, δ) -differentially private. \square

Theorem 32. *Let $\beta > 0$. Suppose the program has a distribution as a feasible solution. Then, with probability at least $1 - \beta$, Algorithm 6 run with the exponential mechanism (Lemma 30) as oracle finds a point x^* such that $Ax^* \leq b + \alpha \cdot \mathbf{1}$, where*

$$\alpha = \tilde{O}\left(\frac{\Delta_1^{1/2}}{\epsilon^{1/2}} \cdot \text{polylog}\left(d, m, \frac{1}{\beta}, \frac{1}{\delta}\right)\right).$$

Proof. Let ϵ' be as in Theorem 31, T be as in Algorithm 6, and $\gamma = \beta/2dT$. Letting the dual oracle be the ϵ' -private exponential mechanism, the proof is nearly identical to Theorem 28. The main difference is that we need

$$\frac{2\Delta_1}{\epsilon'} \cdot \log\left(\frac{1}{\gamma}\right) \leq \frac{\alpha}{6}$$

for everything to go through. By unfolding definitions, it suffices to take

$$\alpha \geq \frac{12\Delta_1^{1/2}(\log d)^{1/4}(\log(1/\delta))^{1/4}}{\epsilon^{1/2}} \cdot \left(\log \frac{288m \log d}{\alpha^2 \beta} \right)^{1/2}.$$

□

Comparing the two previous algorithms, note $\Delta_\infty \leq \Delta_1 \leq d\Delta_\infty$. Algorithm 5 performs better when the right inequality is tighter, i.e., when all the coefficients in a row can differ by a small amount. In contrast, Algorithm 6 performs better when the left inequality is tighter, that is, when a few coefficients in a row can differ by a larger amount.

4.5 Objective Private LPs

For our final type of low-sensitivity LP, we consider linear programs with objectives that depend on private data. We show that a very simple approach—*randomized response*—can solve these types of LPs accurately. Throughout, we will assume that the optimal solution to the LP has ℓ_1 weight equal to 1. We start with an LP in general form:

$$\begin{aligned} & \max c^\top x \\ & \text{s.t. } Ax \leq b, \end{aligned}$$

On instances corresponding to neighboring database D, D' , the objective may change by Δ_1 in ℓ_1 norm: $\|c(D) - c(D')\|_1 \leq \Delta_1$. Formally:

Definition 33. A randomized algorithm \mathcal{M} with inputs vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^d$ and matrix $A \in \mathbb{R}^{m \times d}$, and outputting a vector in \mathbb{R}^d is (ϵ, δ) -*low sensitivity objective private with sensitivity* Δ_1 if for any c, c' such that $\|c - c'\|_1 \leq \Delta_1$,

$$\Pr[\mathcal{M}(c, b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(c', b, A) \in S] + \delta$$

for any set $S \subseteq \mathbb{R}^d$.

For a concrete case, a single objective coefficient may change by Δ_1 . All other parts of the LP do not change: $A(D) = A(D')$, and $b(D) = b(D')$. If we add Laplace noise to the objective and solve the resulting LP, we will get an almost optimal, exactly feasible solution.

Theorem 34. *Suppose an objective private LP has optimal objective OPT, and has optimal solution with ℓ_1 weight 1. Define*

$$\hat{c} = c + \text{Lap} \left(\frac{\Delta_1 \sqrt{8d \log(1/\delta)}}{\epsilon} \right)^d,$$

where the noise is d independent draws from the Laplace distribution with the given parameter. Then, releasing the perturbed LP

$$\begin{aligned} & \max \hat{c}^\top x \\ & \text{s.t. } Ax \leq b \quad \text{and} \quad \mathbf{1}^\top x = 1 \end{aligned}$$

is (ϵ, δ) -*low sensitivity objective private with sensitivity* Δ_1 . With probability $1 - \beta$, solving the perturbed LP non-privately yields a point x^* such that $Ax^* \leq b$ and $c^\top x^* \geq \text{OPT} - \alpha$, where

$$\alpha = \frac{4\Delta_1 \sqrt{8d \log(d/\delta)}}{\epsilon}.$$

Proof. Since the ℓ_1 sensitivity of c is 1 and d numbers are released, (ϵ, δ) -privacy follows from the composition theorem (Theorem 6).⁶

For the accuracy, note that with probability at least $1 - \beta/d$, a single draw of the Laplace distribution is bounded by

$$\frac{\alpha}{2} = \frac{2\Delta_1 \sqrt{8d \log(d/\delta)}}{\epsilon}.$$

By a union bound, this happens with probability at least $1 - \beta$ for all d draws; condition on this event. Then, note that if x^* is the optimal solution to the original LP, then it is also a feasible solution to the perturbed LP. Let \hat{x}^* be the optimal solution of the perturbed LP. Since the noise added to each objective coefficient is bounded by $\alpha/2$, if

$$c^\top \hat{x}^* < \text{OPT} - \alpha$$

then

$$\hat{c}^\top \hat{x}^* < \text{OPT} - \alpha/2 \quad \text{but also} \quad \hat{c}^\top x^* \geq \text{OPT} - \alpha/2,$$

contradicting optimality of \hat{x}^* in the perturbed program. Thus, this algorithm finds an exactly feasible, α -optimal solution. \square

5 Lower Bounds

Now that we have considered various low-sensitivity LPs, let us turn to high-sensitivity LPs. In this section, we show that most high-sensitivity LPs cannot be solved privately to non-trivial accuracy. The exception is constraint private LPs—as we saw (Section 3), these can be solved in a relaxed sense. Our lower bounds are all reductions to reconstruction attacks: as the following theorem shows, differential privacy precludes reconstructing a non-trivial fraction of a database. The idea of reconstruction being a key feature of privacy violation is due to Dinur and Nissim [7]. The following theorem is folklore; we provide a proof for completeness.

Theorem 35. *Let mechanism $\mathcal{M} : \{0, 1\}^n \rightarrow [0, 1]^n$ be (ϵ, δ) -differentially private, and suppose that for all database D , with probability at least $1 - \beta$, $\|\mathcal{M}(D) - D\|_1 \leq \alpha n$. Then,*

$$\alpha \geq \frac{1}{2} - \frac{e^\epsilon + \delta}{2(1 + e^\epsilon)(1 - \beta)} := c(\epsilon, \delta, \beta).$$

The same is true even if D is restricted to have exactly $n/2$ zero entries.

Proof. If we have \mathcal{M} as in the hypothesis, then we can round each entry of $\mathcal{M}(D)$ to $\{0, 1\}$ while preserving (ϵ, δ) -differential privacy. Note that by assumption $\|\mathcal{M}(D) - D\|_1 \leq \alpha n$, so the number of entries $\mathcal{M}(D)_i$ that are more than $1/2$ away from D_i is at most $2\alpha n$. Thus, rounding reconstructs a database in $\{0, 1\}^n$ at most $2\alpha n$ distance from the true database in ℓ_1 norm; hence we may assume that $\mathcal{M}(D) \in \{0, 1\}^n$ with ℓ_1 norm at most $2\alpha n$ from D .

Assume n is even; we prove the case where the input database D has exactly $n/2$ zero entries. Let $D \in \{0, 1\}^n$ have exactly $n/2$ zero entries, and sample an index i such that $D_i = 1$, and an index j such that $D_j = 0$, both uniformly at random from $[n]$. Let D' be identical to D except with bits i and j swapped. By assumption, we have that with probability at least $1 - \beta$

$$\|\mathcal{M}(D) - D\|_1 \leq 2\alpha n \quad \text{and} \quad \|\mathcal{M}(D') - D'\|_1 \leq 2\alpha n.$$

⁶This is similar to the case of privately releasing *histogram queries*.

Since i is chosen uniformly, we also know

$$\Pr[\mathcal{M}(D)_i = D_i] \geq (1 - 2\alpha)(1 - \beta) \quad \text{and} \quad \Pr[\mathcal{M}(D')_i = D'_i] \geq (1 - 2\alpha)(1 - \beta).$$

Hence, $\Pr[\mathcal{M}(D')_i = D_i] \leq 1 - (1 - 2\alpha)(1 - \beta)$ because $D_i \neq D'_i$. By (ϵ, δ) -differential privacy, we get

$$(1 - 2\alpha)(1 - \beta) \leq \Pr[\mathcal{M}(D)_i = D_i] \leq e^\epsilon \Pr[\mathcal{M}(D')_i = D_i] + \delta \leq e^\epsilon(1 - (1 - 2\alpha)(1 - \beta)) + \delta.$$

Finally,

$$1 - 2\alpha \leq \frac{e^\epsilon + \delta}{(1 + e^\epsilon)(1 - \beta)},$$

as desired. \square

For each type of impossible private LP, we show how to convert a database $D \in \{0, 1\}^n$ to a LP, such that neighboring databases D, D' lead to neighboring LPs. We then show that a LP solver that privately solves this LP to non-trivial accuracy leads to a reconstruction attack on D , violating Theorem 35.

First, some notation. For the general LP

$$\begin{aligned} & \max c^\top x \\ \text{s.t. } & Ax \leq b, \end{aligned}$$

we say that x^* is an α -feasible solution if $Ax^* \leq b + \alpha \cdot \mathbf{1}$. Likewise, we say that x^* is an α -optimal solution if it is feasible, and

$$c^\top x^* \geq \max_{x: Ax \leq b} c^\top x - \alpha.$$

5.1 High-Sensitivity Scalars

Consider a database $D \in \{0, 1\}^n$, and the following LP:

$$\begin{aligned} & \text{find } x \\ \text{s.t. } & x_i = D_i \quad \text{for each } i \end{aligned}$$

Note that changing a single bit in D will change a single right hand side in a constraint by 1.

Theorem 36. *Suppose mechanism \mathcal{M} is (ϵ, δ) -high sensitivity scalar private, and with probability at least $1 - \beta$, finds an α -feasible solution. Then, $\alpha \geq 1/2$.*

Proof. Consider the gadget LP above. Note that if \mathcal{M} guarantees $\alpha < 1/2$, then $|x_i - D_i| < 1/2$ so rounding x_i to 0 or 1 will reconstruct D_i exactly. By Theorem 35, this is impossible under differential privacy. \square

5.2 High-Sensitivity Objective

Consider a database $D \in \{0, 1\}^n$ with exactly $n/2$ zeros, and the following LP:

$$\begin{aligned} & \text{maximize } \sum_i D_i x_i - n/2 \\ \text{s.t. } & \sum_i x_i = n/2, \quad x_i \in [0, 1] \end{aligned}$$

Note that swapping a zero and a non-zero bit in D will change exactly two objective coefficients in the LP by 1. Observe that this is similar to the objective private LP (Section 4.5) because we are only allowing the objective to change. However here we consider the setting where a single objective coefficient changes arbitrarily, rather than by a small amount.

Theorem 37. *Suppose mechanism \mathcal{M} is (ϵ, δ) -high-sensitivity objective private, and with probability at least $1 - \beta$, finds an exactly feasible, additive (αn) -optimal solution. Then, $\alpha \geq c(2\epsilon, \delta(1 + e^\epsilon), \beta)$.*

Proof. Consider the gadget LP above. Note that the optimal solution is $x_i = D_i$, with objective 0. With probability at least $1 - \beta$, \mathcal{M} finds a solution x^* with objective at least $-\alpha n$. In this case, x^* places at most αn mass on indices with $D_i = 0$, so at least $(1 - \alpha)n$ mass of D and x^* are shared. Thus,

$$\|D - x^*\|_1 \leq \alpha n.$$

Since a change in D leads to a distance two change in the LP, the composition is $(2\epsilon, \delta(1 + e^\epsilon))$ -private. By Theorem 35, $\alpha \geq c(2\epsilon, \delta(1 + e^\epsilon), \beta)$. \square

5.3 High-Sensitivity Constraints/Columns

Consider a database $D \in \{0, 1\}^n$ with exactly $n/2$ zeros, and the following LP:

$$\begin{aligned} & \text{find } x_i \\ \text{s.t. } & \sum_i D_i x_i = n/2 \\ & x_i \in [0, 1] \quad \text{and} \quad \sum_i x_i = n/2 \end{aligned}$$

Note that changing a single bit in D will change coefficients in a single constraint in the LP by 1. Observe that this is similar to the column private LP (Section 4.4) because we are allowing the coefficients for a single variable to change. However here we consider the setting where this coefficient can change arbitrarily, rather than by only a small amount.

This problem is also a special case of constraint private LPs (Section 3) because the coefficients in one (i.e., the only) constraint can change arbitrarily. In the current setting, we want a solution that approximately satisfies all constraints, rather than just satisfying most of the constraints.

Theorem 38. *Suppose mechanism \mathcal{M} is (ϵ, δ) -high-sensitivity constraint private, and finds an α -feasible solution that satisfies all public constraints with probability at least $1 - \beta$. Then, $\alpha \geq c(2\epsilon, \delta(1 + e^\epsilon), \beta)$.*

Proof. Consider the gadget LP above. Suppose with probability $1 - \beta$, \mathcal{M} finds x^* such that $\|Ax^* - b\|_\infty \leq \alpha$ for the gadget LP. By reasoning analogous to Theorem 37, at least $(1 - \alpha)n$ of the mass of x^* will coincide with D , hence $\alpha \geq c(2\epsilon, \delta(1 + e^\epsilon), \beta)$ by Theorem 35. \square

Also note that the LPs produced by this reduction differ only in the coefficients corresponding to two variables. Hence, Theorem 38 also shows that privately solving column private LPs to non-trivial accuracy is impossible. It's possible that a relaxed solution, similar to allowing unsatisfied constraints in the constraint-private case, could be possible under column privacy.

However, it is not enough to allow some constraints to be unsatisfied. Since we can simply duplicate the constraint in our lower bound gadget multiple times, producing a solution satisfying any *single* constraint to non-trivial accuracy is impossible under high-sensitivity column privacy. A different relaxation would be needed for non-trivial accuracy under column privacy.

6 Discussion

In this paper, we have initiated the study of privately solving general linear programs. We have given a taxonomy of private linear programs, classified by how the private data affects the LP. For each type of linear program in the taxonomy, we have either given an efficient algorithm, or an impossibility result.

One natural question is, to what extent do our results extend to other private convex programs, e.g., semidefinite programs (SDPs)? A tempting approach is to use the *Matrix Multiplicative Weights* algorithm of Arora and Kale [1] for solving SDPs. However, certain features of the standard multiplicative weights algorithm which we use crucially—such as compatibility with Bregman projections—seem more delicate when using Matrix Multiplicative Weights.

References

- [1] Sanjeev Arora and Satyen Kale. [A combinatorial, primal-dual approach to semidefinite programs.](#) In *ACM SIGACT Symposium on Theory of Computing (STOC)*, San Diego, Illinois, pages 227–236, 2007.
- [2] Sanjeev Arora, Elad Hazan, and Satyen Kale. [The multiplicative weights update method: a meta-algorithm and applications.](#) *Theory of Computing*, 8(1):121–164, 2012.
- [3] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. [Practical privacy: the sulq framework.](#) In *ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems (PODS)*, Baltimore, Maryland, 2005.
- [4] Avrim Blum, Katrina Ligett, and Aaron Roth. [A learning theory approach to noninteractive database privacy.](#) *Journal of the ACM*, 60(2):12, 2013.
- [5] Mark Bun, Jonathan Ullman, and Salil P Vadhan. [Fingerprinting codes and the price of approximate differentially private aggregation.](#) In *ACM SIGACT Symposium on Theory of Computing (STOC)*, New York, New York. ACM, 1–3 June 2014.
- [6] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. [Differentially private empirical risk minimization.](#) *Journal of Machine Learning Research*, 12: 1069–1109, 2011.

- [7] Irit Dinur and Kobbi Nissim. [Revealing information while preserving privacy](#). In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, San Diego, California, pages 202–210, 2003.
- [8] Cynthia Dwork. [Differential privacy: A survey of results](#). In *Theory and Applications of Models of Computation (TAMC)*, Xi'an, China, pages 1–19. Springer-Verlag, 2008.
- [9] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. [Calibrating noise to sensitivity in private data analysis](#). In *IACR Theory of Cryptography Conference (TCC)*, New York, New York, 2006.
- [10] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. [On the complexity of differentially private data release: efficient algorithms and hardness results](#). In *ACM SIGACT Symposium on Theory of Computing (STOC)*, Bethesda, Maryland, pages 381–390, 2009.
- [11] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. [Boosting and differential privacy](#). In *IEEE Symposium on Foundations of Computer Science (FOCS)*, Las Vegas, Nevada, pages 51–60, 2010.
- [12] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. [Differentially private combinatorial optimization](#). In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Austin, Texas, pages 1106–1125, 2010.
- [13] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. [Privately releasing conjunctions and the statistical query barrier](#). In *ACM SIGACT Symposium on Theory of Computing (STOC)*, San Jose, California, pages 803–812, 2011.
- [14] Anupam Gupta, Aaron Roth, and Jonathan Ullman. [Iterative constructions and private data release](#). In *IACR Theory of Cryptography Conference (TCC)*, Taormina, Italy, 2012.
- [15] Moritz Hardt and Guy N. Rothblum. [A multiplicative weights mechanism for privacy-preserving data analysis](#). In *IEEE Symposium on Foundations of Computer Science (FOCS)*, Las Vegas, Nevada, pages 61–70, 2010.
- [16] Moritz Hardt, Katrina Ligett, and Frank McSherry. [A simple and practical algorithm for differentially private](#). In *Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, California, pages 2348–2356. 2012.
- [17] Mark Herbster and Manfred K. Warmuth. [Tracking the best linear predictor](#). *Journal of Machine Learning Research*, 1:281–309, 2001.
- [18] Justin Hsu, Aaron Roth, and Jonathan Ullman. [Differential privacy for the analyst via private equilibrium com](#). In *ACM SIGACT Symposium on Theory of Computing (STOC)*, Palo Alto, California, pages 341–350, 2013.
- [19] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. [What can we learn privately?](#) *SIAM Journal on Computing*, 40(3):793–826, 2011.

- [20] Michael Kearns, Malleesh Pai, Aaron Roth, and Jonathan Ullman. [Mechanism design in large games: Incentives and privacy](#). In *ACM SIGACT Innovations in Theoretical Computer Science (ITCS)*, Princeton, New Jersey, 2014.
- [21] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. [Private convex empirical risk minimization and high-dimensional regression](#). *Journal of Machine Learning Research*, 1:41, 2012.
- [22] N. Littlestone and Manfred K. Warmuth. [The weighted majority algorithm](#). *Information and Computation*, 108(2):212–261, 1994.
- [23] Frank McSherry and Kunal Talwar. [Mechanism design via differential privacy](#). In *IEEE Symposium on Foundations of Computer Science (FOCS)*, Providence, Rhode Island, 2007.
- [24] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. [Smooth sensitivity and sampling in private data analysis](#). In *ACM SIGACT Symposium on Theory of Computing (STOC)*, San Diego, Illinois, pages 75–84, 2007.
- [25] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. [Fast approximation algorithms for fractional packing and covering problems](#). *Mathematics of Operations Research*, 20(2):257–301, 1995.
- [26] Aaron Roth and Tim Roughgarden. [Interactive privacy via the median mechanism](#). In *ACM SIGACT Symposium on Theory of Computing (STOC)*, Cambridge, Massachusetts, pages 765–774.
- [27] Jonathan Ullman. [Answering \$n^{2+o\(1\)}\$ counting queries with differential privacy is hard](#). In *ACM SIGACT Symposium on Theory of Computing (STOC)*, Palo Alto, California, pages 361–370, 2013.
- [28] Jonathan Ullman and Salil Vadhan. [PCPs and the hardness of generating private synthetic data](#). In *IACR Theory of Cryptography Conference (TCC)*, Providence, Rhode Island, pages 400–416, 2011.