Computing Lewis Weights to High Precision *

Maryam Fazel † Yin Tat Lee† Swati Padmanabhan† Aaron Sidford ‡

Abstract

We present an algorithm for computing approximate ℓ_p Lewis weights to high precision. Given a full-rank $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ and a scalar $p > 2$, our algorithm computes ϵ-approximate ℓ_p Lewis weights of $\mathbf{A}$ in $\widetilde{O}_p(\log(1/\epsilon))$ iterations; the cost of each iteration is linear in the input size plus the cost of computing the leverage scores of $\mathbf{DA}$ for diagonal $\mathbf{D} \in \mathbb{R}^{m \times m}$. Prior to our work, such a computational complexity was known only for $p \in (0, 4)$ [CP15], and combined with this result, our work yields the first polylogarithmic-depth polynomial-work algorithm for the problem of computing ℓ_p Lewis weights to high precision for *all* constant $p > 0$. An important consequence of this result is also the first polylogarithmic-depth polynomial-work algorithm for computing a nearly optimal self-concordant barrier for a polytope.

1 Introduction to Lewis Weights

In this paper, we study the problem of computing the ℓ_p Lewis weights[1] of a matrix.

DEFINITION 1. *[Lew78, CP15] Given a full-rank matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ and a scalar $p \in (0, \infty)$, the Lewis weights of $\mathbf{A}$ are the entries of the unique[2] vector $\overline{w} \in \mathbb{R}^m$ satisfying the equation*

$$\overline{w}_i^{2/p} = a_i^\top (\mathbf{A}^\top \overline{\mathbf{W}}^{1-2/p} \mathbf{A})^{-1} a_i \text{ for all } i \in [m], \tag{1.1}$$

where a_i is the i'th row of matrix $\mathbf{A}$ and $\overline{\mathbf{W}}$ is the diagonal matrix with vector $\overline{w}$ on the diagonal.

Motivation. We contextualize our problem with a simpler, geometric notion. Given a set of m points $\{a_i\}_{i=1}^m \in \mathbb{R}^n$ (the rows of the preceding matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$), their *John ellipsoid* [Joh48] is the minimum[3] volume ellipsoid enclosing them. This ellipsoid finds use across experiment design and computational geometry [Tod16] and is central to certain cutting-plane methods [Vai89, LSW15], an algorithm fundamental to mathematical optimization (Section 1.3). It turns out that the John ellipsoid of a set of points $\{a_i\}_{i=1}^m \in \mathbb{R}^n$ is expressible [BV04] as the solution to the following convex program, with the objective being a stand-in for the volume of the ellipsoid and the constraints encoding the requirement that each given point a_i lie *within* the ellipsoid:

$$\text{minimize}_{\mathbf{M} \succeq 0} \det(\mathbf{M})^{-1}, \text{ subject to } a_i^\top \mathbf{M} a_i \leq 1, \text{ for all } i \in [m]. \tag{1.2}$$

The problem (1.2) may be generalized by the following convex program [Woj96, CP15], the generalization immediate from substituting $p = \infty$ in (1.3):

$$\text{minimize}_{\mathbf{M} \succeq 0} \det(\mathbf{M})^{-1}, \text{ subject to } \sum_{i=1}^m (a_i^\top \mathbf{M} a_i)^{p/2} \leq 1. \tag{1.3}$$

Geometrically, (1.3) seeks the minimum volume ellipsoid with a bound on the $p/2$-norm of the distance of the points to the ellipsoid, and its solution $\mathbf{M}$ is the "Lewis ellipsoid" [CP15] of $\{a_i\}_{i=1}^m$.

The optimality condition of (1.3), written using $\overline{w} \in \mathbb{R}^m$ defined as $\overline{w}_i \stackrel{\text{def}}{=} (a_i^\top \mathbf{M} a_i)^{p/2}$, is equivalent to (1.1), and this demonstrates that solving (1.3) is one approach to obtaining the Lewis weights of $\mathbf{A}$ (see [CP15]). This

*The full version of the paper can be accessed at https://arxiv.org/abs/2110.15563

†University of Washington

‡Stanford University

[1] From hereon, we refer to these simply as "Lewis weights" for brevity.

[2] Existence and uniqueness was first proven by D.R.Lewis [Lew78], after whom the weights are named.

[3] The John ellipsoid may also refer to the maximal volume ellipsoid enclosed by the set $\{x : |x^\top a_i| \leq 1\}$, but in this paper, we use the former definition.

equivalence also underscores the fact that the problem of computing Lewis weights is a natural ℓ_p generalization of the problem of computing the John ellipsoid.

More broadly, Lewis weights are ubiquitous across statistics, machine learning, and mathematical optimization in diverse applications, of which we presently highlight two (see Section 1.3 for details). First, their interpretation as "importance scores" of rows of matrices makes them key to shrinking the row dimension of input data [DMM06]. Second, through their role in constructing self-concordant barriers of polytopes [LS14], variants of Lewis weights have found prominence in recent advances in the computational complexity of linear programming.

From a purely optimization perspective, Lewis weights may be viewed as the optimal solution to the following convex optimization problem (which is in fact essentially dual to (1.3)):

$$\overline{w} = \arg\min_{w \in \mathbb{R}^m_{>0}} \mathcal{F}(w) \stackrel{\text{def}}{=} -\log\det\left(\mathbf{A}^\top \mathbf{W} \mathbf{A}\right) + \frac{1}{1+\alpha} \mathbf{1}^\top w^{1+\alpha}, \text{ for } \alpha = \tfrac{2}{p-2}. \tag{1.4}$$

As elaborated in [CP15, LS19], the reason this problem yields the Lewis weights is that an appropriate scaling of its solution $\overline{w}$ transforms its optimality condition from $\overline{w}_i^\alpha = a_i^\top (\mathbf{A}^\top \overline{\mathbf{W}} \mathbf{A})^{-1} a_i$ to (1.1). The problem (1.4) is a simple and natural one and, in the case of $\alpha = 1$ (corresponding to the John ellipsoid), has been the subject of study for designing new optimization methods [Tod16].

In summary, Lewis weights naturally arise as generalizations of extensively studied problems in convex geometry and optimization. This, coupled with their role in machine learning, makes understanding the complexity of computing Lewis weights, i.e., solving (1.4), a fundamental problem.

Our Goal. We aim to design *high-precision* algorithms for computing ε-*approximate* Lewis weights, i.e., a vector $w \in \mathbb{R}^m$ satisfying

$$w_i \approx_\varepsilon \overline{w}_i, \text{ for all } i \in [m], \text{ where } \overline{w} \text{ is defined in (1.1) and (1.4).} \tag{1.5}$$

where $a \approx_\varepsilon b$ is used to denote $(1-\varepsilon)a \leq b \leq (1+\varepsilon)a$. To this end, we design algorithms to solve the convex program (1.4) to $\widetilde{\varepsilon}$-additive accuracy for an appropriate $\widetilde{\varepsilon} = \text{poly}(\varepsilon, n)$, which we prove suffices in Lemma 2.1.

By a "high-precision" algorithm, we mean one with a runtime polylogarithmic in ε. We emphasize that for several applications such as randomized sampling [CP15], *approximate* Lewis weights suffice; however, we believe that high-precision methods such as ours enrich our understanding of the structure of the optimization problem (1.4). Further, as stated in Theorem 1.3, such methods yield new runtimes for directly computing a near-optimal self-concordant barrier for polytopes.

We use *number of leverage score computations* as the complexity measure of our algorithms. Our choice is a result of the fact that leverage scores of appropriately scaled matrices appear in both $\nabla \mathcal{F}(w)$ (see Lemma 2.3) and in the verification of correctness of Lewis weights. This measure of complexity stresses the *number of iterations* rather than the details of iteration costs (which depend on exact techniques used for leverage core computation, e.g., fast matrix multiplication) and is consistent with many prior algorithms (see Table 1).

Prior Results. The first polynomial-time algorithm for computing Lewis weights was presented by [CP15] and performed only $\widetilde{O}_p(\log(1/\varepsilon))$[4] leverage score computations. However, *their result holds only for* $p \in (0, 4)$. We explain the source of this limited range in Section 1.2.

In comparison, for $p \geq 4$, existing algorithms are slower: the algorithms by [CP15], [Lee16], and [LS19] perform $\widetilde{\Omega}(n)$, $\widetilde{O}(1/\varepsilon)$, and $\widetilde{O}(\sqrt{n})$ leverage score computations, respectively. [CP15] also gave an algorithm with total runtime $\mathcal{O}(\frac{1}{\varepsilon}\,\mathbf{nnz}(\mathbf{A}) + c_p n^{O(p)})$. Of note is the fact that the algorithms with runtimes polynomial in $1/\varepsilon$ ([Lee16, CP15]) satisfy the weaker approximation condition $\overline{w}_i^{2/p} \approx_\varepsilon a_i^\top (\mathbf{A}^\top \overline{\mathbf{W}}^{1-2/p} \mathbf{A})^{-1} a_i$, which is in fact implied by our condition (1.5).

We display these runtimes in Table 1, assuming that the cost of a leverage score computation is $O(mn^2)$ (which, we reiterate, may be reduced through the use of fast matrix multiplication). In terms of the number of leverage score computations, Table 1 highlights the contrast between the *polylogarithmic* dependence on input size and accuracy for $p \in (0, 4)$ and *polynomial* dependence on these factors for $p \geq 4$. *The motivation behind our paper is to close this gap.*

[4] We use O_p to hide a polynomial in p and $\widetilde{O}$ and $\widetilde{\Omega}$ to hide factors polylogarithmic in p, n, and m.

1.1 Our Contribution. We design *an algorithm that computes Lewis weights to high precision for all $p > 2$ using only $\widetilde{O}_p(\log(1/\varepsilon))$ leverage score computations.* Together with [CP15]'s result for $p \in (0, 4)$, our result therefore completes the picture on a near-optimal reduction from leverage scores to Lewis weights for all $p > 0$.

THEOREM 1.1. (MAIN THEOREM (PARALLEL)) *Given a full-rank matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $p \geq 4$, we can compute (Algorithms 1 and 2) its ε-approximate Lewis weights (1.5) in $O(p^3 \log(mp/\varepsilon))$ iterations[5]. Each iteration computes the leverage scores of a matrix $\mathbf{DA}$ for a diagonal matrix $\mathbf{D}$. The total runtime is $O(p^3 mn^2 \log(mp/\varepsilon))$, with $O(p^3 \log(mp/\varepsilon) \log^2(m))$ depth.*

Theorem 1.1 is attained by a parallel algorithm for computing Lewis weights that consists of polylogarithmic rounds of leverage score computations and therefore has polylogarithmic-depth, a result that was not known prior to this work.

THEOREM 1.2. (MAIN THEOREM (SEQUENTIAL)) *Given a full-rank matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $p \geq 4$, we can compute (Algorithms 1 and 3) its ε-approximate Lewis weights (1.5) in $O(pm \log(mp/\varepsilon))$ iterations. Each iteration computes the leverage score of one row of $\mathbf{DA}$ for a diagonal matrix $\mathbf{D}$. The total runtime is $O(pmn^2 \log(mp/\varepsilon))$.*

REMARK 1.1. *The solution to (1.3) characterizes a "Lewis ellipsoid," and the ℓ_∞ Lewis ellipsoid of $\mathbf{A}$ is precisely its John ellipsoid. After symmetrization [Tod16], computing the John ellipsoid is equivalent to solving a linear program (LP). Therefore, computing Lewis weights in $O(\log(mp/\varepsilon))$ iterations would imply a polylogarithmic-depth algorithm for solving LPs, which, given the current $O(\sqrt{n})$ depth [LS19], would be a significant breakthrough in the field of optimization. We therefore believe that it would be difficult to remove the polynomial dependence on p in our runtime.*

Authors	Range of p	Number of Leverage Score Computations/Depth	Total Runtime
[CP15]	$p \in (0, 4)$	$O\left(\frac{1}{1-\lvert 1-p/2\rvert} \cdot \log\left(\frac{\log(m)}{\varepsilon}\right)\right)$	$O\left(\frac{1}{1-\lvert 1-p/2\rvert} \cdot mn^2 \cdot \log\left(\frac{\log(m)}{\varepsilon}\right)\right)$
[CP15]	$p \geq 4$	$\Omega(n)$	$\Omega(mn^3 \cdot \log\left(\frac{m}{\varepsilon}\right))$
[CP15]*	$p \geq 4$	not applicable	$O\left(\frac{\text{nnz}(\mathbf{A})}{\varepsilon} + c_p n^{O(p)}\right)$
[Lee16]*	$p \geq 4$	$O\left(\frac{1}{\varepsilon} \cdot \log(m/n)\right)$	$O\left(\left(\frac{\text{nnz}(\mathbf{A})}{\varepsilon} + \frac{n^3}{\varepsilon^3}\right) \cdot \log(m/n)\right)$
[LS19]	$p \geq 4$	$O(p^2 \cdot n^{1/2} \cdot \log\left(\frac{1}{\varepsilon}\right))$	$O(p^2 \cdot mn^{2.5} \cdot \text{poly}\log\left(\frac{m}{\varepsilon}\right))$
Theorem 1.1	$p \geq 4$	$O(p^3 \cdot \log\left(\frac{mp}{\varepsilon}\right))$	$O(p^3 \cdot mn^2 \cdot \log\left(\frac{mp}{\varepsilon}\right))$

Table 1: Runtime comparison for computing Lewis weights. Results with asterisks use a weaker notion of approximation than our paper (1.1). All dependencies on n in the running times of these methods can be improved using fast matrix multiplication.

1.2 Overview of Approach. Before presenting our algorithm, we describe obstacles to directly extending previous work on the problem for $p \in (0, 4)$ to the case $p \geq 4$. For $p \in (0, 4)$, [CP15, LS19] design algorithms that, with a single computation of leverage scores, make constant (dependent on p) multiplicative progress on error (such as function error or distance to optimal point), thus attaining runtimes polylogarithmic in ε. However, these methods crucially rely on *contractive properties* that, in contrast to our work, do *not* necessarily hold for $p \geq 4$.

For example, one of the algorithms in [CP15] starts with a vector $v \approx_c \overline{w}$, where $\overline{w}$ is the vector of true Lewis weights and c some constant. Consequently, we have $(a_i^\top (\mathbf{A}^\top \mathbf{V}^{1-2/p} \mathbf{A})^{-1} a_i)^{p/2} \approx_{c^{|p/2-1|}} (a_i^\top (\mathbf{A}^\top \overline{\mathbf{W}}^{1-2/p} \mathbf{A})^{-1} a_i)^{p/2}$. Due to this map being a contraction for $|p/2 - 1| < 1$, or equivalently, for $p \in (0, 4)$,

[5] Our algorithms work for all $p > 2$, as can be seen in our proof in Section 3.1. However, for $p \in (2, 4)$, the algorithm of [CP15] is faster, and therefore, in our main theorems, we state runtimes only for $p \geq 4$.

$O(\log(\log n))$ recursive calls to it give Lewis weights for $p < 4$, but the contraction - and, by extension, this method - does not immediately extend to the setting $p \geq 4$.

Prior algorithms for $p \geq 4$ therefore resort to alternate optimization techniques. [CP15] frames Lewis weights computation as determinant maximization (1.3) (see Section D) and applies cutting plane methods [GLS81, LSW15]. [Lee16] uses mirror descent, and [LS19] uses homotopy methods. These approaches yield runtimes with $\text{poly}(n)$ or $\text{poly}(\frac{1}{\varepsilon})$ leverage score computations, and therefore, in order to attain runtimes of $\text{polylog}(1/\varepsilon)$ leverage score computations, we need to rethink the algorithm.

Our Approach. As stated in Section 1, to obtain ε-approximate Lewis weights for $p \geq 4$, we compute a w that satisfies $\mathcal{F}(\overline{w}) \leq \mathcal{F}(w) \leq \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$, where $\mathcal{F}$ and $\overline{w}$ are as defined in (1.4) and $\widetilde{\varepsilon} = O(\text{poly}(n, \varepsilon))$. In light of the preceding bottlenecks in prior work, we circumvent techniques that directly target constant multiplicative progress (on some potential) in each iteration.

Our main technical insight is that *when the leverage scores for the current weight $w \in \mathbb{R}^n_{>0}$ satisfy a certain technical condition (inequality (1.6))*, it is indeed possible to update w to get multiplicative decrease in function error $(\mathcal{F}(w) - \mathcal{F}(\overline{w}))$, thus resulting in our target runtime. To turn this insight into an algorithm, we design a corrective procedure that ensures that (1.6) is always satisfied: in other words, whenever (1.6) is *violated*, this procedure updates w so that the new w *does* satisfy (1.6), setting the stage for the aforementioned multiplicative progress. An important additional property of this procedure is that it does *not* increase the objective function and is therefore in keeping with our goal of minimizing (1.4).

Specifically, the technical condition that our geometric decrease in function error hinges on is

$$\max_{i \in [m]} \frac{a_i^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_i}{w_i^\alpha} \leq 1 + \alpha \,. \tag{1.6}$$

This ratio follows naturally from the gradient and Hessian of the function objective (see Lemma 2.3). Our algorithm's update rule to solve (1.4) is obtained from minimizing a second-order approximation to the objective at the current point, and the condition specified in (1.6) allows us to relate the progress of a type of quasi-Newton step to lower bounds on the progress there is to make, which is critical to turning a runtime of $\text{poly}(1/\varepsilon)$ into $\text{polylog}(1/\varepsilon)$ (Lemma 2.5).

The process of updating w so that (1.6) goes from being violated to being satisfied corresponds, geometrically, to sufficiently rounding the ellipsoid $\mathcal{E}(w) = \{x : x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x \leq 1\}$; specifically, the updated ellipsoid satisfies $\mathcal{E}(w) \subseteq \{\|\mathbf{W}^{\frac{1}{2-p}} \mathbf{A} x\|_\infty \leq \sqrt{1+\alpha}\}$ (see Section C), and this is the reason we use the term "rounding" to describe our corrective procedure to get w to satisfy (1.6) and the term "rounding condition" to refer to (1.6).

We develop two versions of rounding: a parallel method and a sequential one that has an improved dependence on p. Each version is based on the principles that (1) one can increase those entries of w at which the rounding condition (1.6) does not hold *while decreasing* the objective value, and (2) the vector w obtained after this update is closer to satisfying (1.6).

We believe that such a principle of identifying a technical condition needed for fast convergence and the accompanying rounding procedures could be useful in other optimization problems. Additionally, we develop Algorithm 4, which, by varying the step sizes in the update rule, maintains (1.6) as invariant, thereby eliminating the need for a separate rounding and progress steps.

1.3 Applications and Related Work. We elaborate here on the applications of Lewis weights we briefly alluded to in Section 1. While for many applications (such as pre-processing in optimization [CP15]) approximate weights suffice, solving regularized D-optimal and computing $\tilde{O}(n)$ self-concordant barriers to high precision do use high precision Lewis weights.

Pre-processing in optimization. Lewis weights are used as scores to sample rows of an input tall data matrix so the ℓ_p norms of the product of the matrix with vectors are preserved. They have been used in row sampling algorithms for data pre-processing [DMM06, DMIMW12, LMP13, CLM+15, PPP21], for computing dimension-free strong coresets for k-median and subspace approximation [SW18], and for fast tensor factorization in the streaming model [CCDS20]. Lewis weights are also used for ℓ_1 regression, a popular model in machine learning used to capture robustness to outliers, in [DLS18] for stochastic gradient descent pre-conditioning, [LWYZ20] for quantile regression, [BDM+20] to provide algorithms for linear algebraic problems in the sliding window model, and in [CD21] for bounds on query complexity of least absolute deviation regression.

John ellipsoid and D-optimal design. As noted in Remark 1.1, a fast algorithm for Lewis weights could yield faster algorithms for computing John ellipsoid, a problem with a long history of work [Kha96, SF04, KY05, DAST08, CCLY19, ZF20]. It is known [Tod16] that the John ellipsoid problem is dual to the (relaxed) D-optimal experiment design problem [Puk06]. D-optimal design seeks to select a set of linear experiments with the largest confidence ellipsoid for its least-square estimator [AZLSW17, MSTX19, SX20].

Our problem (1.4) is equivalent to $\frac{p}{p-2}$-regularized D-optimal design, which can be interpreted as enforcing a polynomial experiment cost: viewing w_i as the fraction of resources allocated to experiment i, each w_i is penalized by $w_i^{\frac{p}{p-2}}$. This regularization also appears in fair packing and fair covering problems [MSZ16, DFO20] from operations research.

Self-concordance. Self-concordant barriers are fundamental in convex optimization [NN94], combinatorial optimization [LS14], sampling [KN09, LLV20], and online learning [AHR08]. Although there are (nearly) optimal self-concordant barriers for any convex set [NN94, BE15, LY18], computing them involves sampling from log-concave distributions, itself an expensive process with a poly$(1/\varepsilon)$ runtime. [LS14] shows how to construct nearly optimal barriers for polytopes using Lewis weights. Unfortunately, doing so still requires polynomial-many steps to compute these weights; [LS14] bypass this issue by showing it suffices to work with Lewis weights for $p \approx 1$. In this paper, we show how to compute Lewis weights by computing leverage scores of polylogarithmic-many matrices. This gives the first nearly optimal self-concordant barrier for polytopes that can be evaluated to high accuracy with depth polylogarithmic in the dimension.

THEOREM 1.3. (APPLYING THEOREM 1.1 TO [LS19, SECTION 5]) *Given a non-empty polytope $P = \{x \in \mathbb{R}^n \mid \mathbf{A}x > b\}$ for full rank $\mathbf{A} \in \mathbb{R}^{m \times n}$, there is a $O(n \log^5 m)$-self concordant barrier ψ for P such that for any $\epsilon > 0$ and $x \in P$, in $O(mn^{\omega-1} \log^3 m \log(m/\epsilon))$-work and $O(\log^3 m \log(m/\epsilon))$-depth, we can compute $g \in \mathbb{R}^n$ and $\mathbf{H} \in \mathbb{R}^{n \times n}$ with $\|g - \nabla\psi(x)\|_{\nabla^2\psi(x)^{-1}} \leq \epsilon$ and $\nabla^2\psi(x) \preceq \mathbf{H} \preceq O(\log m)\nabla^2\psi(x)$. With an additional $O(m^{\omega+o(1)})$ work, $\mathbf{H} \in \mathbb{R}^{n \times n}$ with $(1-\epsilon)\nabla^2\psi(x) \preceq \mathbf{H} \preceq O(1+\epsilon)\nabla^2\psi(x)$ can be computed as well.*

1.4 Notation and Preliminaries. We use $\mathbf{A}$ to denote our full-rank $m \times n$ ($m \geq n$) real-valued input matrix and $\overline{w} \in \mathbb{R}^m$ to denote the vector of Lewis weights of $\mathbf{A}$, as defined in (1.1) and (1.4). All matrices appear in boldface uppercase and vectors in lowercase. For any vector (say, σ), we use its uppercase boldfaced form ($\mathbf{\Sigma}$) to denote the diagonal matrix $\mathbf{\Sigma}_{ii} = \sigma_i$. For a matrix $\mathbf{M}$, the matrix $\mathbf{M}^{(2)}$ is the Schur product (entry-wise product) of $\mathbf{M}$ with itself. For matrices $\mathbf{A}$ and $\mathbf{B}$, we use $\mathbf{A} \succeq \mathbf{B}$ to mean $\mathbf{A} - \mathbf{B}$ is positive-semidefinite. For vectors a and b, the inequality $a \leq b$ applies entry-wise. We use e_i to denote the i'th standard basis vector. We define $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$. As in (1.4), since we defined $\alpha \stackrel{\text{def}}{=} \frac{2}{p-2}$, the ranges of $p \in (2, 4)$ and $p \geq 4$ translate to $\alpha > 1$ and $\alpha \in (0, 1]$, respectively. From hereon, we work with α. We also define $\bar{\alpha} = \max\{1, \alpha\}$. For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $w \in \mathbb{R}^m_{>0}$, we define the projection matrix $\mathbf{P}(w) \stackrel{\text{def}}{=} \mathbf{W}^{1/2}\mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{W}^{1/2} \in \mathbb{R}^{m \times m}$. The quantity $\mathbf{P}(w)_{ii}$ is precisely the leverage score of the i'th row of $\mathbf{W}^{1/2}\mathbf{A}$:

$$\sigma_i(w) \stackrel{\text{def}}{=} w_i \cdot a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i. \tag{1.7}$$

FACT 1.1. ([LS14]) *For all $w \in \mathbb{R}^m_{>0}$ we have that $0 \leq \sigma_i(w) \leq 1$ for all $i \in [m]$, $\sum_{i \in [m]} \sigma_i(w) \leq n$, and $\mathbf{0} \preceq \mathbf{P}(w)^{(2)} \preceq \mathbf{\Sigma}(w)$.*

2 Our Algorithm

We present Algorithm 1 to compute an $\widetilde{\varepsilon}$-additive solution to (1.4). We first provide the following definitions that we frequently refer to in our algorithm and analysis. Given $\alpha > 0$ and $w \in \mathbb{R}^m_{>0}$, the i'th coordinate of $\rho(w) \in \mathbb{R}^m$ is

$$\rho_i(w) \stackrel{\text{def}}{=} \frac{\sigma_i(w)}{w_i^{1+\alpha}}. \tag{2.1}$$

Based on this quantity, we define the following procedure, derived from approximating a quasi-Newton update on the objective $\mathcal{F}$ from (1.4):

$$[\mathbf{Descent}(w, \mathrm{C}, \eta)]_i \stackrel{\text{def}}{=} w_i \left[1 + \eta_i \cdot \frac{\rho_i(w) - 1}{\rho_i(w) + 1}\right] \text{ for all } i \in \mathrm{C} \subseteq \{1, 2, \ldots, m\}. \tag{2.2}$$

Using these definitions, we can now describe our algorithm. Depending on whether the following condition ("rounding condition") holds, we run either **Descent**(·) or **Round**(·) in each iteration:

$$\rho_{\max}(w) \stackrel{\text{def}}{=} \max_{i\in[m]} \rho_i(w) \le 1+\alpha. \tag{2.3}$$

Specifically, if (2.3) is *not* satisfied, we run **Round**(·), which returns a vector that *does* satisfy it without increasing the objective value. We design two versions of **Round**(·), one parallel (Algorithm 2) and one sequential (Algorithm 3), with the sequential algorithm having an improved dependence on α, to update the coordinates violating (2.3). We apply one extra step of rounding to the vector returned after $\mathcal{T}_{\text{total}}$ iterations of Algorithm 1 and transform it appropriately to obtain our final output. In the following lemma (proved in Section B), we justify that this output is indeed the solution to (1.5).

LEMMA 2.1. (LEWIS WEIGHTS FROM OPTIMIZATION SOLUTION) *Let $w \in \mathbb{R}^m_{>0}$ be a vector at which the objective (1.4) is $\widetilde{\varepsilon}$-suboptimal in the additive sense for $\widetilde{\varepsilon} = \frac{\alpha^8\varepsilon^4}{(25m(\sqrt{n}+\alpha)(\alpha+\alpha^{-1}))^4}$, i.e., $\mathcal{F}(\overline{w}) \le \mathcal{F}(w) \le \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$. Further assume that w satisfies the rounding condition: $\rho_{\max}(w) \le 1+\alpha$. Then, the vector $\widehat{w}$ defined as $\widehat{w}_i = (a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i)^{1/\alpha}$ satisfies $\widehat{w}_i \approx_\varepsilon \overline{w}_i$ for all $i \in [m]$, thus achieving the goal spelt out in (1.5).*

Algorithm 1 Lewis Weight Computation Meta-Algorithm

Input: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, parameter $p > 2$, accuracy ε
Output: Vector $\widehat{w} \in \mathbb{R}^m_{>0}$ that satisfies (1.5)
For all $i \in [m]$, initialize $w_i^{(0)} = \frac{n}{m}$.
Set $\alpha = \frac{2}{p-2}$, $\bar{\alpha} = \max(\alpha, 1)$, $\widetilde{\varepsilon} = \frac{\alpha^8\varepsilon^4}{(25m(\sqrt{n}+\alpha)(\alpha+\alpha^{-1}))^4}$, and $\mathcal{T}_{\text{total}} = \mathcal{O}(\max(\alpha^{-1}, \alpha)\log(m/\widetilde{\varepsilon}))$.
for $k = 1, 2, 3, \ldots, \mathcal{T}_{total}$ **do**
  $\widetilde{w}^{(k)} \leftarrow$ **Round**$(w^{(k-1)}, \mathbf{A}, \alpha)$ ▷ Invoke Algorithm 2 (parallel) or 3 (sequential)
  $w^{(k)} \leftarrow$ **Descent**$(\widetilde{w}^{(k)}, [m], \frac{1}{3\bar{\alpha}}\mathbf{1})$ ▷ See (2.2) and Lemma 2.2
end
Set $w_{\text{R}} \leftarrow$ **Round**$(w^{(\mathcal{T}_{\text{total}})}, \mathbf{A}, \alpha)$
Return $\widehat{w} \in \mathbb{R}^m_{>0}$, where $\widehat{w}_i = (a_i^\top(\mathbf{A}^\top\mathbf{W}_{\text{R}}\mathbf{A})^{-1}a_i)^{1/\alpha}$. ▷ See Section B

Algorithm 2 RoundParallel$(w, \mathbf{A}, \alpha)$

Input: Vector $w \in \mathbb{R}^m_{>0}$, matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, parameter $\alpha > 0$
Output: Vector $w \in \mathbb{R}^m_{>0}$ satisfying (2.3)
Define $\rho(w)$ as in (2.1)
while $\text{C} = \{i \mid \rho_i(w) > 1+\alpha\} \neq \emptyset$ **do**
  $w \leftarrow$ **Descent**$(w, \text{C}, \frac{1}{3\bar{\alpha}}\mathbf{1})$ ▷ See Section 3
end
Return w

Algorithm 3 RoundSequential$(w, \mathbf{A}, \alpha)$

Input: Vector $w \in \mathbb{R}^m_{>0}$, matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, parameter $\alpha > 0$
Output: Vector $w \in \mathbb{R}^m_{>0}$ satisfying (2.3)
Define $\rho(w)$ as in (2.1) and $\sigma(w)$ as in (1.7)
Define $\text{C} = \{i \mid \rho_i(w) \ge 1\}$
for $i \in \text{C}$ **do**
  $w_i \leftarrow w_i(1+\delta_i)$, where δ_i solves $\rho_i(w) = (1+\delta_i\sigma_i(w))(1+\delta_i)^\alpha$ ▷ see Section 4
end
Return w

2.1 Analysis of Descent(·). We first analyze **Descent**(·) since it is common to both the parallel and sequential algorithms.

LEMMA 2.2. (ITERATION COMPLEXITY OF **Descent**(·)) *Each iteration of* ***Descent***(·) *(described in (2.2)) decreases the value of* $\mathcal{F}$*. Assuming that* ***Round***(·) *does not increase the value of the objective in (1.4), for any given accuracy parameter* $0 < \widetilde{\varepsilon} < 1$*, the number of* ***Descent***(·) *steps that Algorithm 1 performs before achieving* $\mathcal{F}(w) \leq \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$ *is given by the following bound:*

$$\mathcal{T}_{total} = \mathcal{O}(\max(\alpha^{-1}, \alpha)\log(m/\widetilde{\varepsilon})).$$

As is often the case to obtain such an iteration complexity, we prove Lemma 2.2 by incorporating the maximum sub-optimality in function value (Lemma 2.5) and the initial error bound (Lemma 2.4) into the inequality describing minimum function progress (Lemma 2.6). The assumption that **Round**(·) does not increase the value of the objective is justified in Lemma 3.1.

Since our algorithm leverages quasi-Newton steps, we begin our analysis by stating the gradient and Hessian of the objective in (1.4) as well as the error at the initial vector $w^{(0)}$, as measured against the optimal function value. The Hessian below is positive semidefinite when $\alpha \geq 0$ (equivalently, when $p \geq 2$) and not necessarily so otherwise. Consequently, the objective is convex for $\alpha \geq 0$, and we therefore consider only this setting throughout.

LEMMA 2.3. (GRADIENT AND HESSIAN) *For any* $w \in \mathbb{R}^m_{>0}$*, the objective in (1.4),* $\mathcal{F}(w) = -\log\det\left(\mathbf{A}^\top \mathbf{W}\mathbf{A}\right) + \frac{1}{1+\alpha}\mathbf{1}^\top w^{1+\alpha}$*, has gradient and Hessian given by the following expressions.*

$$[\nabla \mathcal{F}(w)]_i = w_i^{-1} \cdot (w_i^{1+\alpha} - \sigma_i(w)) \text{ and } \nabla^2 \mathcal{F}(w) = \mathbf{W}^{-1}\mathbf{P}(w)^{(2)}\mathbf{W}^{-1} + \alpha \mathbf{W}^{\alpha-1}.$$

LEMMA 2.4. (INITIAL SUB-OPTIMALITY) *At the start of Algorithm 1, the value of the objective of (1.4) differs from the optimum objective value as* $\mathcal{F}(w^{(0)}) \leq \mathcal{F}(\overline{w}) + n\log(m/n)$*.*

2.1.1 Minimum Progress and Maximum Sub-optimality in Descent(·). We first prove an upper bound on objective sub-optimality, necessary to obtain a runtime polylogarithmic in $1/\varepsilon$. Often, to obtain such a rate, the bound involving objective sub-optimality has a quadratic term derived from the Hessian; our lemma is somewhat non-standard in that it uses only the convexity of $\mathcal{F}$. Note that this lemma crucially uses (2.3).

LEMMA 2.5. (OBJECTIVE SUB-OPTIMALITY) *Suppose* $w \in \mathbb{R}^m_{>0}$ *and* $\rho_{\max}(w) \leq 1 + \alpha$*. Then the value of the objective of (1.4) at* w *differs from the optimum objective value as follows.*

$$\mathcal{F}(w) - \mathcal{F}(\overline{w}) \leq \sum_{i\in[m]} \frac{w_i^{1+\alpha}}{1+\alpha}\left(1 + \frac{\rho_i(w)}{\alpha}\right)(\rho_i(w) - 1)^2 \leq 5\max\{1, \alpha^{-1}\}\sum_{i\in[m]} w_i^{1+\alpha}\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1}.$$

Proof. Since $g(w) \stackrel{\text{def}}{=} -\log\det\left(\mathbf{A}^\top\mathbf{W}\mathbf{A}\right)$ is convex and $[\nabla g(w)]_i = -w_i^{-1}\sigma_i(w)$, we have

$$g(\overline{w}) \geq g(w) + \nabla g(w)^\top(\overline{w} - w) = g(w) + \sum_{i\in[m]}\left(-\frac{\sigma_i(w)\overline{w}_i}{w_i} + \sigma_i(w)\right),$$

and therefore,

$$\begin{aligned}\mathcal{F}(\overline{w}) - \mathcal{F}(w) &= g(\overline{w}) - g(w) + \frac{1}{1+\alpha}\sum_{i\in[m]}\left([\overline{w}]_i^{1+\alpha} - w_i^{1+\alpha}\right) \\ &\geq \sum_{i\in[m]} c_i \text{ where } c_i \stackrel{\text{def}}{=} -\frac{\sigma_i(w)\overline{w}_i}{w_i} + \sigma_i(w) + \frac{1}{1+\alpha}\left([\overline{w}]_i^{1+\alpha} - w_i^{1+\alpha}\right).\end{aligned}$$

To prove the claim, it suffices to bound each c_i from below. First, note that

$$c_i \geq \min_{v \geq 0} -\frac{v \cdot \sigma_i(w)}{w_i} + \sigma_i(w) + \frac{1}{1+\alpha}\left(v^{1+\alpha} - w_i^{1+\alpha}\right) = -\frac{\alpha}{1+\alpha}\left(\frac{\sigma_i(w)}{w_i}\right)^{1+\frac{1}{\alpha}} + \sigma_i(w) - \frac{w_i^{1+\alpha}}{1+\alpha}$$

$$= \frac{w_i^{1+\alpha}}{1+\alpha}\left[-\alpha\rho_i(w)^{1+\frac{1}{\alpha}} + (1+\alpha)\rho_i(w) - 1\right] \tag{2.4}$$

where the first equality used that the minimization problem is convex and the solutions to $-\sigma_i(w)w_i^{-1} + v^\alpha = 0$ (i.e. where the gradient is 0) is a minimizer, and the second equality used $\rho_i(w) = \sigma_i(w)/w_i^{1+\alpha}$. Applying $\rho_i(w) \leq 1+\alpha$, $1+x \leq \exp x$, and $\exp x \leq 1 + x + x^2$ for $x \leq 1$ yields

$$\rho_i(w)^{\frac{1}{\alpha}} = (1-(1-\rho_i(w)))^{\frac{1}{\alpha}} \leq \exp\left(\tfrac{1}{\alpha}(\rho_i(w)-1)\right) \leq 1 + \frac{1}{\alpha}(\rho_i(w)-1) + \frac{1}{\alpha^2}(\rho_i(w)-1)^2. \tag{2.5}$$

Combining (2.5) with (2.4) yields that

$$c_i \geq \frac{w_i^{1+\alpha}}{1+\alpha}\left[-\alpha\rho_i(w)\left[1 + \left(\frac{\rho_i(w)-1}{\alpha}\right) + \left(\frac{\rho_i(w)-1}{\alpha}\right)^2\right] + (1+\alpha)\rho_i(w) - 1\right]$$

$$= \frac{w_i^{1+\alpha}}{1+\alpha}\left[-1 + 2\rho_i(w) - \rho_i(w)^2 - \frac{\rho_i(w)}{\alpha}\cdot(\rho_i(w)-1)^2\right] = -\frac{w_i^{1+\alpha}}{1+\alpha}\left(1 + \frac{\rho_i(w)}{\alpha}\right)\cdot(\rho_i(w)-1)^2$$

The claim then follows from the fact that for $\rho_i(w) \leq 1+\alpha$, we have $\frac{(1+\rho_i(w)\alpha^{-1})(1+\rho_i(w))}{1+\alpha} \leq \frac{1}{1+\alpha} + \frac{1}{\alpha} + 1 + 1 + \frac{1}{\alpha} \leq 5\max\{1, \alpha^{-1}\}$. ∎

LEMMA 2.6. (FUNCTION DECREASE IN **Descent**$(\cdot)$) *Let $w, \eta \in \mathbb{R}^m_{>0}$ with $\eta_i \in [0, \frac{1}{3\bar{\alpha}}]$ for all $i \in [m]$. Further, let $w^+ =$ **Descent**$(w, [m], \eta)$, where **Descent** is defined in (2.2). Then, $w^+ \in \mathbb{R}^m_{>0}$ with the following decrease in function objective.*

$$\mathcal{F}(w^+) \leq \mathcal{F}(w) - \sum_{i\in[m]} \frac{\eta_i}{2}\cdot w_i^{1+\alpha}\cdot\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1}.$$

The proof of this lemma resembles that of quasi-Newton method: first, we write a second-order Taylor approximation of $\mathcal{F}(w^+)$ around w and apply Fact 1.1 to Lemma 2.3 to obtain the upper bound on Hessian: $\nabla^2\mathcal{F}(\widetilde{w}) = \widetilde{\mathbf{W}}^{-1}\mathbf{P}(\widetilde{w})^{(2)}\widetilde{\mathbf{W}}^{-1} + \alpha\widetilde{\mathbf{W}}^{\alpha-1} \preceq \widetilde{\mathbf{W}}^{-1}\Sigma(\widetilde{w})\widetilde{\mathbf{W}}^{-1} + \alpha\widetilde{\mathbf{W}}^{\alpha-1}$. We further use the expression for $\nabla\mathcal{F}(w)$ in this second-order approximation and simplify to obtain the claim, as detailed in Section A.

2.1.2 Iteration Complexity of Descent$(\cdot)$.

Proof. [Proof of Lemma 2.2] Since Algorithm 1 calls **Descent**$(\cdot)$ after running **Round**$(\cdot)$, the requirement $\rho_{\max}(w) \leq 1+\alpha$ in Lemma 2.5 is met. Therefore, we may combine Lemma 2.5 alongwith Lemma 2.6 and our choice of $\eta_i = \frac{1}{3\bar{\alpha}}$ in Algorithm 1 to get a geometric decrease in function error as follows.

$$\mathcal{F}(w^+) - \mathcal{F}(\overline{w}) \leq \mathcal{F}(w) - \mathcal{F}(\overline{w}) - \frac{1}{6\max(\alpha,1)}\sum_{i=1}^m w_i^{1+\alpha}\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1}$$

$$\leq \left(1 - \frac{1}{30\max(1,\alpha)\cdot\max(1,\alpha^{-1})}\right)(\mathcal{F}(w) - \mathcal{F}(\overline{w})). \tag{2.6}$$

We apply this inequality recursively over all iterations of Algorithm 1, while also using the assumption that **Round**$(\cdot)$ does not increase the objective value. Setting the final value of (2.6) to $\widetilde{\varepsilon}$, bounding the initial error as $\mathcal{F}(w) - \mathcal{F}(\overline{w}) \leq n\log(m/n) \leq m^2$ by Lemma 2.4, observing $\max(1,\alpha)\cdot\max(1,\alpha^{-1}) = \max(\alpha,\alpha^{-1})$, and taking logarithms on both sides of the inequality gives the claimed iteration complexity of **Descent**$(\cdot)$. ∎

3 Analysis of Round$(\cdot)$: The Parallel Algorithm

The main export of this section is the proof of our main theorem about the parallel algorithm (Theorem 1.1). This proof combines the iteration count of **Descent**$(\cdot)$ from the preceding section with the analysis of Algorithm 2

(invoked by **Round**(·) in the parallel setting), shown next. In Lemma 3.1, we show that **RoundParallel**(·) decreases the function objective, thereby justifying the key assumption in Lemma 2.2. Lemma 3.1 also shows an upper bound on the new value of ρ after one `while` loop of **RoundParallel**(·), and by combining this with the maximum value of ρ for termination in Algorithm 2, we get the iteration complexity of **RoundParallel**(·) in Corollary 3.1.

LEMMA 3.1. (OUTCOME OF **RoundParallel**(·)) *Let $w^+ \in \mathbb{R}^m_{>0}$ be the state of $w \in \mathbb{R}^m_{>0}$ at the end of one `while` loop of **RoundParallel**(·) (Algorithm 2). Then, $\mathcal{F}(w^+) \leq \mathcal{F}(w)$ and $\rho_{\max}(w^+) \leq (1 + \frac{\alpha}{3\bar{\alpha}(2+\alpha)})^{-\alpha}\rho_{\max}(w)$.*

Proof. Each iteration of the `while` loop in **RoundParallel**(·) performs **Descent**$(w, \mathrm{C}, \frac{1}{3\bar{\alpha}}\mathbf{1})$ over the set of coordinates $\mathrm{C} = \{i : \rho_i(w) > 1 + \alpha\}$. Lemma 2.6 then immediately proves $\mathcal{F}(w^+) \leq \mathcal{F}(w)$, which is our first claim.

To prove the second claim, note that in Algorithm 2, for every $i \in \mathrm{C}$

$$w_i^+ = w_i + \frac{w_i}{3\bar{\alpha}} \cdot \left[\frac{\rho_i(w) - 1}{\rho_i(w) + 1}\right] \geq w_i + \frac{w_i}{3\bar{\alpha}} \cdot \left[\frac{\alpha}{1 + 1 + \alpha}\right] = w_i \cdot \left(1 + \frac{\alpha}{3\bar{\alpha}(2+\alpha)}\right),$$

where the second step is by the monotonicity of $x \to \frac{x-1}{x+1}$ for $x \geq 1$. Combining it with $w_i^+ = w_i$ for all $i \notin \mathrm{C}$ implies that $w^+ \geq w$. Therefore, for all $i \in \mathrm{C}$, we have

$$\rho(w^+)_i = [w_i^+]^{-\alpha}[\mathbf{A}(\mathbf{A}^\top \mathbf{W}^+ \mathbf{A})^{-1}\mathbf{A}^\top]_{ii} \leq \left[1 + \frac{\alpha}{3\bar{\alpha}(2+\alpha)}\right]^{-\alpha} \cdot w_i^{-\alpha}[\mathbf{A}(\mathbf{A}^\top \mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top]_{ii}. \tag{3.1}$$

∎

COROLLARY 3.1. *Let w be the input to **RoundParallel**(·). Then, the number of iterations of the `while` loop of **RoundParallel**(·) is at most $O\left((1 + \alpha^{-2}) \log\left(\frac{\rho_{\max}(w)}{1+\alpha}\right)\right)$.*

Proof. Let $w^{(i)}$ be the value of w at the start of the i'th execution of the `while` loop of **RoundParallel**(·). Repeated application of Lemma 3.1 over k executions of the `while` loop gives $\rho_{\max}(w^{(k)}) \leq \rho_{\max}(w)\left(1 + \frac{\alpha}{3\bar{\alpha}(2+\alpha)}\right)^{-\alpha k}$. We set $\rho_{\max}(w)\left(1 + \frac{\alpha}{3\bar{\alpha}(2+\alpha)}\right)^{-\alpha k} \leq 1 + \alpha$ in accordance with the termination condition of **RoundParallel**(·). Next, applying $1 + x \leq \exp(x)$, and taking logarithms on both sides yields the claimed limit on the number of iterations, k. ∎

LEMMA 3.2. *Over the entire run of Algorithm 1, the `while` loop of **RoundParallel**(·) runs for at most $O\left(\mathcal{T}_{total} \cdot \alpha^{-2} \cdot \log\left(\frac{m}{n(1+\alpha)}\right)\right)$ iterations if $\alpha \in (0, 1]$ and $\mathcal{O}\left(\mathcal{T}_{total} \cdot \alpha \cdot \log\left(\frac{m}{n(1+\alpha)}\right)\right)$ iterations if $\alpha \geq 1$.*

Proof. Note that $\rho_{\max}(\frac{n}{m}) \leq (\frac{m}{n})^{1+\alpha}$; consequently, in the first iteration of Algorithm 1, there are at most $O((\alpha + \alpha^{-2}) \log(m/(n(1+\alpha))))$ iterations of the `while` loop of **RoundParallel**(·) by Corollary 3.1. Note that between each call to **RoundParallel**(·), for all $i \in [m]$,

$$w_i^+ = w_i + \frac{w_i}{3\bar{\alpha}} \cdot \left[\frac{\rho_i(w) - 1}{\rho_i(w) + 1}\right] \geq w_i + \frac{w_i}{3\bar{\alpha}} \cdot \left[\frac{-1}{1 + 1 + \alpha}\right] = w_i \cdot \left(1 - \frac{1}{(3\bar{\alpha})(2+\alpha)}\right),$$

where the first inequality is by using the fact that the output w of **RoundParallel**(·) satisfies $\rho_{\max}(w) \leq 1 + \alpha$. Therefore, applying the same logic as in (3.1), we get that between two calls to **RoundParallel**(·), the value of $\rho_i(w)$ increases by at most $\left(1 - \frac{1}{(3\bar{\alpha})(2+\alpha)}\right)^{-(1+\alpha)} = O(1)$ for all $i \in [m]$. Combining this with Corollary 3.1 and the total initial iteration count and observing that $\mathcal{T}_{\text{total}}$ is the total number of calls to **RoundParallel**(·) finishes the proof. ∎

3.1 Proof of Main Theorem (Parallel).

Proof. (Proof of Theorem 1.1) First, we show correctness. Note that, as a corollary of Lemma 2.2, $\mathcal{F}(w^{(\mathcal{T}_{\text{total}})}) \leq \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$. By the properties of **Round** as shown in Lemma 3.1, we also have that $\mathcal{F}(w_{\text{R}}) \leq \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$ and $\rho_{\max}(w_{\text{R}}) \leq 1 + \alpha$ for $w_{\text{R}} = \textbf{Round}(w^{(\mathcal{T}_{\text{total}})}, \mathbf{A}, \alpha)$. Therefore, Lemma 2.1 is applicable, and by the choice of $\widetilde{\varepsilon} = \frac{\alpha^4 \varepsilon^4}{(2m(\sqrt{n}+\alpha)(\alpha+\alpha^{-1}))^4}$, we conclude that $\widehat{w} \in \mathbb{R}^m$ defined as $\widehat{w}_i = (a_i^\top (\mathbf{A}^\top \mathbf{W}_{\text{R}} \mathbf{A})^{-1} a_i)^{1/\alpha}$ satisfies $\widehat{w}_i \approx_\varepsilon \overline{w}_i$ for all $i \in [m]$. Combining the iteration counts of **Descent**$(\,\cdot\,)$ from Lemma 2.2 and of **RoundParallel**$(\,\cdot\,)$ from Lemma 3.2 yields the total iteration count as $O(\alpha^{-3} \log(m/(\varepsilon\alpha)))$ if $\alpha \leq 1$ and $O(\alpha^2 \log(m/\varepsilon))$ if $\alpha > 1$. As stated in Section 1.4, $\alpha = \frac{2}{p-2}$, and so translating these rates in terms of p gives $O(p^3 \log(mp/\varepsilon))$ for $p \geq 4$ and $O(p^{-2} \log(mp/\varepsilon))$ for $p \in (2, 4)$, thereby proving the stated claim. The cost per iteration is $O(mn^2)$[6] for multiplying two $m \times n$ matrices. ∎

4 Analysis of Round$(\,\cdot\,)$: Sequential Algorithm

We now analyze Algorithm 3. Note that these proofs work for all $\alpha > 0$.

LEMMA 4.1. (COORDINATE STEP PROGRESS) *Given $w \in \mathbb{R}^m_{>0}$, a coordinate $i \in [m]$, and $\delta_i \in \mathbb{R}$, we have*

$$\mathcal{F}(w + \delta_i w_i e_i) = \mathcal{F}(w) - \log(1 + \delta_i \sigma_i(w)) + \frac{w_i^{1+\alpha}}{1+\alpha}((1+\delta_i)^{1+\alpha} - 1).$$

Proof. By definition of $\mathcal{F}$, we have

$$\mathcal{F}(w + \delta_i w_i e_i) = -\log\det\big(\mathbf{A}^\top \mathbf{W} \mathbf{A} + \delta_i w_i a_i a_i^\top\big) + \frac{1}{1+\alpha} \sum_{j \neq i} w_j^{1+\alpha} + \frac{w_i^{1+\alpha}}{1+\alpha}(1+\delta_i)^{1+\alpha}.$$

Recall the matrix determinant lemma: $\det\big(\mathbf{A} + uv^\top\big) = (1 + v^\top \mathbf{A}^{-1} u) \det(\mathbf{A})$. Applying it to $\det\big(\mathbf{A}^\top \textbf{diag}(w + \delta_i w_i e_i) \mathbf{A}\big)$ in the preceding expression for $\mathcal{F}(w + \delta_i w_i e_i)$ proves the lemma. ∎

LEMMA 4.2. (COORDINATE STEP OUTCOME) *Given $w \in \mathbb{R}^m_{>0}$ and $\text{C} = \{i : \rho_i(w) \geq 1\}$, let $w^+ = w + \delta_i w_i e_i$ for any $i \in \text{C}$, where $\delta_i = \arg\min_\delta \left[-\log(1 + \delta\sigma_i(w)) + \frac{1}{1+\alpha} w_i^{1+\alpha}((1+\delta)^{1+\alpha} - 1)\right]$. Then, we have $\mathcal{F}(w^+) \leq \mathcal{F}(w)$ and $\rho_i(w^+) \leq 1$.*

Proof. We note that $\min_\delta \left[-\log(1 + \delta\sigma_i(w)) + \frac{1}{1+\alpha} w_i^{1+\alpha}((1+\delta)^{1+\alpha} - 1)\right] \leq 0$. Then, Lemma 4.1 implies the first claim. Since the update rule optimizes over $\mathcal{F}$ coordinate-wise, at each step the optimality condition given by $\rho_i(w^+) = 1$ is met for each $i \in \text{C}$. The second claim is then proved by noting that for $j \neq i$, $w_j^+ = w_j$ and by the Sherman-Morrison-Woodbury identity, $\rho_j(w^+) \leq \rho_j(w)$:

$$a_j^\top (\mathbf{A}^\top \mathbf{W}^+ \mathbf{A})^{-1} a_j = a_j^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_j - \delta_i w_i \frac{(a_j^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_j)^2}{1 + \delta_i w_i a_i^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_i} \leq a_j^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_j.$$

∎

LEMMA 4.3. (NUMBER OF COORDINATE STEPS) *For any $0 \leq \widetilde{\varepsilon} \leq 1$, over all $\mathcal{T}_{total}$ iterations of Algorithm 1, there are at most $O(m \max(\alpha, \alpha^{-1}) \log(m/\widetilde{\varepsilon}))$ coordinate steps (see Algorithm 3).*

Proof. There are at most m coordinate steps in each call to Algorithm 3. Combining this with the value of $\mathcal{T}_{\text{total}}$ in Algorithm 1 gives the count of $O(m\alpha^{-1} \log(m/\widetilde{\varepsilon}))$ coordinate steps. ∎

4.1 Proof of Main Theorem (Sequential).

We now combine the preceding results to prove the main theorem about the sequential algorithm (Algorithm 1 with Algorithm 3).

[6] This can be improved to $O(mn^{\omega-1})$ using fast matrix multiplication.

Proof. (Proof of Theorem 1.2) The proof of correctness is the same as that for Theorem 1.1 since the parallel and sequential algorithms share the same meta-algorithm. Computing leverage scores in the sequential version (Algorithm 1 with Algorithm 3) takes $O(m \max(\alpha, \alpha^{-1}) \log(m/(\alpha\varepsilon)))$ coordinate steps. The costliest component of a coordinate step is computing $a_i^\top (\mathbf{A}^\top (\mathbf{W} + \delta_i w_i e_i e_i^\top)\mathbf{A})^{-1} a_i$. By the Sherman-Morrison-Woodbury formula, computing this costs $O(n^2)$ for each coordinate. Since the initial cost to compute $(\mathbf{A}^\top \mathbf{W}\mathbf{A})^{-1}$ is $O(mn^2)$, the total run time is $O(\max(\alpha, \alpha^{-1}) mn^2 \log(m/\varepsilon))$. When translated in terms of p, this gives $O(pmn^2 \log(mp/\varepsilon))$ for $p \geq 4$ and $O(p^{-1} mn^2 \log(mp/\varepsilon))$ for $p \in (2, 4)$. ∎

5 A "One-Step" Parallel Algorithm

We conclude our paper with an alternative algorithm (Algorithm 4) in which each iteration avoids any rounding and performs only **Descent**(·).

Algorithm 4 One-Step Algorithm

Input: Matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, parameter $p > 2$, accuracy ε
Output: Vector $\widehat{w} \in \mathbb{R}^m_{>0}$ that satisfies (1.5)
For all $i \in [m]$, initialize $w_i^{(0)} = 1$. Set $\alpha = \frac{2}{p-2}$. Set $\widetilde{\varepsilon} = \frac{\alpha^4 \varepsilon^4}{(2m(\sqrt{n}+\alpha)(\alpha+\alpha^{-1}))^4}$.
Set $\beta = \frac{1}{1000} \min(\alpha^2, 1)$ and $\mathcal{T}_{\text{total}} = \begin{cases} \mathcal{O}(\alpha^{-3} \log(mp/\widetilde{\varepsilon})) & \text{if } \alpha \in (0, 1] \\ \mathcal{O}(\alpha^2 \log(mp/\widetilde{\varepsilon})) & \alpha > 1 \end{cases}$
for $k = 0, 1, 2, 3, \ldots, \mathcal{T}_{total} - 1$ **do**
  Let $\eta^{(k)} \in \mathbb{R}^m$ where for all $i \in [m]$ we let $\eta_i^{(k)} = \begin{cases} \frac{1}{3\bar{\alpha}} & \text{if } \rho_i(w^{(k)}) \geq 1 \\ \frac{1}{3\bar{\alpha}}\beta & \text{if } \rho_i(w^{(k)}) < 1 \end{cases}$
  $w^{(k+1)} \leftarrow$ **Descent**$(w^{(k)}, [m], \eta^{(k)})$ ▷ See (2.2) and Lemma 2.2
end
Return $\widehat{w} \in \mathbb{R}^m_{>0}$, where $\widehat{w}_i = (a_i^\top (\mathbf{A}^\top \mathbf{W}^{(\mathcal{T}_{\text{total}})} \mathbf{A})^{-1} a_i)^{1/\alpha}$. ▷ See Section B

THEOREM 5.1. (MAIN THEOREM (ONE-STEP PARALLEL ALGORITHM)) *Given a full rank matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $p \geq 4$, we can compute ε-approximate Lewis weights (1.5) in $O(p^3 \log(mp/\varepsilon)$ iterations. Each iteration computes the leverage score of one row of $\mathbf{DA}$ for some diagonal matrix $\mathbf{D}$. The total runtime is $O(p^3 mn^2 \log(mp/\varepsilon))$.*

We first spell out the key idea of the proof of Theorem 5.1 in Lemma 5.1 next, which is that (2.3) is maintained in every iteration through the use of varying step sizes, *without explicitly invoking rounding procedures.* Since (2.3) always holds, we may use Lemma 2.5 in bounding the iteration complexity.

LEMMA 5.1. (ROUNDING CONDITION INVARIANCE) *For any iteration $k \in [\mathcal{T}_{total} - 2]$ in Algorithm 4, if $\rho_{\max}(w^{(k)}) \leq 1 + \alpha$, then $\rho_{\max}(w^{(k+1)}) \leq 1 + \alpha$.*

Proof. By the definition of **Descent**(·) in (2.2) and choice of $\eta_i^{(k)}$ in Algorithm 4, we have,

$$w_i^{(k+1)} = w_i^{(k)} \cdot \left[1 + \eta_i^{(k)} \left(\frac{\rho_i(w^{(k)}) - 1}{\rho_i(w^{(k)}) + 1}\right)\right] \tag{5.1}$$

$$\geq w_i^{(k)}(1 - \eta_i^{(k)}) \geq w_i^{(k)} \left(1 - \frac{\beta}{3\bar{\alpha}}\right). \tag{5.2}$$

Applying this inequality to the definition of $\rho(w)$ in (2.1), for all $i \in [m]$, we have

$$\rho_i(w^{(k+1)}) = \left[\frac{w_i^{(k+1)}}{w_i^{(k)}}\right]^{-\alpha} \frac{1}{[w_i^{(k)}]^\alpha} a_i^\top (\mathbf{A}^\top \mathbf{W}^{(k+1)} \mathbf{A})^{-1} a_i \leq \left(1 - \frac{\beta}{3\bar{\alpha}}\right)^{-1} \left[\frac{w_i^{(k+1)}}{w_i^{(k)}}\right]^{-\alpha} \rho_i(w^{(k)}). \tag{5.3}$$

Plugging (5.2) into (5.3) when $\rho_i(w^{(k)}) \leq 1$ and using the upper bound on β yields that

$$\rho_i(w^{(k+1)}) \leq \left(1 - \frac{\beta}{3\bar{\alpha}}\right)^{-(1+\alpha)} \leq 1 + \alpha \ .$$

If $\rho_i(w^{(k)}) \geq 1$, then (5.3), the equality in (5.2), the bound on β, and $\rho_i(w^{(k)}) \leq 1+\alpha$ imply that

$$\rho_i(w^{(k+1)}) \leq \left(1 - \frac{\beta}{3\bar{\alpha}}\right)^{-1} \left[1 + \frac{1}{3\bar{\alpha}}\left(\frac{\rho_i(w^{(k)})-1}{\rho_i(w^{(k)})+1}\right)\right]^{-\alpha} \rho_i(w^{(k)}) \leq 1+\alpha.$$

□

Proof. [Proof of Theorem 5.1] By our choice of $w_i^{(0)} = 1$ for all $i \in [m]$, we have that $\rho_i(w^{(0)}) = \sigma_i(w^{(0)}) \leq 1$ by Fact 1.1. Then applying Lemma 5.1 yields by induction that $\rho_{\max}(w^{(k)}) \leq 1+\alpha$ at every iteration k. We may now therefore upper bound the objective sub-optimality from Lemma 2.5; as before, combining this with the lower bound on progress from Lemma 2.6 (noticing that $\eta_i \geq \frac{\beta}{3\bar{\alpha}}$) yields

$$\begin{aligned}\mathcal{F}(w^+) - \mathcal{F}(\overline{w}) &\leq \mathcal{F}(w) - \mathcal{F}(\overline{w}) - \frac{\beta}{6\bar{\alpha}} \sum_{i=1}^m w_i^{1+\alpha} \frac{(\rho_i(w)-1)^2}{\rho_i(w)+1} \\ &\leq \left(1 - \frac{\beta}{30\max(1,\alpha)\max(1,\alpha^{-1})}\right)(\mathcal{F}(w) - \mathcal{F}(\overline{w})). \end{aligned} \tag{5.4}$$

Thus, **Descent**(·) decreases $\mathcal{F}$. Using $\mathcal{F}(w) - \mathcal{F}(\overline{w}) \leq n\log(m/n) \leq m^2$ from Lemma 2.4 and setting (5.4) to $\widetilde{\varepsilon}$ gives an iteration complexity of $\mathcal{O}(\beta^{-1}\alpha^{-1}\log(m/\widetilde{\varepsilon})) = \mathcal{O}(\alpha^{-3}\log(m/\widetilde{\varepsilon}))$ if $\alpha \in (0,1]$ and $\mathcal{O}(\alpha\beta^{-1}\log(m/\widetilde{\varepsilon})) = \mathcal{O}(\alpha\log(m/\widetilde{\varepsilon}))$ otherwise. As in the proofs of Theorems 1.1 and 1.2, we can then invoke Lemma 2.1 to construct the vector that is ε-approximate to the Lewis weights. □

6 Acknowledgements

We are grateful to the anonymous reviewers of SODA 2022 for their careful reading and thoughtful comments that helped us improve our exposition. Maryam Fazel was supported in part by grants NSF TRIPODS II DMS 2023166, NSF TRIPODS CCF 1740551, and NSF CCF 2007036. Yin Tat Lee was supported in part by NSF awards CCF-1749609, DMS-1839116, DMS-2023166, CCF-2105772, a Microsoft Research Faculty Fellowship, Sloan Research Fellowship, and Packard Fellowship. Swati Padmanabhan was supported in part by NSF TRIPODS II DMS 2023166. Aaron Sidford was supported in part by a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, a PayPal research award, and a Sloan Research Fellowship.

Appendices

We start with a piece of notation we frequently use in the appendix. For a given vector $x \in \mathbb{R}^m$, we use $\mathbf{Diag}(x)$ to describe the diagonal matrix with x on its diagonal. For a matrix $\mathbf{X}$, we use $\mathbf{diag}(\mathbf{X})$ to denote the vector made up of the diagonal entries of $\mathbf{X}$. Further, recall as stated in Section 1.4, that given any vector x, we use its uppercase boldface name $\mathbf{X} \stackrel{\text{def}}{=} \mathbf{Diag}(x)$.

A Technical Proofs: Gradient, Hessian, Initial Error, Minimum Progress

Lemma 2.3. (Gradient and Hessian) *For any $w \in \mathbb{R}^m_{>0}$, the objective in (1.4), $\mathcal{F}(w) = -\log\det(\mathbf{A}^\top\mathbf{W}\mathbf{A}) + \frac{1}{1+\alpha}\mathbf{1}^\top w^{1+\alpha}$, has gradient and Hessian given by the following expressions.*

$$[\nabla\mathcal{F}(w)]_i = w_i^{-1}\cdot(w_i^{1+\alpha} - \sigma_i(w)) \text{ and } \nabla^2\mathcal{F}(w) = \mathbf{W}^{-1}\mathbf{P}(w)^{(2)}\mathbf{W}^{-1} + \alpha\mathbf{W}^{\alpha-1}.$$

Proof. The proof essentially follows by combining Lemmas 48 and 49 of [LS19]. For completeness, we provide the full proof here. Applying chain rule to the log det function and then the definition of $\rho(w)$ from (2.1) gives the claim that

$$\nabla_i\mathcal{F}(w) = -(\mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top)_{ii} + w_i^\alpha = -a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i + w_i^\alpha = \frac{-\sigma_i(w)}{w_i} + w_i^\alpha.$$

We now set some notation to compute the Hessian: let $\mathbf{M} \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top$, let $h \in \mathbb{R}^m$ be any arbitrary

vector, and let $\mathbf{H} \stackrel{\text{def}}{=} \mathbf{Diag}(h)$. For $f : \mathbb{R}^n \to \mathbb{R}$ and for $x, h \in \mathbb{R}^n$ we let $\mathcal{D}_x f(x)[h]$ denote the directional derivative of f at x in the direction h, i.e., $\mathcal{D}_x f(x)[h] = \lim_{t\to 0}(f(x+th) - f(x))/t$. Then we have,

$$\begin{aligned}
\mathcal{D}_w \langle h, -\mathbf{Diag}(\mathbf{A}(\mathbf{A}^\top \mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top)\rangle[h] &= \langle h, -\mathbf{Diag}(\mathbf{A}\mathcal{D}_w(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}[h]\mathbf{A}^\top)\rangle \\
&= \langle h, \mathbf{Diag}(\mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathcal{D}_w(\mathbf{A}^\top\mathbf{W}\mathbf{A})[h]\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top)\rangle \\
&= \langle h, \mathbf{Diag}(\mathbf{M}\mathbf{H}\mathbf{M})\rangle \\
&= \sum_{i,j} h_i h_j \mathbf{M}_{ij}\mathbf{M}_{ji} = \sum_{i,j} h_i h_j \mathbf{M}_{ij}^2,
\end{aligned}$$

where the last step follows by symmetry of $\mathbf{M}$. This implies

$$\nabla^2_{ij}\mathcal{F}(w) = \begin{cases} (a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_j)^2 & \text{if } i \neq j \\ (a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_j)^2 + \alpha w_i^{\alpha-1} & \textit{otherwise} \end{cases},$$

which, in shorthand, is $\nabla^2\mathcal{F}(w) = \mathbf{M}\circ\mathbf{M} + \alpha\mathbf{W}^{\alpha-1}$. We may express this Hessian as in the statement of the lemma by writing $\mathbf{M}$ in terms of $\mathbf{P}(w)$. ∎

LEMMA 2.4. (INITIAL SUB-OPTIMALITY) *At the start of Algorithm 1, the value of the objective of (1.4) differs from the optimum objective value as* $\mathcal{F}(w^{(0)}) \leq \mathcal{F}(\overline{w}) + n\log(m/n)$.

Proof. We study the two terms constituting the objective in (1.4). First, by choice of $w^{(0)} = \frac{n}{m}\mathbf{1}$, we have

$$-\log\det\Big(\mathbf{A}^\top\mathbf{W}^{(0)}\mathbf{A}\Big) = -\log\det\big((n/m)\mathbf{A}^\top\mathbf{A}\big). \tag{A.1}$$

Next, since leverage scores always lie between zero and one, the optimality condition for (1.4), $\sigma(\overline{w}) = (\overline{w})^{1+\alpha}$, implies $\overline{w} \leq 1$, which in turn gives $\overline{\mathbf{W}} \preceq I$. This implies $\mathbf{A}^\top\overline{\mathbf{W}}\mathbf{A} \preceq \mathbf{A}^\top\mathbf{A}$. Therefore,

$$-\log\det\big(\mathbf{A}^\top\mathbf{A}\big) \leq -\log\det\big(\mathbf{A}^\top\overline{\mathbf{W}}\mathbf{A}\big). \tag{A.2}$$

Combining (A.1) and (A.2) gives

$$-\log\det\Big(\mathbf{A}^\top\mathbf{W}^{(0)}\mathbf{A}\Big) \leq -\log\det\big(\mathbf{A}^\top\overline{\mathbf{W}}\mathbf{A}\big) + n\log(m/n). \tag{A.3}$$

Next, observe that $\mathbf{1}^\top(w^{(0)})^{1+\alpha} = m\cdot(n/m)^{1+\alpha}$, and $\mathbf{1}^\top(\overline{w})^{1+\alpha} = \sum_{i=1}^m \sigma_i(\overline{w}) = n$, where we invoked Fact 1.1. By now applying $m \geq n$, we get

$$\mathbf{1}^\top(w^{(0)})^{1+\alpha} \leq \mathbf{1}^\top(\overline{w})^{1+\alpha}. \tag{A.4}$$

Combining (A.3), (A.4), and the definition of the objective (1.4) finishes the claim. ∎

LEMMA 2.6. (FUNCTION DECREASE IN **Descent**$(\cdot)$) *Let* $w, \eta \in \mathbb{R}^m_{>0}$ *with* $\eta_i \in [0, \frac{1}{3\bar{\alpha}}]$ *for all* $i \in [m]$. *Further, let* $w^+ = \textbf{\textit{Descent}}(w, [m], \eta)$, *where* ***Descent*** *is defined in (2.2). Then,* $w^+ \in \mathbb{R}^m_{>0}$ *with the following decrease in function objective.*

$$\mathcal{F}(w^+) \leq \mathcal{F}(w) - \sum_{i\in[m]} \frac{\eta_i}{2}\cdot w_i^{1+\alpha}\cdot\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1}.$$

Proof. By the remainder form of Taylor's theorem, for some $t \in [0,1]$ and $\widetilde{w} = tw + (1-t)w^+$

$$\mathcal{F}(w^+) = \mathcal{F}(w) + \langle\nabla\mathcal{F}(w), w^+ - w\rangle + \frac{1}{2}(w^+-w)^\top\nabla^2\mathcal{F}(\widetilde{w})(w^+-w). \tag{A.5}$$

We prove the result by bounding the quadratic form of $\nabla^2\mathcal{F}(\widetilde{w})$ from above and leveraging the structure of w^+ and $\nabla\mathcal{F}(w)$. Lemma 2.3 and Fact 1.1 imply that

$$\nabla^2\mathcal{F}(\widetilde{w}) = \widetilde{\mathbf{W}}^{-1}\mathbf{P}(\widetilde{w})^{(2)}\widetilde{\mathbf{W}}^{-1} + \alpha\widetilde{\mathbf{W}}^{\alpha-1} \preceq \widetilde{\mathbf{W}}^{-1}\Sigma(\widetilde{w})\widetilde{\mathbf{W}}^{-1} + \alpha\widetilde{\mathbf{W}}^{\alpha-1}\ . \tag{A.6}$$

Further, the positivity of w_i and $\sigma_i(w)$ and the non-negativity of η and ρ imply that $(1-\|\eta\|_\infty)w_i \le w_i^+ \le (1+\|\eta\|_\infty)w_i$ for all $i \in [m]$. Since $\|\eta\|_\infty \le \frac{1}{3\bar{\alpha}}$, this implies that

$$(1-\tfrac{1}{3\bar{\alpha}})w_i \le \widetilde{w}_i \le (1+\tfrac{1}{3\bar{\alpha}})w_i \quad \text{for all} \quad i \in [m] \ .$$

Consequently, for all $i \in [m]$, we bound the first term of (A.6) as

$$\begin{aligned}\left[\widetilde{\mathbf{W}}^{-1}\Sigma(\widetilde{w})\widetilde{\mathbf{W}}^{-1}\right]_{ii} &= e_i^\top \widetilde{\mathbf{W}}^{-1/2}\mathbf{A}(\mathbf{A}^\top\widetilde{\mathbf{W}}\mathbf{A})^{-1}\mathbf{A}^\top\widetilde{\mathbf{W}}^{-1/2}e_i = \frac{1}{\widetilde{w}_i}a_i^\top(\mathbf{A}^\top\widetilde{\mathbf{W}}\mathbf{A})^{-1}a_i \\ &\le (1-\tfrac{1}{3\bar{\alpha}})^{-1}\frac{1}{w_i}a_i^\top(\mathbf{A}^\top\widetilde{\mathbf{W}}\mathbf{A})^{-1}a_i \le (1-\tfrac{1}{3\bar{\alpha}})^{-2}\frac{1}{w_i}a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i \\ &= (1-\tfrac{1}{3\bar{\alpha}})^{-2}\left[\mathbf{W}^{-1}\Sigma(w)\mathbf{W}^{-1}\right]_{ii} \preceq 3\left[\mathbf{W}^{-1}\Sigma(w)\mathbf{W}^{-1}\right]_{ii}\end{aligned} \tag{A.7}$$

Further, when $\alpha \in (0,1]$, we bound the second term of (A.6) as

$$\widetilde{\mathbf{W}}^{\alpha-1} \preceq (1-\tfrac{1}{3\bar{\alpha}})^{\alpha-1}\mathbf{W}^{\alpha-1} \preceq (1-\tfrac{1}{3\bar{\alpha}})^{-1}\mathbf{W}^{\alpha-1} \preceq 3\mathbf{W}^{\alpha-1}, \tag{A.8}$$

and when $\alpha \ge 1$, we have

$$\widetilde{\mathbf{W}}^{\alpha-1} \preceq (1+\tfrac{1}{3\bar{\alpha}})^{\alpha-1}\mathbf{W}^{\alpha-1} \preceq \exp\left(\frac{\alpha-1}{3\bar{\alpha}}\right)\mathbf{W}^{\alpha-1} = \exp\left(\frac{\alpha-1}{3\alpha}\right)\mathbf{W}^{\alpha-1} \preceq 3\mathbf{W}^{\alpha-1}. \tag{A.9}$$

Using (A.7), (A.8), and (A.9) in (A.6), we have that in all cases

$$\nabla^2\mathcal{F}(\widetilde{w}) \preceq 3\left[\mathbf{W}^{-1}\Sigma(w)\mathbf{W}^{-1} + \alpha\mathbf{W}^{\alpha-1}\right] \preceq 3\bar{\alpha}\mathbf{W}^{-1}\left[\Sigma(w) + \mathbf{W}^{1+\alpha}\right]\mathbf{W}^{-1} \ .$$

Applying to the above Loewner inequality the definition of w^+ gives

$$\begin{aligned}(w^+-w)^\top\nabla^2\mathcal{F}(\widetilde{w})(w^+-w) &\le \sum_{i\in[m]} 3\bar{\alpha}\cdot(w_i^{1+\alpha}+\sigma_i(w))\cdot\left(\eta_i\cdot\frac{\rho_i(w)-1}{\rho_i(w)+1}\right)^2 \\ &= \sum_{i\in[m]} 3\bar{\alpha}\cdot\eta_i^2\cdot w_i^{1+\alpha}\cdot\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1} \ .\end{aligned} \tag{A.10}$$

Next, recall that by Lemma 2.3, $[\nabla\mathcal{F}(w)]_i = w_i^{-1}\cdot(w_i^{1+\alpha}-\sigma_i(w))$ for all $i \in [m]$. Consequently,

$$\langle\nabla\mathcal{F}(w), w^+-w\rangle = \sum_{i\in[m]}(w_i^{1+\alpha}-\sigma_i(w))\cdot\left(\eta_i\cdot\frac{\rho_i(w)-1}{\rho_i(w)+1}\right) = -\sum_{i\in[m]}\eta_i\cdot w_i^{1+\alpha}\cdot\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1} \ . \tag{A.11}$$

Combining (A.5), (A.10), and (A.11) yields that

$$\mathcal{F}(w^+) \le \mathcal{F}(w) + \sum_{i\in[m]}\left(-\eta_i + \frac{3\bar{\alpha}\eta_i^2}{2}\right)\cdot w_i^{1+\alpha}\cdot\frac{(\rho_i(w)-1)^2}{\rho_i(w)+1} \ .$$

The result follows by plugging in $\eta_i \in [0, (3\bar{\alpha})^{-1}]$, as assumed. ∎

B From Optimization Problem to Lewis Weights

The goal of this section is to prove how to obtain ε-approximate Lewis weights from an $\widetilde{\varepsilon}$-approximate solution to the problem in (1.4). Our proof strategy is to first utilize the fact that the vector w_{R} obtained after the rounding step following the `for` loop of Algorithm 1 satisfies the properties of being $\widetilde{\varepsilon}$-suboptimal (additively) and also the rounding condition (2.3). In Lemma 2.1, the $\widetilde{\varepsilon}$-suboptimality is used to show a bound on $\|\sigma(w_{\mathrm{R}}) - w_{\mathrm{R}}^{1+\alpha}\|_\infty$. Coupled with the rounding condition, we then show in Lemma B.1 that $\widehat{w_{\mathrm{R}}}$ constructed as per the last line of Algorithm 1 then satisfies approximate optimality, $\sigma(\widehat{w}) \approx_\delta \widehat{w}^{1+\alpha}$, for some small $\delta > 0$. In Lemma B.2, we finally relate this approximate optimality to coordinate-wise multiplicative closeness between $\widehat{w}$ and the vector of

true Lewis weights. Finally, in Lemma 2.1, we pick the appropriate approximation factors for each of the lemmas invoked and prove the desired approximation. Since the vector $w^{\mathcal{T}_{\text{total}}}$ obtained at the end of the `for` loop of Algorithm 4 also satisfies the aforementioned properties of w_{R}, the same set of lemmas apply to Algorithm 4 as well. We begin with some technical lemmas.

B.1 From Approximate Closeness to Approximate Optimality.

LEMMA B.1. *Let $w \in \mathbb{R}^m_{>0}$ such that $\|\sigma(w) - w^{1+\alpha}\|_\infty \leq \bar{\varepsilon}$ for some parameter $0 < \bar{\varepsilon} \leq \frac{1}{100m^2(\alpha+\alpha^{-1})^2}$ and also let $\rho_{\max}(w) \leq 1+\alpha$. Define $\widehat{w}_i = (a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i)^{1/\alpha}$. Then, for the parameter $\delta = 20\sqrt{\bar{\varepsilon}}m(\alpha+\alpha^{-1})$, we have that $\sigma(\widehat{w}) \approx_\delta \widehat{w}^{1+\alpha}$.*

Proof. Our strategy to prove $\sigma(\widehat{w}) \approx_\delta \widehat{w}^{1+\alpha}$ involves first noting that this is the same as proving $\widehat{w}^{-1} \cdot \sigma(\widehat{w}) \approx_\delta \widehat{w}^{\alpha}$ and, from the definition of $\widehat{w}$ in the statement of the lemma, to instead prove $\mathbf{A}^\top\widehat{\mathbf{W}}\mathbf{A} \approx_\delta \mathbf{A}^\top\mathbf{W}\mathbf{A}$.

To this end, we split $\mathbf{W}$ into two matrices based on the size of its coordinates, setting the following notation. Define $\mathbf{W}_{w\leq\eta}$ to be the diagonal matrix $\mathbf{W}$ with zeroes at indices corresponding to $w > \eta$, and $\widehat{\mathbf{W}}_{w\leq\eta}$ to be the diagonal matrix $\widehat{\mathbf{W}}$ with zeroes at indices corresponding to $w > \eta$. We first show that $\mathbf{A}^\top\widehat{\mathbf{W}}_{w\leq\eta}\mathbf{A}$ and $\mathbf{A}^\top\mathbf{W}_{w\leq\eta}\mathbf{A}$ are small compared to $\mathbf{A}^\top\mathbf{W}\mathbf{A}$ and can therefore be ignored in the preceding desired approximation. We then prove that for $w > \eta$, we have $w \approx_\delta \widehat{w}$. This proof technique is inspired by Lemma 4 of [Vai89].

First, we prove that $\mathbf{A}^\top\widehat{\mathbf{W}}_{w\leq\eta}\mathbf{A}$ is small as compared to $\mathbf{A}^\top\mathbf{W}_{w>\eta}\mathbf{A}$. Since (2.3) is satisfied, it means

$$a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i = \sigma_i(w) \cdot w_i^{-1} \leq (1+\alpha)w_i^\alpha.$$

Combining this with the definition of $\widehat{w}_i$ as in the statement of the lemma, we may use non-negativity of α to derive

$$\widehat{w}_i \leq (1+\alpha)^{1/\alpha}w_i \leq 3w_i. \tag{B.1}$$

We apply this inequality in the following expression to obtain

$$\begin{aligned}
\mathrm{Tr}\Big((\mathbf{A}^\top\widehat{\mathbf{W}}_{w\leq\eta}\mathbf{A})(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\Big) &= \sum_{w_i\leq\eta}\widehat{w}_i(a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i) \\
&= \sum_{w_i\leq\eta}(a_i^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_i)^{1+1/\alpha} \\
&\leq (1+\alpha)^{1+1/\alpha}\sum_{w_i\leq\eta}w_i^{1+\alpha} \\
&\leq 3(1+\alpha)m\eta^{1+\alpha}. \qquad \text{(B.2)}
\end{aligned}$$

This implies that[7]

$$\mathbf{A}^\top\widehat{\mathbf{W}}_{w\leq\eta}\mathbf{A} \preceq 3(1+\alpha)m\eta^{1+\alpha}\mathbf{A}^\top\mathbf{W}\mathbf{A}. \tag{B.3}$$

Our next goal is to bound $\mathbf{A}^\top\widehat{\mathbf{W}}_{w>\eta}\mathbf{A}$ in terms of $\mathbf{A}^\top\mathbf{W}\mathbf{A}$, which we do by first bounding it in terms of $\mathbf{A}^\top\mathbf{W}_{w>\eta}\mathbf{A}$ and then bounding $\mathbf{A}^\top\mathbf{W}_{w>\eta}\mathbf{A}$ in terms of $\mathbf{A}^\top\mathbf{W}\mathbf{A}$. By definition, $\widehat{w}_i^\alpha = \sigma_i(w) \cdot w_i^{-1}$. Further, by assumption, $\|\sigma(w) - w^{1+\alpha}\|_\infty \leq \bar{\varepsilon}$. Therefore, for any $w_i \geq \eta$

$$\widehat{w}_i^\alpha \leq (w_i^{1+\alpha} + \bar{\varepsilon}) \cdot w_i^{-1} \leq (1 + \bar{\varepsilon}/\eta^{1+\alpha})w_i^{1+\alpha} \cdot w_i^{-1} = (1+\bar{\varepsilon}/\eta^{1+\alpha})w_i^\alpha,$$

and

$$\widehat{w}_i^\alpha \geq (w_i^{1+\alpha} - \bar{\varepsilon}) \cdot w_i^{-1} \geq (1 - \bar{\varepsilon}/\eta^{1+\alpha})w_i^{1+\alpha} \cdot w_i^{-1} = (1-\bar{\varepsilon}/\eta^{1+\alpha})w_i^\alpha.$$

[7] Given $X, Y \succeq 0$, we have $Y^{1/2}XY^{1/2} \succeq 0$. Then, if $\mathrm{Tr}(XY) \leq 1$, we have $\mathrm{Tr}\big(Y^{1/2}XY^{1/2}\big) \leq 1$, and combining these with the previous matrix inequality, we conclude that $Y^{1/2}XY^{1/2} \preceq I$, which implies that $X \preceq Y^{-1}$.

By our choice of $\overline{\varepsilon}$, for $w_i \geq \eta$, we have

$$\left(1 - \frac{2\overline{\varepsilon}}{\alpha\eta^{1+\alpha}}\right) w_i \leq \widehat{w}_i \leq \left(1 + \frac{2\overline{\varepsilon}}{\alpha\eta^{1+\alpha}}\right) w_i. \tag{B.4}$$

Further, we have the following inequality:

$$\mathbf{A}^\top \mathbf{W}_{w>\eta} \mathbf{A} \preceq \mathbf{A}^\top \mathbf{W} \mathbf{A}. \tag{B.5}$$

Hence, we can combine (B.5), (B.4), and (B.3) to see that

$$\begin{aligned}
\mathbf{A}^\top \widehat{\mathbf{W}} \mathbf{A} &= \mathbf{A}^\top \widehat{\mathbf{W}}_{w>\eta} \mathbf{A} + \mathbf{A}^\top \widehat{\mathbf{W}}_{w\leq\eta} \mathbf{A} \\
&\preceq \left(1 + \frac{2\overline{\varepsilon}}{\alpha\eta^{1+\alpha}}\right) \mathbf{A}^\top \mathbf{W}_{w>\eta} \mathbf{A} + 3(1+\alpha) m\eta^{1+\alpha} \mathbf{A}^\top \mathbf{W} \mathbf{A} \\
&\preceq \mathbf{A}^\top \mathbf{W} \mathbf{A} \left(1 + \frac{2\overline{\varepsilon}}{\alpha\eta^{1+\alpha}} + 3(1+\alpha) m\eta^{1+\alpha}\right).
\end{aligned}$$

Set $\eta^{1+\alpha} = \sqrt{\overline{\varepsilon}}$ for the upper bound.

For the lower bound, we bound $\mathbf{A}^\top \mathbf{W}_{w\leq\eta} \mathbf{A}$ and, therefore, also $\mathbf{A}^\top \mathbf{W}_{w>\eta} \mathbf{A}$. Observe that

$$\begin{aligned}
\operatorname{Tr}\big((\mathbf{A}^\top \mathbf{W}_{w\leq\eta} \mathbf{A})(\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1}\big) &= \sum_{w_i \leq \eta} w_i a_i^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_i = \sum_{w_i \leq \eta} \sigma_i(w) \\
&\leq \sum_{w_i \leq \eta} (w_i^{1+\alpha} + \overline{\varepsilon}) \leq m(\eta^{1+\alpha} + \overline{\varepsilon}),
\end{aligned}$$

where the second step is by $\|\sigma(w) - w^{1+\alpha}\|_\infty \leq \overline{\varepsilon}$, as assumed in the lemma. This implies that

$$\mathbf{A}^\top \mathbf{W}_{w\leq\eta} \mathbf{A} \preceq m(\eta^{1+\alpha} + \overline{\varepsilon}) \mathbf{A}^\top \mathbf{W} \mathbf{A},$$

and therefore that

$$\mathbf{A}^\top \mathbf{W}_{w>\eta} \mathbf{A} \succeq (1 - m(\eta^{1+\alpha} + \overline{\varepsilon})) \mathbf{A}^\top \mathbf{W} \mathbf{A}.$$

Repeating the method for the upper bound then finishes the proof. ∎

B.2 From Approximate Optimality to Approximate Lewis Weights. In this section, we go from the previous notion of approximation to the one we finally seek in (1.5). Specifically, we show that if $\sigma(w) \approx_\beta w^{1+\alpha}$, then $w \approx_{O((\beta/\alpha)\sqrt{n})} \overline{w}$. To prove this, we first give a technical result. We recall notation stated in Section 1.4: for any projection matrix $\mathbf{P}(w) \in \mathbb{R}^{m\times m}$, we have the vector of leverage scores $\sigma(w) = \mathbf{diag}(\mathbf{P}(w))$.

Claim 1. *For any projection matrix $\mathbf{P}(w) \in \mathbb{R}^{m\times m}$, $\alpha \geq 0$, and vector $x \in \mathbb{R}^m$, we have that*

$$\left\| \left[\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w)\right]^{-1} \mathbf{\Sigma}(w) x \right\|_\infty \leq \frac{1}{\alpha}\|x\|_\infty + \frac{1}{\alpha^2}\|x\|_{\mathbf{\Sigma}(w)} \leq \left(\frac{1+\sqrt{n}/\alpha}{\alpha}\right)\|x\|_\infty$$

Proof. Let $y \stackrel{\text{def}}{=} \left[\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w)\right]^{-1} \mathbf{\Sigma}(w) x$. Since $\mathbf{0} \preceq \mathbf{P}(w)^{(2)} \preceq \mathbf{\Sigma}(w)$ (Fact 1.1), we have that $\mathbf{\Sigma}(w) \preceq \frac{1}{\alpha}\left[\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w)\right]$ and $(\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w))^{-1} \preceq \alpha^{-1}\mathbf{\Sigma}(w)^{-1}$. Consequently, taking norms in terms of these matrices gives

$$\|y\|_{\mathbf{\Sigma}(w)} = \left\| \left[\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w)\right]^{-1} \mathbf{\Sigma}(w) x \right\|_{\mathbf{\Sigma}(w)} \leq \frac{1}{\sqrt{\alpha}} \|\mathbf{\Sigma}(w)x\|_{\left[\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w)\right]^{-1}} \leq \frac{1}{\alpha}\|x\|_{\mathbf{\Sigma}(w)}. \tag{B.6}$$

Next, since by Lemma 47 of [LS19], $\left\|\mathbf{\Sigma}(w)^{-1}\mathbf{P}(w)^{(2)} z\right\|_\infty \leq \|z\|_{\mathbf{\Sigma}(w)}$ for all $z \in \mathbb{R}^m$, we see that $\left|[\mathbf{P}(w)^{(2)} y]_i\right| \leq \sigma_i(w)\|y\|_{\mathbf{\Sigma}(w)}$ for all $i \in [m]$, and since by definition of y, we have $[(\mathbf{P}(w)^{(2)} + \alpha\mathbf{\Sigma}(w))y]_i = \sigma_i(w)x_i$ for all $i \in [m]$,

we have that

$$\|y\|_\infty = \max_{i\in[m]} |y_i| = \max_{i\in[m]} \left|\frac{1}{\alpha}x_i + \frac{1}{\alpha\sigma_i(w)}\left[\mathbf{P}(w)^{(2)}y\right]_i\right| \le \frac{1}{\alpha}\|x\|_\infty + \frac{1}{\alpha}\|y\|_{\mathbf{\Sigma}(w)}\,. \tag{B.7}$$

Combining (B.6) and (B.7) and using that $\sum_{i\in[m]}\sigma_i(w) \le n$ yields the claim. ∎

LEMMA B.2. *Let $\widehat{w} \in \mathbb{R}^m_{>0}$ be a vector that satisfies approximate optimality of (1.4) in the following sense:*

$$\sigma(\widehat{w}) = \widehat{\mathbf{W}}^{1+\alpha}v, \text{ for } \exp(-\mu)\mathbf{1} \le v \le \exp(\mu)\mathbf{1}.$$

Then, $\widehat{w}$ is also coordinate-wise multiplicatively close to $\overline{w}$, the true vector of Lewis weights, as formalized below.

$$\exp\left(-\frac{1}{\alpha}(1+\sqrt{n}/\alpha)\mu\right)\overline{w} \le \widehat{w} \le \exp\left(\frac{1}{\alpha}(1+\sqrt{n}/\alpha)\mu\right)\overline{w}\,.$$

Proof. For all $t\in[0,1]$, let $[v_t]_i = [v_i^t]$ so that $v_1 = v$ and $v_0 = \mathbf{1}$. Further, for all $t\in[0,1]$, let w_t be the unique solution to

$$w_t = \operatorname*{argmin}_{w\in\mathbb{R}^m_{>0}} f_t(w) \stackrel{\text{def}}{=} -\log\det\left(\mathbf{A}^\top\mathbf{W}\mathbf{A}\right) + \frac{1}{1+\alpha}\sum_{i\in[m]}[v_t]_i w_i^{1+\alpha}. \tag{B.8}$$

Then we have the following gradients.

$$\nabla_w f_t(w) = -\mathbf{W}^{-1}\sigma(w) + \mathbf{W}^\alpha v_t\,,$$

$$\nabla_w(\frac{d}{dt}f_t)(w) = \mathbf{W}^\alpha\frac{d}{dt}v_t = \mathbf{W}^\alpha v_t\ln(v) \tag{B.9}$$

$$\nabla^2_{ww} f_t(w) = \mathbf{W}^{-1}\left[\mathbf{P}(w)^{(2)} + \alpha\mathbf{W}^{1+\alpha}\mathbf{V}\right]\mathbf{W}^{-1}\,. \tag{B.10}$$

Consequently, by optimality of w_t as defined in (B.8), we have $\mathbf{0} = \nabla_w f_t(w_t) = -\mathbf{W}_t^{-1}\sigma(w_t) + \mathbf{W}_t^\alpha v_t$. Rearranging the terms of this equation yields that

$$\sigma(w_t) = \mathbf{W}_t^{1+\alpha}v_t, \tag{B.11}$$

and therefore $w_1 = \widehat{w}$ and $w_0 = \overline{w}$. To prove the lemma, it therefore suffices to bound

$$\ln(\widehat{w}/\overline{w}) = \ln(w_1/w_0) = \int_{t=0}^1\left[\frac{d}{dt}\ln(w_t)\right]dt = \int_{t=0}^1 \mathbf{W}_t^{-1}\left[\frac{d}{dt}w_t\right]dt\,. \tag{B.12}$$

To bound (B.12), it remains to compute $\frac{d}{dt}w_t$ and apply Claim 1. To do this, note that

$$\mathbf{0} = \frac{d}{dt}\mathbf{\nabla}_w\left[f_t(w_t)\right] = \nabla_w(\frac{d}{dt}f_t)(w_t) + \nabla^2_{ww}f_t(w_t)\cdot\frac{d}{dt}w_t\,.$$

Using that $\mathbf{P}(w_t)^{(2)} + \mathbf{W}_t^{1+\alpha}\mathbf{V}_t \succ \mathbf{0}$, we have, by rearranging the above equation and applying (B.9) and (B.10) that

$$\frac{d}{dt}w_t = -\left[\nabla^2_{ww}f_t(w_t)\right]^{-1}\cdot\left[\nabla_w(\frac{d}{dt}f_t)(w_t)\right] = -\mathbf{W}_t\left[\mathbf{P}(w_t)^{(2)} + \alpha\mathbf{W}_t^{1+\alpha}\mathbf{V}_t\right]^{-1}\mathbf{W}_t^{1+\alpha}v_t\ln(v)\,. \tag{B.13}$$

Applying (B.11) to (B.13), we have that

$$\mathbf{W}_t^{-1}\left[\frac{d}{dt}w_t\right] = -\left[\mathbf{P}(w_t)^{(2)} + \alpha\mathbf{\Sigma}(w_t)\right]^{-1}\mathbf{\Sigma}(w_t)\ln(v)\,.$$

Applying Claim 1 to the above equality, substituting in (B.12) and $\|\ln(v)\|_\infty \le \mu$ therefore yields

$$\|\ln(\widehat{w}/\overline{w})\|_\infty = \|\ln(w_1/w_0)\|_\infty \le \int_{t=0}^{1} \left\| \mathbf{W}_t^{-1} \left[\frac{d}{dt} w_t \right] \right\|_\infty dt \le \int_{t=0}^{1} \left(\frac{1+\sqrt{n}/\alpha}{\alpha} \right) \mu dt \,.$$

∎

B.3 From Optimization Problem to Approximate Lewis Weights.

LEMMA 2.1. (LEWIS WEIGHTS FROM OPTIMIZATION SOLUTION) *Let $w \in \mathbb{R}^m_{>0}$ be a vector at which the objective (1.4) is $\widetilde{\varepsilon}$-suboptimal in the additive sense for $\widetilde{\varepsilon} = \frac{\alpha^8 \varepsilon^4}{(25m(\sqrt{n}+\alpha)(\alpha+\alpha^{-1}))^4}$, i.e., $\mathcal{F}(\overline{w}) \le \mathcal{F}(w) \le \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$. Further assume that w satisfies the rounding condition: $\rho_{\max}(w) \le 1+\alpha$. Then, the vector $\widehat{w}$ defined as $\widehat{w}_i = (a_i^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} a_i)^{1/\alpha}$ satisfies $\widehat{w}_i \approx_\varepsilon \overline{w}_i$ for all $i \in [m]$, thus achieving the goal spelt out in (1.5).*

Proof. We are given a vector $w \in \mathbb{R}^m$ satisfying $\mathcal{F}(\overline{w}) \le \mathcal{F}(w) \le \mathcal{F}(\overline{w}) + \widetilde{\varepsilon}$. Then by Lemma 2.5, we have that $\frac{(\sigma_i(w) - w_i^{1+\alpha})^2}{\sigma_i(w) + w_i^{1+\alpha}} \le \widetilde{\varepsilon}$ for each $i \in [m]$. This bound implies that $w_i \le 3$ for all i because, if not, then because of $\sigma_i(w) \in [0,1]$ and the decreasing nature of $(x-a)^2/(x+a)$ over $x \in [0,1]$ for a fixed $a \ge 3$, we obtain $\frac{(\sigma_i(w) - w_i^{1+\alpha})^2}{\sigma_i(w) + w_i^{1+\alpha}} \ge \frac{(1 - w_i^{1+\alpha})^2}{1 + w_i^{1+\alpha}} \ge 1$, a contradiction. Therefore $\|\sigma(w) - w^{1+\alpha}\|_\infty \le 2\sqrt{\widetilde{\varepsilon}}$. Coupled with the provided guarantee $\rho_{\max}(w) \le 1+\alpha$, we see that the requirements of Lemma B.1 are met with $\overline{\varepsilon} = 2\sqrt{\widetilde{\varepsilon}}$, for $\widetilde{\varepsilon} \stackrel{\text{def}}{=} \frac{\widehat{\varepsilon}^4}{(25m(\alpha+\alpha^{-1}))^4}$, and Algorithm 1 therefore guarantees a $\widehat{w}$ satisfying $\sigma(\widehat{w}) \approx_{\widehat{\varepsilon}} \widehat{w}^{1+\alpha}$. Therefore, we can now apply Lemma B.2 with $\mu = \widehat{\varepsilon}$, and choosing $\widehat{\varepsilon} = \frac{\alpha^2}{\alpha+\sqrt{n}}\varepsilon$ lets us conclude that $\widehat{w}_i \approx_\varepsilon \overline{w}_i$, as claimed. ∎

C A Geometric View of Rounding

At the end of Algorithms 2 and 3, the iterate w satisfies the condition $\rho_{\max}(w) \le 1+\alpha$. We now show the geometry implied by the preceding condition, thereby provide the reason behind the terminology "rounding."

LEMMA C.1. *Given $w \in \mathbb{R}^m_{>0}$ such that $\rho_{\max}(w) \le 1+\alpha$. Define the ellipsoid $\mathcal{E}(w) := \{x : x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x \le 1\}$. Then, we have that*

$$\mathcal{E}(w) \subset \{x \in \mathbb{R}^n \mid \|\mathbf{W}^{-\alpha/2} \mathbf{A} x\|_\infty \le \sqrt{1+\alpha}\}.$$

Proof. Consider any point $x \in \mathcal{E}(w)$. Then, by Cauchy-Schwarz inequality and $\rho_{\max}(w) \le 1+\alpha$,

$$\begin{aligned}
\|\mathbf{W}^{-\alpha/2}\mathbf{A}x\|_\infty &= \max_{i\in[m]} e_i^\top \mathbf{W}^{-\alpha/2}\mathbf{A}x = \max_{i\in[m]} e_i^\top \mathbf{W}^{-\alpha/2}\mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-\frac{1}{2}}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{\frac{1}{2}}x \\
&\le \max_{i\in[m]} \sqrt{e_i^\top \mathbf{W}^{-\alpha/2}\mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{W}^{-\alpha/2}e_i}\sqrt{x^\top\mathbf{A}^\top\mathbf{W}\mathbf{A}x} \\
&\le \max_{i\in[m]} \sqrt{e_i^\top \mathbf{W}^{-\alpha/2}\mathbf{A}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{W}^{-\alpha/2}e_i} = \max_{i\in[m]} \sqrt{\frac{\sigma_i(w)}{w_i^{1+\alpha}}} \le \sqrt{1+\alpha}.
\end{aligned}$$

∎

D Explanations of Runtimes in Prior Work

The convex program (1.3) formulated by [CP15] has a variable size of n^2. Therefore, by [LSW15], the number of iterations to solve it using the cutting plane method is $O(n^2 \log(n\varepsilon^{-1})$, each iteration computing $a_i^\top \mathbf{M} a_i$ for $i \in [m]$. This can be computed by multiplying an $n \times n$ matrix with an $n \times m$ matrix, which costs between $O(mn)$ (at least the size of the larger input matrix) and $O(mn^2)$ (each entry of the resulting $m \times n$ matrix obtained by an inner product of length n vectors). Further, there is at least a total of $O(n^6)$ additional work done by the cutting plane method. This gives us a cost of at least $n^2(mn + n^4)$. The runtime of [Lee16] follows from Theorem 5.3.4.

References

[AHR08] Jacob Abernethy, Elad E Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory, COLT 2008*, 2008. 1.3

[AZLSW17] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal design of experiments via regret minimization. In *Proceedings of the 34th International Conference on Machine Learning*, 2017. 1.3

[BDM+20] Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, 2020. 1.3

[BE15] Sébastien Bubeck and Ronen Eldan. The entropic barrier: a simple and optimal universal self-concordant barrier. In *Conference on Learning Theory*, 2015. 1.3

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. 1

[CCDS20] Rachit Chhaya, Jayesh Choudhari, Anirban Dasgupta, and Supratim Shit. Streaming coresets for symmetric tensor factorization. In *International Conference on Machine Learning*, 2020. 1.3

[CCLY19] Michael B. Cohen, Ben Cousins, Yin Tat Lee, and Xin Yang. A near-optimal algorithm for approximating the john ellipsoid. In *Proceedings of the Thirty-Second Conference on Learning Theory*, 2019. 1.3

[CD21] Xue Chen and Michal Derezinski. Query complexity of least absolute deviation regression via robust uniform convergence. In *COLT*, 2021. 1.3

[CLM+15] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*. ACM, 2015. 1.3

[CP15] Michael B. Cohen and Richard Peng. Lp row sampling by lewis weights. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15. Association for Computing Machinery, 2015. (document), 1, 1, 1, 1, 1, 1.1, **??**, **??**, **??**, 1.2, 5, 1.3, D

[DAST08] S Damla Ahipasaoglu, Peng Sun, and Michael J Todd. Linear convergence of a modified frank–wolfe algorithm for computing minimum-volume enclosing ellipsoids. *Optimisation Methods and Software*, 23(1), 2008. 1.3

[DFO20] Jelena Diakonikolas, Maryam Fazel, and Lorenzo Orecchia. Fair packing and covering on a relative scale. *SIAM J. Optim.*, 30(4), 2020. 1.3

[DLS18] David Durfee, Kevin A Lai, and Saurabh Sawlani. \ell_1 regression using lewis weights preconditioning and stochastic gradient descent. In *Conference On Learning Theory*, 2018. 1.3

[DMIMW12] Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1), 2012. 1.3

[DMM06] Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Sampling algorithms for l 2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, 2006. 1, 1.3

[GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2), 1981. 1.2

[Joh48] Fritz John. Extremum problems with inequalities as subsidiary conditions, studies and essays presented to r. courant on his 60th birthday, january 8, 1948, 1948. 1

[Kha96] Leonid G Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21(2), 1996. 1.3

[KN09] Ravi Kannan and Hariharan Narayanan. Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, New York, NY, USA, 2009. Association for Computing Machinery. 1.3

[KY05] Piyush Kumar and E Alper Yildirim. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and applications*, 126(1), 2005. 1.3

[Lee16] Yin Tat Lee. *Faster Algorithms for Convex and Combinatorial Optimization*. PhD thesis, Massachusetts Institute of Technology, 2016. 1, **??**, 1.2, D

[Lew78] D Lewis. Finite dimensional subspaces of l_{p}. *Studia Mathematica*, 63(2), 1978. 1, 2

[LLV20] Aditi Laddha, Yin Tat Lee, and Santosh Vempala. Strong self-concordance and sampling. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, New York, NY, USA, 2020. Association for Computing Machinery. 1.3

[LMP13] Mu Li, Gary L Miller, and Richard Peng. Iterative row sampling. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 2013. 1.3

[LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in õ(sqrt(rank)) iterations and faster algorithms for maximum flow. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, 2014. 1, 1.3, 1.1

[LS19] Yin Tat Lee and Aaron Sidford. Solving linear programs with sqrt(rank) linear system solves. *CoRR*, abs/1910.08033, 2019. 1, 1, 1.1, **??**, 1.2, 1.3, A, B.2

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for

combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, 2015. 1, 1.2, D

[LWYZ20] Yi Li, Ruosong Wang, Lin Yang, and Hanrui Zhang. Nearly linear row sampling algorithm for quantile regression. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. 1.3

[LY18] Yin Tat Lee and Man-Chung Yue. Universal barrier is n-self-concordant. *arXiv preprint arXiv:1809.03011*, 2018. 1.3

[MSTX19] Vivek Madan, Mohit Singh, Uthaipon Tantipongpipat, and Weijun Xie. Combinatorial algorithms for optimal design. In *Proceedings of the Thirty-Second Conference on Learning Theory*, 2019. 1.3

[MSZ16] Jelena Marasevic, Clifford Stein, and Gil Zussman. A fast distributed stateless algorithm for *alpha*-fair packing problems. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, 2016. 1.3

[NN94] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994. 1.3

[PPP21] Aditya Parulekar, Advait Parulekar, and Eric Price. L1 regression with lewis weights subsampling. In *APPROX-RANDOM*, 2021. 1.3

[Puk06] Friedrich Pukelsheim. *Optimal Design of Experiments (Classics in Applied Mathematics) (Classics in Applied Mathematics, 50)*. Society for Industrial and Applied Mathematics, USA, 2006. 1.3

[SF04] Peng Sun and Robert M Freund. Computation of minimum-volume covering ellipsoids. *Operations Research*, 52(5), 2004. 1.3

[SW18] Christian Sohler and David P Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018. 1.3

[SX20] Mohit Singh and Weijun Xie. Approximation algorithms for d-optimal design. *Mathematics of Operations Research*, 45(4), 2020. 1.3

[Tod16] Michael J. Todd. *Minimum volume ellipsoids - theory and algorithms*, volume 23 of *MOS-SIAM Series on Optimization*. SIAM, 2016. 1, 1, 1.1, 1.3

[Vai89] Pravin M Vaidya. A new algorithm for minimizing convex functions over convex sets. In *30th Annual Symposium on Foundations of Computer Science*, 1989. 1, B.1

[Woj96] Przemyslaw Wojtaszczyk. *Banach spaces for analysts*. Number 25. Cambridge University Press, 1996. 1

[ZF20] Renbo Zhao and Robert M Freund. Analysis of the frank-wolfe method for logarithmically-homogeneous barriers, with an extension. *arXiv preprint arXiv:2010.08999*, 2020. 1.3