Improved Iteration Complexities for Overconstrained p-Norm Regression

Arun Jambulapati
Stanford University
Stanford, USA
jmblpati@stanford.edu

Yang P. Liu
Stanford University
Stanford, USA
yangpliu@stanford.edu

Aaron Sidford
Stanford University
Stanford, USA
sidford@stanford.edu

ABSTRACT

In this paper we obtain improved iteration complexities for solving ℓ_p regression. We provide methods which given any full-rank $\mathbf{A} \in \mathbb{R}^{n \times d}$ with $n \geq d$, $b \in \mathbb{R}^n$, and $p \geq 2$ solve $\min_{x \in \mathbb{R}^d} \|\mathbf{A}x - b\|_p$ to high precision in time dominated by that of solving $\widetilde{O}_p(d^{\frac{p-2}{3p-2}})$[1] linear systems in $\mathbf{A}^\top \mathbf{D} \mathbf{A}$ for positive diagonal matrices $\mathbf{D}$. This improves upon the previous best iteration complexity of $\widetilde{O}_p(n^{\frac{p-2}{3p-2}})$ (Adil, Kyng, Peng, Sachdeva 2019). As a corollary, we obtain an $\widetilde{O}(d^{1/3}\epsilon^{-2/3})$ iteration complexity for approximate ℓ_∞ regression. Further, for $q \in (1, 2]$ and dual norm $q = p/(p-1)$ we provide an algorithm that solves ℓ_q regression in $\widetilde{O}(d^{\frac{p-2}{2p-2}})$ iterations.

To obtain this result we analyze row reweightings (closely inspired by ℓ_p-norm Lewis weights) which allow a closer connection between ℓ_2 and ℓ_p regression. We provide adaptations of two different iterative optimization frameworks which leverage this connection and yield our results. The first framework is based on iterative refinement and multiplicative weights based width reduction and the second framework is based on highly smooth acceleration. Both approaches yield $\widetilde{O}_p(d^{\frac{p-2}{3p-2}})$ iteration methods but the second has a polynomial dependence on p (as opposed to the exponential dependence of the first algorithm) and provides a new alternative to the previous state-of-the-art methods for ℓ_p regression for large p.[2]

CCS CONCEPTS

• **Theory of computation → Convex optimization**.

KEYWORDS

Regression, Lewis Weights, Acceleration

ACM Reference Format:
Arun Jambulapati, Yang P. Liu, and Aaron Sidford. 2022. Improved Iteration Complexities for Overconstrained p-Norm Regression. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3519935.3519971

1 INTRODUCTION

In this paper, we consider the problem of solving ℓ_p *regression* for $p \in (1, \infty)$ to high precision.

Definition 1.1 (ℓ_p regression). *Given a full-rank matrix*[3] $\mathbf{A} \in \mathbb{R}^{n \times d}$, *a vector* $b \in \mathbb{R}^n$, *and a scalar* $p \geq 1$ *we say that an algorithm solves* ℓ_p *regression to* ε*-accuracy if it outputs* $y \in \mathbb{R}^n$ *satisfying*

$$\|\mathbf{A}y - b\|_p \leq (1 + \varepsilon) \min_{x \in \mathbb{R}^d} \|\mathbf{A}x - b\|_p . \tag{1}$$

We say that such an algorithm is high precision *if the runtime depends polynomially on* $\log(1/\varepsilon)$.

Beyond possible applications and utility for data analysis (see [52, 62] and references therein), the problem of ℓ_p regression is a prominent testbed for new techniques in optimization and numerical analysis. Varying p causes (1) to smoothly interpolate between least squares regression ($p = 2$), which can be solved with a single linear system solve, and linear programming ($p \in \{1, \infty\}$) [41], which is only known to be solvable to high precision with $\widetilde{O}(\sqrt{n})$ linear systems via classical interior point methods (IPMs) [57], and more recently $\widetilde{O}(\sqrt{d})$ linear systems [42].

Interestingly, although [14] showed that IPMs do not directly yield $o(\sqrt{n})$ iteration complexities for ℓ_p regression, there is a line of work [1, 3–5, 14] which obtained improved iteration complexities via alternative methods; the current state-of-the-art iteration complexity for $p \geq 2$ is $\widetilde{O}_p(n^{(p-2)/(3p-2)})$. These improvements touch on a range of advanced optimization techniques including homotopy methods, iterative refinement, high-order acceleration, and width-reduction. This line of work is closely related to work which solves approximate ℓ_∞ regression in $\widetilde{O}(n^{1/3}\varepsilon^{-O(1)})$ iterations [20] where again, improvements and simplifications have been achieved through multiple techniques [18, 32, 33]. Additionally, work on ℓ_p regression for structured graph incidence matrices $\mathbf{A}$ [1, 40] has led to improved running times for unit capacity maxflow, bipartite matching, and mincost flows [8, 38, 47].

Given this progress, a natural open problem is to bridge the gap between the known iteration complexities for ℓ_p regression and the $\tilde{O}(\sqrt{d})$ bound achievable by IPMs for ℓ_1 and ℓ_∞ regression [42] by providing an iteration complexity for ℓ_p regression that depends on d as opposed to n. Additionally, the relationship between various techniques for achieving these iteration complexities, especially acceleration and width-reduction, remains somewhat mysterious

[1] We use $\widetilde{O}_p(\cdot)$ to hide $\log^{O(1)} n$ factors and constants depending only on p. In this work, our dependence on p is at most $p^{O(p)}$ for all algorithms, and can in fact be made polynomial in most cases.
[2] Full version available at https://arxiv.org/pdf/2111.01848.pdf

[3] We assume throughout that the matrix $\mathbf{A}$ is full-rank with $n \geq d$ throughout for simplicity, and our results extend directly to the general case, for example by replacing inverses with pseudoinverses.

STOC '22, June 20–24, 2022, Rome, Italy

ACM ISBN 978-1-4503-9264-8/22/06…$15.00
https://doi.org/10.1145/3519935.3519971

(see [2] for further discussion on this relationship). Consequently, understanding the complexity of ℓ_p regression is fundamental for advancing and clarifying the power of various optimization techniques.

In this paper we take steps to address these questions and improve the complexity for solving ℓ_p regression. Our main result is a pair of algorithms, based respectively on the iterative refinement framework and width reduction techniques [3], and the Monteiro-Svaiter/highly-smooth acceleration framework [15, 18], each of which, for $p \geq 2$ solve ℓ_p regression with $\widetilde{O}_p(d^{(p-2)/(3p-2)})$ linear system solves. This improves an n to a d in the iteration dependencies for the state-of-the-art methods for ℓ_p regression.

The key notion used in our methods are reweightings of $\mathbf{A}$ closely related to Lewis weights [29, 44], which allow for a closer relationship between the ℓ_2 and ℓ_p norms induced by $\mathbf{A}$ (Lemma 2.6). We directly leverage this connection induced by ℓ_p-norm Lewis weights in the context of the optimization frameworks discussed previously to achieve our result, as opposed to previous works on (approximate) ℓ_p regression that used ℓ_p Lewis weights to construct ℓ_p-norm sparsifiers or subspace embeddings [22, 24, 25, 30, 51, 62]. As a result, these previous iteration complexities for ℓ_p regression had iteration complexities of the form $d^{\Omega(p)}$, while ours is always $\widetilde{O}(d^{1/3})$, even for large $p = \widetilde{O}(1)$.

1.1 Our Results

Here we state the main results of our paper. As is the case with several results on regression [3, 14, 18, 20, 33], the primary subroutine used by our algorithms is a linear system solver for $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for positive diagonal matrices $\mathbf{D}$. We focus on bounding the number of iterations or calls to such a linear system solver in our algorithms. Accordingly, let $\mathcal{T}_\mathbf{A}$ denote the time for solving a linear system in $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for positive diagonal matrices $\mathbf{D}$.[4] We also use "with high probability" (whp.) throughout to mean with probability at least $1 - n^{-C}$ for any constant C.

In this work, we focus on presenting iteration complexity improvements for ℓ_p regression problems. We choose to focus on iteration complexity improvements in the work as opposed to runtimes for the sake of achieving a cleaner and simpler presentation. We elaborate on this in the final paragraph in previous works (Section 1.2).

Theorem 1 (High precision ℓ_p regression for $p \geq 2$). *There is an algorithm that given any $\mathbf{A} \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$, $p \geq 2$ whp. returns an ε-accuracy solution to ℓ_p regression in time $\widetilde{O}_p(d^{\frac{p-2}{3p-2}} \cdot \mathcal{T}_\mathbf{A})$.*

Here the $\widetilde{O}_p(\cdot)$ hides $\mathrm{poly}(\log n, \log(1/\varepsilon))$ factors and constants depending on p (at worst exponential). As a corollary we also obtain high precision solvers for the Lagrange dual problem $\min_{\mathbf{A}^\top x = b} \|x\|_q$ for $q = p/(p-1)$ in $\widetilde{O}_p(d^{\frac{p-2}{3p-2}}\mathcal{T}_\mathbf{A})$ time whp. This improves over the $\widetilde{O}_p(n^{\frac{p-2}{3p-2}})$ iteration complexity of [3] for any tall matrix $\mathbf{A}$.

[4] Throughout we assume that all solves to $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ are exact. Typically it suffices to set the solver error to be polynomially small in n, d and the largest entries of the input vectors and matrix $\mathbf{A}$. This increases the running time of the solver by polylogarithmic factors.

Similar ideas as those used to show Theorem 1 can be used to give improved iteration complexities for approximate ℓ_∞ regression, which we show in Appendix A.

Theorem 2 (Approximate ℓ_∞ regression). *There is an algorithm that given any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, whp. computes an ε-accurate solution to ℓ_∞ regression in time $\widetilde{O}(d^{1/3}\varepsilon^{-2/3} \cdot \mathcal{T}_\mathbf{A})$.*

This improves over the $O(n^{1/3}\varepsilon^{-O(1)})$ iteration bounds achieved by [18, 20, 33].

We also obtain improved results for ℓ_q regression for $q \leq 2$ for sufficiently tall matrices $\mathbf{A}$.

Theorem 3 (High precision ℓ_q regression for $q \leq 2$). *There is an algorithm that given any $\mathbf{A} \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$, and $q \in (1, 2]$ whp. returns an ε-accurate solution to ℓ_q regression in time $\widetilde{O}_p(d^{\frac{p-2}{2p-2}} \cdot \mathcal{T}_\mathbf{A})$ for $p = q/(q-1)$.*

As a corollary we also get high accuracy solvers for the Lagrange dual problem $\min_{\mathbf{A}^\top x = b} \|x\|_p$ for $p = q/(q-1)$ in $\widetilde{O}_p(d^{\frac{p-2}{2p-2}}\mathcal{T}_\mathbf{A})$ time whp. This improves over the $\widetilde{O}_p(n^{\frac{p-2}{3p-2}})$ iteration complexity of [3] for any sufficiently tall matrix $\mathbf{A}$ with $n = \omega(d^{\frac{3p-2}{2p-2}})$.

1.2 Previous Work

Here we briefly survey related work to the problems we consider and the optimization and numerical methods we build upon.

Regression: Beyond the works mentioned earlier, there are several results on ℓ_1 or ℓ_∞ regression in both the low precision ($\varepsilon^{-O(1)}$ dependence) [23, 32, 56, 63], and high precision regimes corresponding to linear programming [52]. Additionally the works [2, 18, 19] study the more general problem of quasi-self-concordant optimization, which captures ℓ_p regression as well as logistic regression [9, 45]. There are several results on ℓ_p regression that are based on sparsification or subspace embeddings using (variants of) ℓ_p Lewis weights [22, 24, 25, 30, 51, 62]. While our result also uses a variant of ℓ_p Lewis weights, we do not sparsify. This is key to achieving our iteration complexities because sparsification of an ℓ_p norm objective for $p \geq 2$ requires at least $\Omega(d^{p/2})$ rows [29]. This leads to iteration complexities of at least $(d^{p/2})^{(p-2)/(3p-2)} = d^{(p^2-2p)/(6p-4)}$ and runtimes of $\widetilde{O}(\mathrm{nnz}(\mathbf{A}) + d^{\Omega(p)})$, as was noted in [1, Theorem 2.6]. On the other hand, our iteration complexity is at most $\widetilde{O}(d^{1/3})$, independent of p. This allows us to achieve a $\widetilde{O}(d^{1/3}\varepsilon^{-2/3})$ iteration complexity for ℓ_∞-regression in Theorem 2, while the aforementioned works using sparsification are unable to. Very recently, [35] achieved a $O(n^\theta)$ runtime for ℓ_p norm regression on sufficiently sparse matrices $\mathbf{A}$ for some $\theta < \omega$ (the matrix multiplication exponent). For p near 2, they improved this to $\widetilde{O}(\mathrm{nnz}(\mathbf{A}) + d^\theta)$.

High-order acceleration: Our Monteiro-Svaiter acceleration algorithm builds upon works pertaining to the acceleration of functions with Lipschitz p-th order derivatives. For $p = 1$ this corresponds to classic acceleration of smooth functions that attains error $\widetilde{O}(1/k^2)$ over k iterations [55]. A series of works [6, 7, 15, 17, 18, 34, 53, 54] has shown that the optimal error bound is given by $\widetilde{O}(1/k^{(3p+1)/2})$ over k iterations for functions with Lipschitz p-th order derivatives.

Our Monteiro-Svaiter acceleration algorithm for ℓ_p regression directly utilizes a generalized accelerated proximal-point framework from [15]. Additionally, [16] has given an algorithm that achieves an accelerated convergence rate for the more general problem of minimizing structured convex quartics which captures ℓ_4 regression but has an additional third order tensor term. It is interesting to understand whether our methods extend to that setting.

Width reduction: In addition to its applications for regression problems as described, similar width reduction techniques have been applied to give improved runtimes for the maxflow problem in both approximate regimes [21, 39] and in unit capacity graphs [8, 28, 37, 46, 47, 49, 50]. Additionally, the Iteratively Reweighted Least Squares (IRLS) algorithm of Ene-Vladu [33] gives an alternate approach based on width reduction for achieving a $\widetilde{O}(n^{1/3}\varepsilon^{-2/3})$ iteration complexities for ℓ_1 and ℓ_∞ regression, matching the iteration complexity of [18]. We believe that applying ideas from the analysis of [33] can potentially be used to simplify our width reduction algorithm for ℓ_p regression given in Section 3.

Runtime improvements for regression problems. We briefly discuss why we focus on presenting iteration complexity improvements in this work, as opposed to runtimes for ℓ_p regression. In general, obtaining improving runtimes for regression problems beyond improving the iteration complexity has been through *inverse maintenance* techniques [41, 59, 60], and more recently heavy hitter and iterate maintenance [10–13, 27, 43], to speed up the amortized time to solve the linear systems in $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ and implicitly maintain the iterates. This direction has seen an explosion of work recently, with the state-of-the-art runtimes for solving linear programs (eg. high precision ℓ_1 regression) being some combination of the recent works $\widetilde{O}(n^{\max\{\omega,2+1/18\}})$ [36], $\widetilde{O}(nd+d^{2.5})$ [11], and $\widetilde{O}(\mathrm{nnz}(\mathbf{A})d^{0.5}+d^{2.5})$ [41]. The authors believe that all our improved iteration complexities in Theorems 1 to 3 can be combined with ideas from the aforementioned works to achieve concrete runtime improvements for ℓ_p regression. However, given the rapidly evolving progress in inverse maintenance and relative complexity of the methods, we choose to focus on iteration complexities in this work to give a cleaner and simpler presentation of our ideas.

1.3 Our Approach

Here we focus on presenting our approach for $p \geq 2$ (Theorem 1) and briefly describe our approach for $q \leq 2$ (Theorem 3). Both of our algorithmic frameworks (width reduction and acceleration) are based on leveraging properties of ℓ_p Lewis weights. While ℓ_p Lewis weights have been used in several previous results on ℓ_p regression (as described in Section 1.2), these works primarily used Lewis weights to construct sparsifiers or subspace embeddings. We take a different perspective, and instead leverage a key fact about approximate ℓ_p Lewis weights that they provide an ellipse which approximates the $\|\mathbf{A}x\|_p$. This has appeared in [61, pg.115] and [22, Lemma 3.6]. Precisely, if $w \in \mathbb{R}^n$ are the ℓ_p-Lewis weights for $\mathbf{A}$ then

$$\|\mathbf{A}x\|_p \leq \|\mathbf{W}^{\frac{1}{2}-\frac{1}{p}}\mathbf{A}x\|_2 \leq \|w\|_1^{\frac{1}{2}-\frac{1}{p}}\|\mathbf{A}x\|_p \text{ for all } x \in \mathbb{R}^n. \quad (2)$$

The lower bound follows from the definition of ℓ_p Lewis weights, which we give a self-contained proof of in Lemma 2.6, and the upper bound follows from Hölder's inequality. Because $\|w\|_1 = d$ for the ℓ_p Lewis weights, the distortion between the lower and upper bounds in (2) is $d^{1/2-1/p}$, which leads to d-dependent iteration complexities. While it is not known how to exactly compute the ℓ_p Lewis weights for $p \geq 4$ in $\widetilde{O}(\mathcal{T}_\mathbf{A})$ time we are still able to argue that we can efficiently compute weights w satisfying (2) but $\|w\|_1 \leq 2d$ (Lemma 2.5). This is done by mimicking an argument of [26] for the $p = \infty$ case which corresponds to computing an approximate John ellipse.

We show that it is possible to leverage our perspective on (2) within either iterative refinement [3] or an acceleration framework (based on the acceleration framework of [15]). While these frameworks are largely compatible with (2), there are notable conceptual differences which we now discuss. In the iterative refinement framework, the problem of ℓ_p regression is reduced to approximately minimize problems that are a combination of a linear term, ℓ_p norm term, and ℓ_2 regularization term (Problem 3.1). As in [3], we use a width-reduced multiplicative weights update (MWU) to reduce the iteration complexity. The main difference is that we show an energy boosting lemma in the width-reduced MWU (Lemma 3.6) that allows for resistances to more than double (while still providing significant increase to the energy potential) by leveraging stability from (2), while in standard energy boosting the energy does not increase significantly beyond resistances increasing by a constant factor. While the proof follows gracefully from low-rank update formulas, we believe that this is an interesting conceptual point. Our second acceleration-based algorithm repeatedly solves proximal subproblems of the form $\min_x \|\mathbf{A}x-b\|_p^p + O(p)^p\|x-y\|^p_{\mathbf{A}^\top\mathbf{W}^{1-2/p}\mathbf{A}}$: we show that such regularized problems may be solved efficiently by using stability given by (2). Interestingly, a more naïve application of acceleration of ball-constrained Newton methods [18] leads to an iteration complexity of $\widetilde{O}_p(d^{1/3})$. However, our acceleration method achieves an iteration complexity of $\widetilde{O}_p(d^{(p-2)/(3p-2)})$ and provides an acceleration-based alternative matching the iteration complexities achieved by width-reduction for intermediate values of $p \in [2,\infty)$.

For the case $q \leq 2$ instead of solving $\min_{x\in\mathbb{R}^d}\|\mathbf{A}x-b\|_q$ we solve the dual problem

$$\min_{\mathbf{A}^\top x=0, b^\top x=-1}\|x\|_p$$

where $p = q/(q-1)$ is the dual norm. In this setting we also wish to use ℓ_q Lewis weights. However the presence of the ℓ_2 regularizer induced from iterative refinement or the acceleration framework forces us to use a more complex *regularized Lewis weight*, defined in Definition 5.2 (such a concept was also used in [11, Definition 4.4]). Unfortunately it seems that this type of regularized Lewis weight is not immediately compatible with the width reduction or acceleration type speedups, and we only achieve a $\widetilde{O}_p(d^{(p-2)/(2p-2)})$ iteration complexity as a result. Consequently, we believe that achieving a matching $\widetilde{O}_p(d^{(p-2)/(3p-2)})$ complexity for the case $q \leq 2$ is an important open problem.

1.4 Paper Organization

The remainder of the paper is structured as follows. In Section 2 we give preliminaries for our algorithms, e.g. leverage scores, Lewis

weights, and iterative refinement. In Section 3 we provide an iterative refinement and width reduction framework for showing Theorem 1. In Section 4 we give an alternate approach for the previous result based on the high-order acceleration framework of [15]. In Section 5 we show Theorem 3 which achieves d-dependent (as opposed to n-dependent) iteration complexities for ℓ_q regression for $q \leq 2$. Finally we show our result on approximate ℓ_∞ regression (Theorem 2) in Appendix A.

2 PRELIMINARIES

2.1 General Notation

We use lowercase for vectors, and capital boldface for matrices. We let $\vec{0}, \vec{1}$ denote the all 0, 1 vectors respectively. Additionally, for a vector the matrix with corresponding capital letter is the diagonal matrix. Throughout we let w denote a weight vector, r denote a positive vector, and $\mathbf{W} = \mathbf{diag}(w)$ and $\mathbf{R} = \mathbf{diag}(r)$. We say that a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ is positive semidefinite (PSD) if $x^\top \mathbf{B} x \geq 0$ for all $x \in \mathbb{R}^n$. We say that matrices $\mathbf{A} \preceq \mathbf{B}$ if $\mathbf{B} - \mathbf{A}$ is PSD. For PSD matrices $\mathbf{A}, \mathbf{B}$ we say that $\mathbf{A} \approx_\alpha \mathbf{B}$ for $\alpha \geq 1$ if $\alpha^{-1}\mathbf{B} \preceq \mathbf{A} \preceq \alpha \mathbf{B}$. For a PSD matrix $\mathbf{B}$ we define the seminorm induced by $\mathbf{B}$ as $\|x\|_\mathbf{B} := \sqrt{x^\top \mathbf{B} x}$.

2.2 Lewis Weights

We start by defining the leverage scores and ℓ_p Lewis weights of a matrix $\mathbf{A}$. These are measures of importance of rows of a matrix $\mathbf{A}$.

Definition 2.1 (Leverage scores). *For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ whose i-th row is the vector a_i, the* leverage scores *are given by $\sigma(\mathbf{A})_i := a_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} a_i$ for $i \in [n]$.*

It is known that $\sum_{i \in [n]} \sigma(\mathbf{A})_i = \text{rank}(\mathbf{A})$. Further, the leverage score of the i-th row of a matrix $\mathbf{A}$ is given by the maximum of $|(\mathbf{A}x)_i|$ over all vectors $x \in \mathbb{R}^d$ satisfying $\|\mathbf{A}x\|_2 \leq 1$. This provides a concrete way that the leverage scores are ℓ_2 importance measures of rows.

Fact 2.2 (Leverage scores as ℓ_2 importance). *For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ the leverage score of row $i \in [n]$ is given by*

$$\sigma(\mathbf{A})_i = \max_{x \in \mathbb{R}^d : \mathbf{A}x \neq 0} \frac{(\mathbf{A}x)_i^2}{\|\mathbf{A}x\|_2^2}.$$

Lewis weights are a generalization of leverage scores to ℓ_p norms for $p \neq 2$.

Definition 2.3 (ℓ_p Lewis weights). *For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, the ℓ_p* Lewis weights *are given by the unique vector $w \in \mathbb{R}^n_{\geq 0}$ satisfying $w_i = \sigma(\mathbf{W}^{1/2 - 1/p}\mathbf{A})_i$ for all $i \in [n]$.*

[29] proves the existence and uniqueness of ℓ_p Lewis weights for all $p \in (0, \infty)$. Additionally, they provide an efficient contractive procedure for approximately computing the ℓ_p Lewis weight for $p < 4$. For our applications for $p < 2$, we use a regularized version of these weights, and defer the full statement of the approximation result needed until Lemma 5.3 in Section 5. For our applications for $p \geq 4$ we show that it is possible to compute weights satisfying the weaker guarantee (2).

Definition 2.4 (ℓ_p Lewis weight overestimates). *For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ we say that $w \in \mathbb{R}^n_{\geq 0}$ are* ℓ_p Lewis weight overestimates *if $d \leq \|w\|_1 \leq 2d$ and $w_i \geq \sigma(\mathbf{W}^{1/2 - 1/p}\mathbf{A})_i$ for all $i \in [n]$.*

The factor of 2 is somewhat arbitrary – any constant factor suffices for our algorithms. We require the following lemma showing that Lewis weight overestimates can be computed with a few linear system solves. Its proof is deferred to the full version. Our approach is an extension of that in [26] which provided a procedure for approximately computing the John ellipse, i.e. the $p = \infty$ case.

Lemma 2.5 (Computing Lewis weight overestimates). *There is an algorithm ApproxLargeWeights$(\mathbf{A}, p)$ that given any $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $p \geq 2$ in $\widetilde{O}(\mathcal{T}_\mathbf{A})$ time computes ℓ_p Lewis weight overestimates (Definition 2.4) whp.*

We can show (2) holds for any ℓ_p Lewis weight overestimates.

Lemma 2.6. *For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and ℓ_p Lewis weight overestimates $w \in \mathbb{R}^n_{>0}$ (Definition 2.4) we have that $\|\mathbf{A}x\|_p \leq \|\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A}x\|_2$ for all $x \in \mathbb{R}^n$.*

Proof. By Fact 2.2 we know that

$$\begin{aligned} |(\mathbf{A}x)_i| &= w_i^{-\frac{1}{2} + \frac{1}{p}} |(\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A}x)_i| \\ &\leq w_i^{-\frac{1}{2} + \frac{1}{p}} \sigma(\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A})_i^{1/2} \|\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A}x\|_2 \\ &\leq w_i^{1/p} \|\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A}x\|_2. \end{aligned}$$

Hence, we see that

$$\begin{aligned} \|\mathbf{A}x\|_p^p &= \sum_{i \in [n]} |(\mathbf{A}x)_i|^p \\ &\leq \sum_{i \in [n]} w_i^{\frac{p-2}{p}} \|\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A}x\|_2^{p-2} (\mathbf{A}x)_i^2 = \|\mathbf{W}^{\frac{1}{2} - \frac{1}{p}}\mathbf{A}x\|_2^p. \end{aligned}$$

Taking the p-th root of both sides gives us the result. □

2.3 Iterative refinement

At a high level, the iterative refinement framework for ℓ_p norms introduced by [3] shows that the Bregman divergence of the ℓ_p norm, i.e. the function $f(x) = |x|^p$, can be efficiently approximated by an ℓ_2 and ℓ_p component. Using this, we can reduce solving high accuracy ℓ_p-norm problems to solving approximate ℓ_2-ℓ_p norm problems.

Lemma 2.7 [4, Lemma B.1]). *For $x, \Delta \in \mathbb{R}^n$ and $p \geq 2$, we have for $g, r \in \mathbb{R}^n$ defined by $g_i = p|x_i|^{p-2}x_i$ and $r_i = |x_i|^{p-2}$ for $i \in [n]$ that*

$$\frac{p}{8} \sum_{i \in [n]} r_i \Delta_i^2 + 2^{-p-1} \|\Delta\|_p^p \leq \|x + \Delta\|_p^p - \|x\|_p^p - g^\top \Delta \tag{3}$$

$$\leq 2p^2 \sum_{i \in [n]} r_i \Delta_i^2 + p^p \|\Delta\|_p^p. \tag{4}$$

There are several more restrictive variations of Lemma 2.7 for positive scalars that we use whose proofs are deferred to the full version.

Lemma 2.8. *For all $a, b \geq 0$ and $k \geq 2$ we have that $(a+b)^k - a^k \leq 3ka^{k-1}b + 3k^k b^k$.*

The second corollary is useful in slightly different regimes of the exponent k.

Lemma 2.9. *For all $a, b \geq 0$ and $k \geq 1$ we have that $(a+b)^k - a^k \leq 4^k(a^{k-1}b + b^k)$.*

Searching over the value of $g^\top \Delta$ reduces solving ℓ_p regression to high accuracy to approximately solving constrained ℓ_2-ℓ_p problems. We call a procedure for approximately solving constrained ℓ_2-ℓ_p problems a *γ-solver* and provide this reduction, [4, Theorem 3.1] below.

Definition 2.10 (γ-solver). *We call an algorithm a γ-solver if given $v \geq 0$, $g \in \mathbb{R}^n$, and a positive diagonal matrix $\mathbf{R} \in \mathbb{R}^{n \times n}$, it is the case that for*

$$\mathrm{OPT} = \min_{\mathbf{C}\Delta = 0, g^\top \Delta = -v} \Delta^\top \mathbf{A}^\top \mathbf{R} \mathbf{A} \Delta + \|\mathbf{A}\Delta\|_p^p,$$

the algorithm returns a $\widehat{\Delta}$ satisfying $\mathbf{C}\widehat{\Delta} = 0$, $g^\top \widehat{\Delta} = -v$, and

$$\widehat{\Delta}^\top \mathbf{A}^\top \mathbf{R} \mathbf{A} \widehat{\Delta} \leq \gamma \mathrm{OPT} \quad \text{and} \quad \|\mathbf{A}\widehat{\Delta}\|_p^p \leq \gamma^{p-1} \mathrm{OPT}.$$

Lemma 2.11 [4, Theorem 3.1]). *Given $\mathbf{U} \in \mathbb{R}^{n_1 \times d}, \mathbf{A} \in \mathbb{R}^{n_2 \times d}$ and b, v and $p \geq 2$, we can compute an $x \in \mathbb{R}^d$ satisfying $\mathbf{U}x = v$ and*

$$\|\mathbf{A}x - b\|_p \leq (1+\varepsilon) \min_{\mathbf{U}x = v} \|\mathbf{A}x - b\|_p$$

in $O(p^{3.5}\gamma \log(m/\varepsilon))$ calls to a γ-solver (Definition 2.10).

3 ENERGY BOOSTING ALGORITHM FOR LARGE p

The goal of this section is to give an algorithm to show Theorem 1. By Lemma 2.11 and scaling we may assume that $v = 1$ and $\mathrm{OPT} = 1$ [3, Lemma 5.4] throughout, and we use the following setup throughout the section.

Problem 3.1 (Scaled residual). *In the* scaled residual problem *we are given $\mathbf{A} \in \mathbb{R}^{n \times d}$, $g \in \mathbb{R}^d$, and diagonal $\mathbf{R} \in \mathbb{R}^{n \times n}_{\geq 0}$ such that there exists $x_\star \in \mathbb{R}^d$ satisfying $g^\top x_\star = -1$ with $x_\star^\top \mathbf{A}^\top \mathbf{R} \mathbf{A} x_\star \leq 1$ and $\|\mathbf{A}x_\star\|_p \leq 1$. We call y an α-*approximate solution *to the problem if $g^\top y = -1$, $y^\top \mathbf{A}^\top \mathbf{R} \mathbf{A} x_\star \leq \alpha$ and $\|\mathbf{A}y\|_p \leq \alpha$.*

In the notation of Problem 3.1, proving the following lemma suffices to show Theorem 1.

Lemma 3.2. *Algorithm Oracle$(\mathbf{A}, g, \mathbf{R}, p)$ takes an instance of Problem 3.1 and returns an $O(p)^p$-approximate y in $O(p)^p d^{\frac{p-2}{3p-2}} \cdot \mathcal{T}_\mathbf{A}$ time whp.*

Proof of Theorem 1. Clearly Lemma 3.2 satisfies the conditions of Lemma 2.11 for $\gamma = O(p)^p$. Each call to Lemma 3.2 requires $O(p)^p d^{\frac{p-2}{3p-2}}$ calls to a solver to $\mathbf{A}^\top \mathbf{D} \mathbf{A}$ so the total number of iterations is

$$\gamma \cdot p^{3.5} \cdot O(p)^p d^{\frac{p-2}{3p-2}} \log(m/\varepsilon) = O(p)^p d^{\frac{p-2}{3p-2}} \log(m/\varepsilon).$$

□

To show Lemma 3.2 we use the following Line 11. It follows the multiplicative weights and width reduction approach of [3, 21]. The algorithm solves ℓ_2-norm problems in sequence. When the ℓ_p norm of the resulting solution is small enough, i.e. $\|\mathbf{A}z\|_p^p \leq \tau$,

Algorithm 1: Oracle$(\mathbf{A}, g, \mathbf{R}, p)$. Given $\mathbf{A}, \mathbf{R}, g$ satisfying Problem 3.1, returns a $y \in \mathbb{R}^d$ with $g^\top y = -1$, $y^\top \mathbf{A}^\top \mathbf{R} \mathbf{A} y \leq O_p(1)$, and $\|\mathbf{A}y\|_p = O_p(1)$ in $O_p(d^{\frac{p-2}{3p-2}})$ iterations.

1 $w \leftarrow$ ApproxLewis$(\mathbf{A}, p)$. ▷ Compute ℓ_p Lewis weight overesimates of $\mathbf{A}$ via Lemma 2.5
2 $y \leftarrow 0$ and $s \leftarrow w^{1/p}$. ▷ Iterates
3 $\kappa \leftarrow \kappa_p d^{1/p}$, $\alpha \leftarrow \alpha_p d^{-\frac{p^2-5p+2}{p(3p-2)}}$, $\tau \leftarrow \tau_p d^{\frac{(p-2)(p-1)}{3p-2}}$. ▷ Constants κ_p, τ_p large, α_p small[5]
4 **for** $t = 0, 1, \ldots, T \stackrel{\text{def}}{=} \lfloor \alpha^{-1} d^{1/p} \rfloor$ **do**
5 $z \leftarrow \arg\min_{g^\top x = -1} x^\top \mathbf{A}^\top \left(d^{1-\frac{2}{p}} \mathbf{R} + \mathbf{S}^{p-2} \right) \mathbf{A}x$. ▷ $\mathbf{S} = \mathbf{diag}(s)$
6 **while** $\|\mathbf{A}z\|_p^p \geq \tau$ **do**
7 $S \leftarrow \{i \in [n] : s_i \leq 2^{-\frac{p}{p-2}} \kappa |(\mathbf{A}z)_i|\}$. ▷ Boosting step.
8 $s_i \leftarrow \left(s_i^{p-2} + \frac{\tau^{2/p} |(\mathbf{A}z)_i|^{p-2}}{4 \|\mathbf{A}z\|_p^p} \right)^{\frac{1}{p-2}}$ for $i \in S$.
9 $z \leftarrow \arg\min_{g^\top x = -1} x^\top \mathbf{A}^\top \left(d^{1-\frac{2}{p}} \mathbf{R} + \mathbf{S}^{p-2} \right) \mathbf{A}x$.
10 $y \leftarrow y + \alpha z$ and $s \leftarrow s + \alpha |\mathbf{A}z|$. ▷ Progress step.
11 **return** $(\alpha T)^{-1} y$.

the algorithm performs a *progress step* in line 10, and adds z to the output. However, whenever the ℓ_p norm of the returned solution is large, the algorithm performs a *boosting step* in line 8, and increases the resistance of the large coordinates contributing significantly to the ℓ_p norm $\|\mathbf{A}z\|_p^p$ to force them to be smaller in future iterations.

To analyze Algorithm Oracle$(\mathbf{A}, g, \mathbf{R}, p)$ in Line 11 and thereby prove Lemma 3.2, we analyze two potential functions following the approach and notation of [3]. The first is $\Phi(s) \stackrel{\text{def}}{=} \|s\|_p^p$, and the second is the *energy* (where $\mathbf{S} \stackrel{\text{def}}{=} \mathbf{diag}(s)$)

$$\mathcal{E}(s) \stackrel{\text{def}}{=} \min_{g^\top x = -1} x^\top \mathbf{A}^\top \left(d^{1-\frac{2}{p}} \mathbf{R} + \mathbf{S}^{p-2} \right) \mathbf{A}x.$$

We show that a progress or boosting step doesn't increase Φ by too much, and that a boosting step significantly increases the energy. Combining this with an energy upper bound completes the proof. To reason about the energy increase we use the following alternate characterization of the energy.

Lemma 3.3. *For any symmetric positive definite matrix $\mathbf{B}$ and vector g we have*

$$\arg\min_{g^\top x = -1} x^\top \mathbf{B} x = -\frac{1}{g^\top \mathbf{B}^{-1} g} \mathbf{B}^{-1} g \text{ and } \min_{g^\top x = -1} x^\top \mathbf{B} x = (g^\top \mathbf{B}^{-1} g)^{-1}. \tag{5}$$

Proof. Let x^* be the minimizer of (5). Note that $g^\top x^* = -1$ by assumption and $\mathbf{B}x^* = \alpha^* g$ for some unknown α^*. Consequently, $x^* = \alpha^* \mathbf{B}^{-1} g$ and the claim follows from

$$-1 = g^\top x^* = \alpha^* g^\top \mathbf{B}^{-1} g.$$

The second claim follows by using this value to compute $x^{*\top} \mathbf{B} x^*$. □

Lemma 3.4 (Energy upper bound). *In Problem 3.1, for any vector s satisfying $s \geq w^{1/p}$ coordinate-wise for ℓ_p Lewis weight overestimates w (Definition 2.4), we have $\mathcal{E}(s) \leq 2\Phi(s)^{1-\frac{2}{p}}$.*

Proof. Let $x_\star$ be as in Problem 3.1. By Hölder's inequality we have that

$$\begin{aligned}\mathcal{E}(s) &\leq x_\star^\top \mathbf{A}^\top \left(d^{1-\frac{2}{p}}\mathbf{R} + \mathbf{S}^{p-2}\right)\mathbf{A}x_\star \leq d^{1-\frac{2}{p}} + \|\mathbf{A}x_\star\|_p^2 \|s\|_p^{p-2} \\ &\leq d^{1-\frac{2}{p}} + \|s\|_p^{p-2} = d^{1-\frac{2}{p}} + \Phi(s)^{1-\frac{2}{p}} \\ &\leq 2\Phi(s)^{1-\frac{2}{p}},\end{aligned}$$

where the final inequality follows from $\Phi(s) \geq \|w\|_1 \geq d$. □

Lemma 3.5 (Progress step). *Let $s^{\text{new}} = s + \alpha|\mathbf{A}z|$, as defined in line 10 of Line 11. Then we have that $\mathcal{E}(s^{\text{new}}) \geq \mathcal{E}(s)$ and*

$$\Phi(s^{\text{new}}) - \Phi(s) \leq 5p\alpha\Phi(s)^{1-\frac{1}{p}} + 3p^p\alpha^p\tau. \tag{6}$$

Proof. To bound $\mathcal{E}(s^{\text{new}})$, note that $s^{\text{new}} \geq s \geq \vec{0}$ entrywise. Therefore, $\mathcal{E}(s^{\text{new}}) \geq \mathcal{E}(s)$.

To bound $\Phi(s^{\text{new}})$ we compute

$$\begin{aligned}\Phi(s^{\text{new}}) - \Phi(s) &= \|s + \alpha|\mathbf{A}z|\|_p^p - \|s\|_p^p \\ &\overset{(i)}{\leq} 3p\alpha \sum_{i\in[n]} s_i^{p-1}|(\mathbf{A}z)_i| + 3p^p\alpha^p\|\mathbf{A}z\|_p^p \\ &\overset{(ii)}{\leq} 3p\alpha\left(\sum_{i\in[n]} s_i^p\right)^{1/2}\left(\sum_{i\in[n]} s_i^{p-2}(\mathbf{A}z)_i^2\right)^{1/2} + 3p^p\alpha^p\tau \\ &\overset{(iii)}{\leq} 3p\alpha\sqrt{\Phi(s)\mathcal{E}(s)} + 3p^p\alpha^p\tau \overset{(iv)}{\leq} 5p\alpha\Phi(s)^{1-\frac{1}{p}} + 3p^p\alpha^p\tau.\end{aligned}$$

Here, (i) follows from Lemma 2.8 for $k = p$, (ii) follows from the Cauchy-Schwarz inequality, (iii) follows from the fact that z is the minimizer for $\mathcal{E}(s)$, and (iv) follows from Lemma 3.4 that $\Phi(s) \leq 2\mathcal{E}(s)^{1-\frac{2}{p}}$. □

To analyze the boosting step we provide a general lemma about energy increase under boosting edges. Interestingly, this allows for resistances to increase by more than a constant factor, thereby going beyond the standard energy boosting lemmas in [21, 49, 50].

Lemma 3.6 (Energy increase). *Let $w \in \mathbb{R}^n_{\geq 0}$ be ℓ_p Lewis weight overestimates for $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{D} \geq \mathbf{W}^{1-\frac{2}{p}}$ be a diagonal matrix, and $v \in \mathbb{R}^n_{\geq 0}$ satisfy $\|v\|_{\frac{p}{p-2}} \leq 1$. For $\mathcal{E} \overset{\text{def}}{=} \min_{g^\top x=-1} x^\top\mathbf{A}^\top\mathbf{D}\mathbf{A}x$, $\mathcal{E}^{\text{new}} \overset{\text{def}}{=} \min_{g^\top x=-1} x^\top\mathbf{A}^\top(\mathbf{D}+\mathbf{V})\mathbf{A}x$, and $y \overset{\text{def}}{=} \arg\min_{g^\top x=-1} x^\top\mathbf{A}^\top\mathbf{D}\mathbf{A}x$ the following holds*

$$\mathcal{E}^{\text{new}} - \mathcal{E} \geq \frac{1}{2}\sum_{i\in[n]} v_i(\mathbf{A}y)_i^2.$$

Proof. By Lemma 3.3

$$\mathcal{E} = (g^\top(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}g)^{-1} \text{ and } y = -\frac{1}{g^\top(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}g}(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}g.$$

By the Woodbury matrix identity we have that

$$\begin{aligned}&\mathcal{E}^{-1} - (\mathcal{E}^{\text{new}})^{-1} &(7)\\ &= g^\top(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{V}^{\frac{1}{2}}(\mathbf{I}+\mathbf{P})^{-1}\mathbf{V}^{\frac{1}{2}}\mathbf{A}(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}g \\ &= \frac{1}{\mathcal{E}^2}y^\top\mathbf{A}^\top\mathbf{V}^{\frac{1}{2}}(\mathbf{I}+\mathbf{P})^{-1}\mathbf{V}^{\frac{1}{2}}\mathbf{A}y &(8)\end{aligned}$$

where $\mathbf{P} \overset{\text{def}}{=} \mathbf{V}^{\frac{1}{2}}\mathbf{A}(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{V}^{\frac{1}{2}}$. We next claim that $\mathbf{A}^\top\mathbf{V}\mathbf{A} \preceq \mathbf{A}^\top\mathbf{D}\mathbf{A}$. To show this, note that for any $x \in \mathbb{R}^n$ we have that

$$\begin{aligned}x^\top\mathbf{A}^\top\mathbf{V}\mathbf{A}x = \sum_{i\in[n]} v_i(\mathbf{A}x)_i^2 &\overset{(i)}{\leq} \|v\|_{\frac{p}{p-2}}\|\mathbf{A}x\|_p^2 \\ &\overset{(ii)}{\leq} x^\top\mathbf{A}^\top\mathbf{W}^{1-\frac{2}{p}}\mathbf{A}x \overset{(iii)}{\leq} x^\top\mathbf{A}^\top\mathbf{D}\mathbf{A},\end{aligned}$$

where (i) follows from Hölder's inequality's inequality, and (ii) from the condition $\|v\|_{\frac{p}{p-2}} \leq 1$ and Lemma 2.6, and (iii) from $\mathbf{W}^{1-\frac{2}{p}} \preceq \mathbf{D}$. Note that this additionally implies that

$$\mathbf{P} = \mathbf{V}^{\frac{1}{2}}\mathbf{A}(\mathbf{A}^\top\mathbf{D}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{V}^{\frac{1}{2}} \preceq \mathbf{V}^{\frac{1}{2}}\mathbf{A}(\mathbf{A}^\top\mathbf{V}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{V}^{\frac{1}{2}} \preceq \mathbf{I}$$

where the last step follows because the matrix is an orthogonal projection matrix.

Applying these bounds to (8) yields that

$$\mathcal{E}^{-1} - (\mathcal{E}^{\text{new}})^{-1} \geq \frac{1}{2\mathcal{E}^2}\sum_{i\in[n]} v_i(\mathbf{A}y)_i^2.$$

Using that $\mathcal{E}^{\text{new}} \geq \mathcal{E}$ and rearranging yields that

$$\mathcal{E}^{\text{new}} - \mathcal{E} \geq \frac{\mathcal{E}^{\text{new}}\mathcal{E}}{2\mathcal{E}^2}\sum_{i\in[n]} v_i(\mathbf{A}y)_i^2 \geq \frac{1}{2}\sum_{i\in[n]} v_i(\mathbf{A}y)_i^2.$$

□

Lemma 3.7 (Boosting step). *Let s be at the start of a boosting step, and s^{new} be defined as after the operations of line 8 in Oracle (Line 11). If $2^p\kappa^{-(p-2)}\Phi(s)^{1-\frac{2}{p}} \leq \tau/4$ then $\Phi(s^{\text{new}}) - \Phi(s) \leq 20\kappa^2(\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s))$ and $\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s) \geq \tau^{2/p}/16$.*

Proof. For z as in line 5 of Algorithm 1

$$\begin{aligned}\sum_{i\in S}|(\mathbf{A}z)_i|^p &= \|\mathbf{A}z\|_p^p - \sum_{i\notin S}|(\mathbf{A}z)_i|^p &(9)\\ &\overset{(i)}{\geq} \|\mathbf{A}z\|_p^p - \left(2^{-\frac{p}{p-2}}\kappa\right)^{-(p-2)}\sum_{i\notin S} s_i^{p-2}(\mathbf{A}z)_i^2 &(10)\\ &\overset{(ii)}{\geq} \|\mathbf{A}z\|_p^p - 2^{p+1}\kappa^{-(p-2)}\Phi(s)^{1-\frac{2}{p}} \overset{(iii)}{\geq} \|\mathbf{A}z\|_p^p/2, &(11)\end{aligned}$$

where (i) follows by the definition of S in line 7 in Oracle (Line 11), (ii) follows from Lemma 3.4, and (iii) follows by the condition on κ in the hypothesis and $\tau \leq \|\mathbf{A}z\|_p^p$ by the condition of line 6 in Oracle (Line 11).

Now we can lower bound $\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s)$ using Lemma 3.6. Set $\mathbf{D} = d^{1-\frac{2}{p}}\mathbf{R} + \mathbf{S}^{p-2}$ and $v_i = 0$ for $i \notin S$ and $v_i = \frac{\tau^{2/p}|(\mathbf{A}z)_i|^{p-2}}{4\|\mathbf{A}z\|_p^p}$ for $i \in S$. Note that $\|v\|_{\frac{p}{p-2}} \leq \tau^{2/p}/\|\mathbf{A}z\|_p^2 \leq 1$ by the condition

$\tau \le \|\mathbf{A}z\|_p^p$ of line 6 in Oracle (Line 11). Thus Lemma 3.6 gives

$$\begin{aligned}\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s) &\ge \frac{1}{2}\sum_{i\in[n]} v_i(\mathbf{A}z)_i^2 \\ &= \frac{1}{2}\cdot\frac{\tau^{2/p}}{4\|\mathbf{A}z\|_p^p}\sum_{i\in S}|(\mathbf{A}z)_i|^p \ge \tau^{2/p}/16\end{aligned}$$

where we have used (11) above.

To bound $\Phi(s^{\text{new}}) - \Phi(s)$, we use Lemma 2.9 for $k = p/(p-2)$ and $a = s_i^{p-2}$ and $b = v_i$ to get

$$\begin{aligned}\Phi(s^{\text{new}}) - \Phi(s) &= \sum_{i\in[n]}\left((s_i^{p-2}+v_i)^{\frac{p}{p-2}} - s_i^p\right) \\ &\le 4^{\frac{p}{p-2}}\sum_{i\in S}\left(s_i^2 v_i + v_i^{\frac{p}{p-2}}\right) \\ &\overset{(i)}{\le} 4^{\frac{p}{p-2}}\sum_{i\in S}\left(4^{-\frac{p}{p-2}}\kappa^2 v_i(\mathbf{A}z)_i^2 + v_i^{\frac{p}{p-2}}\right) \\ &\overset{(ii)}{\le} 2\kappa^2(\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s)) + 1.\end{aligned}$$

Here, (i) uses $s_i \le 2^{-\frac{p}{p-2}}\kappa|(\mathbf{A}z)_i|$ for all $i \in S$ by line 7 in Oracle (Line 11) and (ii) uses Lemma 3.6 and $\|v\|_{\frac{p}{p-2}} \le 1/4$. To conclude, note that $\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s) \ge \tau^{2/p}/16 \ge 1/16$, as $\tau \ge 1$. Also, $\kappa \ge 1$, so $2\kappa^2(\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s)) + 1 \le 20\kappa^2(\mathcal{E}(s^{\text{new}}) - \mathcal{E}(s))$. This completes the proof. □

Now we can combine the bounds on $\Phi(s)$ and $\mathcal{E}(s)$ in Lemmas 3.5 and 3.7 to prove Lemma 3.2.

Proof of Lemma 3.2. We set $\tau_p = 40^p$. Choose $\alpha_p = 1/(1000p)$ so that $p^p\alpha^p\tau \le p\alpha d^{1-\frac{1}{p}}$. Then $p^p\alpha^p\tau \le p\alpha\Phi(s)^{1-\frac{1}{p}}$ as $\Phi(s) \ge d$ for all $s \ge w^{1/p}$ for a Lewis weight overestimate w. Let $\kappa_p = p$.

Let s^{final} be the final value of s in a call to Oracle (Line 11). We show by induction that $\Phi(s^{\text{final}}) \le (20\kappa)^p$ and $2^p\kappa^{-(p-2)}\Phi(s)^{1-\frac{2}{p}} \le \tau/4$ during a successful execution of Algorithm 1 always (so the condition of Lemma 3.7 is satisfied). We start by bounding $\Phi(s^{\text{final}})$. As $\sum_{i\in[n]} w_i \le 2d$ by the definition of ℓ_p Lewis weight overestimates (Definition 2.4), initially $\Phi(s) \le 2d$. We calculate that

$$\begin{aligned}\Phi(s^{\text{final}}) &\overset{(i)}{\le} 2d + \alpha^{-1}d^{1/p}\left(5p\alpha\Phi(s^{\text{final}})^{1-\frac{1}{p}} + 3p^p\alpha^p\tau\right) + 20\kappa^2\mathcal{E}(s^{\text{final}}) \\ &\overset{(ii)}{\le} 2d + \alpha^{-1}d^{1/p}\cdot 8p\alpha\Phi(s^{\text{final}})^{1-\frac{1}{p}} + 40\kappa^2\Phi(s^{\text{final}})^{1-\frac{2}{p}} \\ &= 2d + 8p\Phi(s^{\text{final}})^{1-\frac{1}{p}}d^{1/p} + 40\kappa^2\Phi(s^{\text{final}})^{1-\frac{2}{p}}.\end{aligned}$$

where (i) follows from Lemmas 3.5 and 3.7, and (ii) follows from Lemma 3.4 and the bound $p^p\alpha^p\tau \le p\alpha\Phi(s)^{1-\frac{1}{p}}$ from our choice of τ_p and α_p. If $\Phi(s^{\text{final}}) > (20\kappa)^p$ then we get that

$$\begin{aligned}&2d\Phi(s^{\text{final}})^{-1} + 8pd^{1/p}\Phi(s^{\text{final}})^{-\frac{1}{p}} + 40\kappa^2\Phi(s^{\text{final}})^{-\frac{2}{p}} \\ &< \frac{1}{20} + \frac{8pd^{1/p}}{20\kappa} + \frac{40\kappa^2}{400\kappa^2} < 1,\end{aligned}$$

contradicting the above equation. Hence $\Phi(s^{\text{final}}) \le (20\kappa)^p$.

Now we check that $2^p\kappa^{-(p-2)}\Phi(s^{\text{final}})^{1-\frac{2}{p}} \le \tau/4$ to complete the induction. From the choice $\tau_p = 40^p$ and $\tau \ge \tau_p$, note that

$$2^p\kappa^{-(p-2)}\Phi(s^{\text{final}})^{1-\frac{2}{p}} \le 2^p\kappa^{-(p-2)}(20\kappa)^{p-2} \le 40^p/4 \le \tau/4.$$

We now show that the returned vector $x = (\alpha T)^{-1}y$ (for $T \overset{\text{def}}{=} \lfloor\alpha^{-1}d^{1/p}\rfloor$) satisfies $\|\mathbf{A}x\|_p \le O(p)$ and $x^\top\mathbf{A}^\top\mathbf{R}\mathbf{A}x \le O(p)^p$. Note that $(\alpha T)^{-1} \le 2d^{-1/p}$. For the first of these note that

$$\|\mathbf{A}x\|_p \le 2d^{-1/p}\|\mathbf{A}s^{\text{final}}\|_p = 2d^{-1/p}\Phi(s^{\text{final}})^{1/p} \le 40\kappa d^{-1/p} \le 40p$$

by the choice of κ_p. For the latter, note that

$$z^\top\mathbf{A}^\top\mathbf{R}\mathbf{A}z \le d^{\frac{2}{p}-1}\mathcal{E}(s) \le 2d^{\frac{2}{p}-1}\Phi(s)^{1-\frac{2}{p}} = O(p)^p$$

at each step – now apply the triangle inequality on the norm $\|\mathbf{R}^{1/2}\mathbf{A}z\|_2$.

Finally we bound the number of progress and boosting steps. The number of progress steps is bounded by $\alpha^{-1}d^{1/p} = O\left(pd^{\frac{p-2}{3p-2}}\right)$ by the choice of α. To bound the number of boosting steps, note that $\mathcal{E}(s)$ increases by $\tau^{2/p}/16$ per boosting step by Lemma 3.7, and is increasing every progress step by Lemma 3.5. As $\mathcal{E}(s^{\text{final}}) \le 2\Phi(s)^{1-\frac{2}{p}} \le 2(20\kappa)^{p-2}$ at the end we get that the number of boosting steps is bounded by

$$\frac{2(20\kappa)^{p-2}}{\tau^{2/p}/16} \le O(p)^p\cdot d^{1-\frac{2}{p}}\cdot d^{\frac{-2(p-2)(p-1)}{p(3p-2)}} = O(p)^p d^{\frac{p-2}{3p-2}}.$$

To compute the ℓ_p Lewis weights overestimates in line 1 in Oracle (Line 11) there are an additional $\widetilde{O}(1)$ solves to $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ by Lemma 2.5. Together, this gives the total iteration bound. □

4 MONTEIRO-SVAITER ACCELERATION ALGORITHM FOR LARGE p

In Section 3, we gave an algorithm for ℓ_p regression for $p \ge 2$ based on the iterative refinement framework of [3]. Here we give an alternate scheme with an improved dependence on p based on *highly-smooth* optimization. More specifically, we leverage an optimization framework from [15], which reduces the task of minimizing a convex function f to approximately solving proximal subproblems of the form

$$\text{Prox}(y) = \min_x f(x) + C_p\|x - y\|_{\mathbf{M}}^p$$

for arbitrary positive semidefinite matrix $\mathbf{M}$. Our result is a refinement of the $O(p^{14/3}n^{1/3})$ iteration complexity achieved in [18]. Our main technical ingredient is an improved Hessian stability bound (Lemma 4.3) which works for all $p \ge 2$ and allows us to take steps bounded in the norm induced by a matrix $\mathbf{M} \le \mathbf{A}^\top\mathbf{A}$. We leverage this to give an efficient algorithm for proximal subproblems, and combine with the acceleration framework of [15] to obtain our result.

4.1 Hessian stability

In this section, we prove our Hessian stability bound Lemma 4.3. We begin with a straightforward scalar inequality which we use in our proof.

Lemma 4.1. *Let $\alpha, \beta \geq 1$ satisfy $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. For any $n \geq 0$ and any x, y,*

$$|x+y|^n \leq |\alpha x|^n + |\beta y|^n.$$

Additionally, $|x+y|^{p-2} \leq e\,|x|^{p-2} + p^{p-2}\,|y|^{p-2}$ for $p \geq 2$.

Proof. Observe

$$\begin{aligned}|x+y|^n &= \left|\frac{\alpha x}{\alpha} + \frac{\beta y}{\beta}\right|^n \leq |\max\{|\alpha x|, |\beta y|\}|^n \\ &= \max\{|\alpha x|^n, |\beta y|^n\} \leq |\alpha x|^n + |\beta y|^n.\end{aligned}$$

Applying this inequality with $\alpha = \frac{p-1}{p-2}$, $\beta = p-1$, $n = p-2$ yields

$$\begin{aligned}|x+y|^{p-2} &\leq \left(1 + \frac{1}{p-2}\right)^{p-2} |x|^{p-2} + |(p-1)y|^{p-2} \\ &\leq e\,|x|^{p-2} + p^{p-2}\,|y|^{p-2}\end{aligned}$$

where the last inequality follows from $\left(1 + \frac{1}{x}\right)^x < e$ for any $x \geq 0$ and $p - 1 \leq p$. □

With this scalar inequality, we define a matrix $\mathbf{M}$ we will repeatedly appeal to in this section.

Definition 4.2. *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a matrix, and let $w \in \mathbb{R}^n$ be a vector of overestimates of the ℓ_p-Lewis weights of $\mathbf{A}$ (Definition 2.4). We set $\mathbf{M} \stackrel{\text{def}}{=} \mathbf{A}^\top \mathbf{W}^{1-2/p} \mathbf{A}$.*

With this, we prove our main Hessian stability fact Lemma 4.3:

Lemma 4.3. *Let $p \geq 2$, and define $f(x) = \|\mathbf{A}x - b\|_p^p$. Let $\mathbf{M} = \mathbf{A}^\top \mathbf{W}^{1-2/p} \mathbf{A}$ (Definition 4.2). For any $y \in \mathbb{R}^d$, define $f_y(x) = f(x) + C_p \|x - y\|_{\mathbf{M}}^p$ and $h_y(x) = \|x - y\|_{\nabla^2 f(y)}^2 + C_p \|x - y\|_{\mathbf{M}}^p$. Then if $C_p = e \cdot p^p$, for any x*

$$\frac{1}{e}\nabla^2 h_y(x) \preceq \nabla^2 f_y(x) \preceq e\nabla^2 h_y(x).$$

Proof. We first note

$$\nabla^2 f(x) = p(p-1)\mathbf{A}^\top \mathbf{diag}\left(|\mathbf{A}x - b|\right)^{p-2} \mathbf{A}.$$

For any vector z, we use Lemma 4.1 to get

$$\begin{aligned}z^\top \nabla^2 f(x) z &= p(p-1) \sum_{i \in [n]} |\mathbf{A}x - b|_i^{p-2} (\mathbf{A}z)_i^2 \\ &= p(p-1) \sum_{i \in [n]} |\mathbf{A}y - b + \mathbf{A}(x-y)|_i^{p-2} (\mathbf{A}z)_i^2 \\ &\leq \sum_{i \in [n]} \left(ep(p-1)|\mathbf{A}y - b|_i^{p-2} + p^p |\mathbf{A}(x-y)|_i^{p-2}\right)(\mathbf{A}z)_i^2.\end{aligned}$$

Now, by Hölder's inequality and Lemma 2.6 we get

$$\begin{aligned}\sum_{i \in [n]} p^p\,|\mathbf{A}(x-y)|_i^{p-2} (\mathbf{A}z)_i^2 &\leq p^p \left\||\mathbf{A}(x-y)|^{p-2}\right\|_{\frac{p}{p-2}} \left\|(\mathbf{A}z)^2\right\|_{\frac{p}{2}} \\ &= p^p \|\mathbf{A}(x-y)\|_p^{p-2} \|\mathbf{A}z\|_p^2 \\ &\leq p^p \|x-y\|_{\mathbf{M}}^{p-2} \|z\|_{\mathbf{M}}^2.\end{aligned}$$

Combining the above two inequalities yields

$$\begin{aligned}z^\top \nabla^2 f(x) z &\leq ep(p-1) z^\top \mathbf{A}^\top \mathbf{diag}\left(|\mathbf{A}y - b|^{p-2}\right)\mathbf{A}z \qquad (12) \\ &\quad + p^p \|x-y\|_{\mathbf{M}}^{p-2} \|z\|_{\mathbf{M}}^2 \\ &= e\,\|z\|_{\nabla^2 f(y)}^2 + p^p \|x-y\|_{\mathbf{M}}^{p-2} \|z\|_{\mathbf{M}}^2. \qquad (13)\end{aligned}$$

Define $g_y(x) = C_p \|x - y\|_{\mathbf{M}}^p$. We have

$$\begin{aligned}\nabla^2 g_y(x) &= pC_p \|x-y\|_{\mathbf{M}}^{p-2} \mathbf{M} \\ &\quad + p(p-2)C_p \|x-y\|_{\mathbf{M}}^{p-4} \mathbf{M}(x-y)(x-y)^\top \mathbf{M}\end{aligned}$$

and thus

$$pC_p \|x-y\|_{\mathbf{M}}^{p-2} \mathbf{M} \preceq \nabla^2 g_y(x).$$

Combining the two inequalities yields

$$\begin{aligned}\nabla^2 f_y(x) &= \nabla^2 f(x) + \nabla^2 g_y(x) \\ &\leq e(\nabla^2 h_y(x) - \nabla^2 g_y(x)) + \frac{1}{ep}\nabla^2 g_y(x) + \nabla^2 g_y(x) \\ &\leq e\nabla^2 h_y(x).\end{aligned}$$

For the lower bound, we exchange x and y in Equation (13) and obtain

$$z^\top \nabla^2 f(x) z \geq \frac{1}{e} z^\top \nabla^2 f(y) z - \frac{p^p}{e} \|x-y\|_{\mathbf{M}}^{p-2} \|z\|_{\mathbf{M}}^2.$$

Consequently,

$$\begin{aligned}\nabla^2 f_y(x) &= \nabla^2 f(x) + \nabla^2 g_y(x) \\ &\geq \frac{1}{e}(\nabla^2 h_y(x) - \nabla^2 g_y(x)) - \frac{1}{ep}\nabla^2 g_y(x) + \nabla^2 g_y(x) \\ &\geq \frac{1}{e}\nabla^2 h_y(x).\end{aligned}$$

□

4.2 Efficient implementation of proximal subproblems

We now leverage Lemma 4.3 to give an efficient oracle for the problem

$$\text{Prox}(y) = \arg\min_x \|\mathbf{A}x - b\|_p^p + ep^p \|x - y\|_{\mathbf{M}}^p.$$

Our algorithm is based on the *relative smoothness* framework from [48]. We use the following:

Lemma 4.4 (Theorem 3.1 from [48]). *Let f, h be convex twice-differentiable functions satisfying*

$$\mu\nabla^2 h(x) \preceq \nabla^2 f(x) \preceq L\nabla^2 h(x)$$

for all x. There is an algorithm which given a point x_0 computes a point x with

$$f(x) - \arg\min_y f(y) \leq \varepsilon\left(f(x_0) - \arg\min_y f(y)\right)$$

in $O(\frac{L}{\mu}\log(1/\varepsilon))$ iterations, where each iteration requires computing gradients of f and h at a point, $O(n)$ additional work, and solving a subproblem of the form

$$\min\{\langle g, x\rangle + Lh(x)\} \qquad (14)$$

for vectors g.

Applying this to the p-norm regression objective yields the following result.

Lemma 4.5. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}, b \in \mathbb{R}^n$ be given. Let $\mathbf{M} = \mathbf{A}^\top \mathbf{W}^{1-2/p}\mathbf{A}$ (Definition 4.2). There exists an algorithm which computes*

$$\arg\min_x \|\mathbf{A}x - b\|_p^p + ep^p \|x - y\|_{\mathbf{M}}^p$$

to high accuracy using $\widetilde{O}(1)$ linear system solves on matrices $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for $\mathbf{D} \geq 0$, i.e. in $\widetilde{O}(\mathcal{T}_A)$ time.

Proof. For the function $f_y(x) = \|\mathbf{A}x - b\|_p^p + ep^p \|x - y\|_{\mathbf{M}}^p$, we define the regularizer $h_y(x) = \|x - y\|_{\nabla^2 f(y)}^2 + ep^p \|x - y\|_{\mathbf{M}}^p$. We observe by Lemma 4.3 that $\nabla^2 f_y(x) \approx_{O(1)} \nabla^2 h_y(x)$ for all x. Thus Lemma 4.4 ensures we compute a minimizer to f_y using $\widetilde{O}(1)$ calls to an oracle which solves subproblems of the form

$$\min\left\{\langle g, x - z\rangle + 4\left(\|x - y\|_{\nabla^2 f(y)}^2 + ep^p \|x - y\|_{\mathbf{M}}^p\right)\right\}.$$

To solve this problem, we employ the algebra fact that $\frac{1}{s}x^s = \max_{y\geq 0} xy - \frac{1}{r}y^r$ for any $x \geq 0$ and $\frac{1}{s} + \frac{1}{r} = 1$. Thus, we have

$$\begin{aligned}\|x - y\|_{\mathbf{M}}^p &= \frac{p}{2}\cdot\frac{2}{p}\left(\|x - y\|_{\mathbf{M}}^2\right)^{p/2} \\ &= \frac{p}{2}\max_{\tau\geq 0}\left\{\tau\|x - y\|_{\mathbf{M}}^2 - \frac{p-2}{p}\tau^{\frac{p}{p-2}}\right\}.\end{aligned}$$

We may therefore write the subproblem as

$$\begin{aligned}\min_x \max_{\tau\geq 0}\Big\{ &\langle g, x - z\rangle + 4\|x - y\|_{\nabla^2 f(y)}^2 \\ &+ 2ep^{p+1}\left(\tau\|x - y\|_{\mathbf{M}}^2 - \frac{p-2}{p}\tau^{\frac{p}{p-2}}\right)\Big\}.\end{aligned}$$

This problem is convex in x and concave in τ: we may exchange the min and max above. Further, this is a convex quadratic in x, and thus for any fixed τ we may compute the minimizing x with a single linear system solve of the form $\nabla^2 f(y) + C\mathbf{M} = \mathbf{A}^\top\mathbf{D}\mathbf{A}$, for some constant $C \geq 0$ and $\mathbf{D} \geq 0$. Further, for any $C > 0$ we have $\nabla^2 f(y) + C\mathbf{M} > 0$: for any fixed $\tau > 0$ the minimizing value of x is unique. We conclude by binary searching for τ to high accuracy. Thus, each proximal subproblem may be solved using $\widetilde{O}(\mathcal{T}_\mathbf{A})$ time. □

We note that a high-accuracy solution to the proximal problem in Lemma 4.5 gives an approximate stationary point (exactly the condition later in Definition 4.6).

4.3 Putting it all together

We finish by using the above subroutine in the acceleration framework of [15]. We summarize the main claim here:

Definition 4.6 (Approximate Proximal Step Oracle, Definition 5 [15]). *We call $\mathcal{O}_{prox}$ an (α, δ)-approximate proximal oracle for convex $f : \mathbb{R}^d \to \mathbb{R}$ if, when queried at any $x \in \mathbb{R}^d$ it returns $y = \mathcal{O}_{prox}(x) \in \mathbb{R}^d$ such that*

$$\left\|\nabla f(y) + ep^{p+1}\|y - x\|_{\mathbf{M}}^{p-2}\cdot\mathbf{M}(y - x)\right\|_{\mathbf{M}^{-1}} \leq e\alpha p^{p+1}\|x - y\|_{\mathbf{M}}^{p-1} + \delta$$

Theorem 4 (Theorem 7 from [15]). *Let $g : \mathbb{R}^d \to \mathbb{R}$ be a convex twice-differentiable function minimized at $x_\star$, and let x_0 be a point with $\|x_0 - x_\star\| \leq R$. For any parameter $\varepsilon \geq 0$, there is an algorithm which for all k computes x with*

$$g(x) - g(x_\star) \leq \max\left\{\varepsilon, \frac{100p^p\cdot 40^{p-2}R^p}{k^{\frac{3p-2}{2}}}\right\}$$

using $\lceil k(6+\log_2[10^{20}R^p\cdot(10^5 p)^{p+6}\varepsilon^{-1}])^2\rceil = O(p^2 k\log^2(pR\varepsilon^{-1/p}))$ gradients of f and queries to an $(\frac{1}{128p^2}, \delta)$-approximate proximal oracle, provided that both $\delta \leq \varepsilon/[10^{20}p^2R]$ and $\varepsilon \leq 10^{20}p^p\gamma^4R^{p+1}$.

Proof. Define the convex function $g(x) = f(\mathbf{M}^{-1/2}x)$, and choose $\omega(x) = ep^p x^{p-2}$. The optimality conditions of Definition 4.6 are equivalent to those in Definition 5 of [15] after applying this change of basis. Theorem 4 then follows from applying Theorem 7 in [15] to g with $\gamma = p$ and $\alpha = \frac{1}{128p^2}$. □

Our application of this fact relies on a diameter-shrinking argument from [18]. We first recall a standar bound on the strong convexity of $\|x\|_p^p$, which we cite from [3] for simplicity.

Lemma 4.7 (Lemma 4.5 from [3]). *Let $p \in (1, \infty)$. Then for any two vectors $y, \Delta \in \mathbb{R}^n$,*

$$\|y\|_p^p + v^\top\Delta + \frac{p-1}{p2^p}\|\Delta\|_p^p \leq \|y + \Delta\|_p^p$$

where $v_i = p|y|_i^{p-2}y_i$ is the gradient of $\|y\|_p^p$.

We finally need the following lemma which allows us to convert points which low function error into points with small distance to the minimizer.

Corollary 4.8. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}, b \in \mathbb{R}^n$ be given. Let $\mathbf{M} = \mathbf{A}^\top\mathbf{W}^{1-2/p}\mathbf{A}$ for ℓ_p Lewis weight overestimates w (Definition 2.3). Let $f(x) = \|\mathbf{A}x - b\|_p^p$ be minimized at $x_\star$. If x satisfies $f(x) - f(x_\star) \leq \mathcal{E}$, then $\|x - x_\star\|_{\mathbf{M}} \leq 2^{3/2}d^{1/2-1/p}\mathcal{E}^{1/p}$.*

Proof. Applying Lemma 4.7, we have

$$\begin{aligned}&\|\mathbf{A}x - b\|_p^p = \|\mathbf{A}x_\star - b + \mathbf{A}(x - x_\star)\|_p^p \\ &\geq \|\mathbf{A}x_\star - b\|_p^p + \nabla f(x_\star)^\top(x - x_\star) + \frac{p-1}{p2^p}\|\mathbf{A}(x - x_\star)\|_p^p.\end{aligned}$$

Note $\nabla f(x_\star) = 0$ by optimality of $x_\star$. By Equation (2), we obtain

$$\|x - x_\star\|_{\mathbf{M}}^2 \leq \|w\|_1^{1-2/p}\|\mathbf{A}(x - x_\star)\|_p^2.$$

by Hölder's inequality and the fact that $p \geq 2$. Recall that $\|w\|_1 \leq 2d$ by Lemma 2.5: this implies

$$\|x - x_\star\|_{\mathbf{M}}^p \leq (2d)^{p/2-1}\|\mathbf{A}(x - x_\star)\|_p^p.$$

Thus,

$$\begin{aligned}\mathcal{E} \geq f(x) - f(x_\star) &\geq \frac{p-1}{p2^p}\|\mathbf{A}(x - x_\star)\|_p^p \\ &\geq 2^{-p-1}(2d)^{1-p/2}\|x - x_\star\|_{\mathbf{M}}^p:\end{aligned}$$

taking p^{th} roots yields $\|x - x_\star\|_{\mathbf{M}} \leq 2^{3/2}d^{1/2-1/p}\mathcal{E}^{1/p}$ as desired. □

We now prove the main decrease lemma which in turn shows Theorem 1.

Lemma 4.9. *Let $\mathbf{A}, b$ be given, and let $f(x) = \|\mathbf{A}x - b\|_p^p$ have minimizer $x_\star$. Let x_0 be a point such that $f(x_0) - f(x_\star) \le \mathcal{E}$. There is an algorithm which returns x' with $f(x') - f(x_\star) \le \frac{\mathcal{E}}{2}$ using*

$$\widetilde{O}\left(p^{8/3} d^{\frac{p-2}{3p-2}} \mathcal{T}_{\mathbf{A}}\right)$$

time.

Proof. Applying Corollary 4.8 yields

$$R \equiv \|x_0 - x_\star\|_{\mathbf{M}} \le 2^{3/2} d^{1/2-1/p} \mathcal{E}^{1/p}$$

for $\mathbf{M} = \mathbf{A}^\top \mathbf{W}^{1-2/p} \mathbf{A}$. Note that $\log(pR\mathcal{E}^{-1/p}) = O(\log d)$. We now apply Theorem 4 to $f(x)$ with $\varepsilon = \frac{\mathcal{E}}{2}$: in $\widetilde{O}(p^2 k)$ gradient computations and proximal oracle calls we compute x with

$$f(x) - f(x_\star) \le \max\left\{\frac{1}{2}\mathcal{E}, \frac{100 p^p \cdot 40^{p-2} R^p}{k^{\frac{3p-2}{2}}}\right\} \le \max\left\{\frac{1}{2}\mathcal{E}, \frac{100(120p)^p d^{p/2-1}\mathcal{E}}{k^{\frac{3p-2}{2}}}\right\}.$$

For $k = O(p^{2/3} d^{\frac{p-2}{3p-2}})$, this bound is $\frac{1}{2}\mathcal{E}$ as desired. We additionally require $\widetilde{O}(p^2 k) = \widetilde{O}(p^{8/3} d^{\frac{p-2}{3p-2}})$ gradient computations and calls to a proximal oracle for f – these proximal oracle calls can each be implemented in $\widetilde{O}(\mathcal{T}_{\mathbf{A}})$ time by Lemma 4.5. □

Proof of Theorem 1. Let $x_\star = \arg\min_y \|\mathbf{A}y - b\|_p^p$ and $\mathrm{OPT} = \|\mathbf{A}x_\star - b\|_p^p$. We may initialize $\mathcal{E} = n^{\frac{p-2}{2}} \mathrm{OPT}$ in Lemma 4.9 by setting $x = \arg\min_x \|\mathbf{A}y - b\|_2^p$ instead, and noting that

$$\|\mathbf{A}y - b\|_p^p \le \|\mathbf{A}y - b\|_2^p \le \|\mathbf{A}x_\star - b\|_2^p \le n^{\frac{p-2}{2}} \|\mathbf{A}x_\star - b\|_p^p = n^{\frac{p-2}{2}} \mathrm{OPT}.$$

Now Theorem 1 follows from running $\log(n^{\frac{p-2}{2}}) = \widetilde{O}(p)$ iterations of Lemma 4.9. □

Discussion on numerical stability. Throughout the section (eg. in the application of Lemmas 4.4 and 4.5), we have assumed that high accuracy solutions to problems lead to exact or high accuracy stationary points, i.e. the KKT conditions are satisfied. There are several ways to make this rigorous. In particular, if one assumes that all parameters, including the condition number of $\mathbf{A}$, are quasipolynomially bounded (i.e. at most $\exp(\mathrm{poly}\log m)$), then one can add a small strongly-convex regularizer (eg. $\delta\|x\|_{\mathbf{A}}^2$ for $\delta \le \exp(-\mathrm{poly}\log m)\varepsilon$) which barely affects the optimal value. Strong convexity allows us to get an approximate stationary point from approximate minimizers, which suffices for the all our applications (including the proof of [48]).

5 ALGORITHM FOR SMALL q

In this section, we provide an algorithm to show Theorem 3. Because there isn't a clean version of iterative refinement for the objective $\|\mathbf{A}x - b\|_q$ for $q < 2$, we instead work with the dual problem. Precisely, we can use Sion's minimax theorem to get for $p = q/(q-1)$

$$\min_{x\in\mathbb{R}^d} \|\mathbf{A}x - b\|_q = \min_{x\in\mathbb{R}^d} \max_{\|y\|_p \le 1} y^\top(\mathbf{A}x - b) = \max_{\|y\|_p\le 1} \min_{x\in\mathbb{R}^d} y^\top(\mathbf{A}x - b) = -\min_{\substack{\|y\|_p\le 1 \\ \mathbf{A}^\top y = 0}} b^\top y = \left(\min_{\mathbf{A}^\top y = 0, b^\top y = 1} \|y\|_p\right)^{-1}. \quad (15)$$

Using an high precision solution y to (15), we can return a high precision minimizer to $\min_{x\in\mathbb{R}^d} \|\mathbf{A}x - b\|_q$. In particular for the true optimum y^*, by KKT conditions (that $\nabla\|y^*\|_p^p = p\,\mathrm{sign}(y^*)|y^*|^{p-2}$ is in the kernel of $\begin{bmatrix}\mathbf{A} & b\end{bmatrix}^\top$) we know that there exists a vector $x \in \mathbb{R}^d$ satisfying $\lambda\,\mathrm{sign}(y^*)|y^*|^{p-2} = \mathbf{A}x - b$. We return this x. If we have a high precision minimizer y instead of the true optimum y^*, we can instead return an ℓ_2-projection, i.e. $x = \arg\min_{x\in\mathbb{R}^d} \|\mathbf{A}x - b - \lambda\,\mathrm{sign}(y)|y|^{p-2}\|_2$ for the proper scaling λ.

We use Lemma 2.11 (for $\mathbf{A} = \mathbf{I}$ and $b = 0$ in the lemma statement) to solve the problem in (15), where we assume $v = 1$ and $\mathrm{OPT} = 1$ by scaling. This leads to the following optimization problem for some $g \in \mathbb{R}^n$ and $\mathbf{R} = \mathbf{diag}(r)$ for $r \in \mathbb{R}^n_{>0}$:

$$\min_{\mathbf{A}^\top x = 0, b^\top x = 1, g^\top x = -1} x^\top \mathbf{R} x + \|x\|_p^p.$$

Let $\mathbf{U} = \begin{bmatrix}\mathbf{A} & b & g\end{bmatrix}$ and $v = \begin{bmatrix}\mathbf{0} & 1 & -1\end{bmatrix}^\top$. Through these reductions and Lemma 2.11 it suffices to show the following.

Lemma 5.1. *For matrix $\mathbf{U} \in \mathbb{R}^{n\times d}$ and $v \in \mathbb{R}^n$, assume there is $x \in \mathbb{R}^n$ satisfying $\mathbf{U}^\top x = v$, $x^\top \mathbf{R} x \le 1$ and $\|x\|_p^p \le 1$. Then there is an algorithm that in time $\widetilde{O}(\mathcal{T}_{\mathbf{U}})$ outputs a $y \in \mathbb{R}^n$ satisfying $\mathbf{U}^\top y = v$, $y^\top \mathbf{R} y = O(1)$, and $\|y\|_p \le O(d^{\frac{p-2}{2p-2}})$.*

We remark that we could also instead get a result which achieves a $y^\top \mathbf{R} y = O_p(1)$ and $\|y\|_p^p = O_p(1)$ in time $\widetilde{O}_p(\mathcal{T}_{\mathbf{U}} d^{\frac{p-2}{2p-2}})$ via a multiplicative weights style algorithm as done in Section 3 algorithm Oracle (Line 11). However since both runtimes would be the same (up to logarithmic factors), we choose to present this simpler single iteration algorithm. Combining this multiplicative weights style algorithm with energy boosting as in the analysis in Section 3 to achieve a $O_p(d^{\frac{p-2}{3p-2}})$ iteration bound remains an interesting open problem.

Proof of Theorem 3. Lemma 5.1 satisfies Lemma 2.11 for $\gamma = O(d^{\frac{p-2}{2p-2}})$. Each call to Lemma 5.1 requires $\widetilde{O}(1)$ calls to a solver for $\mathbf{U}^\top\mathbf{D}\mathbf{U}$. Also, a solver for $\mathbf{U}^\top\mathbf{D}\mathbf{U}$ can be implemented using $O(1)$ calls to a solver for $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for diagonal matrices $\mathbf{D}$, and $O(1)$ solves on $O(1)\times O(1)$ matrices by computing the inverse via the Schur complement onto the 2×2 block of $\mathbf{U}^\top\mathbf{D}\mathbf{U}$ corresponding to the b, g vectors. Thus the total number of iterations is $\widetilde{O}(p^{3.5}\gamma\log(m/\varepsilon)) = \widetilde{O}_p(d^{\frac{p-2}{2p-2}}\log(m/\varepsilon))$ as desired. □

An algorithm to show a weaker version of Lemma 5.1 with the bound $\|y\|_p \le O\left(n^{\frac{p-2}{2p-2}}\right)$ was given in [4, Lemma 3.3], by simply returning

$$y = \arg\min_{\mathbf{U}^\top x = v} x^\top \left(n^{1-\frac{2}{p}}\mathbf{R} + \mathbf{I}\right) x.$$

Our approach to improve this dependence to $O(d^{\frac{p-2}{2p-2}})$ uses a version of ℓ_q Lewis weights to replace the identity matrix $\mathbf{I}$ in the above. To handle the presence of the resistance term $\mathbf{R}$ we require a regularized version of Lewis weights.

Definition 5.2 (Regularized ℓ_q Lewis weights). *For a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, $1 \le q \le 2$, and vector $c \in \mathbb{R}^n_{\ge 0}$, the c-regularized ℓ_q Lewis weights w are defined as the solution to*

$$w_i = \sigma\left((\mathbf{C} + \mathbf{W})^{\frac{1}{2}-\frac{1}{q}}\mathbf{A}\right)_i \quad \text{for all} \quad i \in [n].$$

We show that these weights can be computed approximately in $\widetilde{O}(1)$ iterations of a contractive map. Each iteration requires the computation of approximate leverage scores. The proof of the following lemma is deferred to the full version.

Lemma 5.3. *Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, $1 \le q \le 2$, and vector $c \in \mathbb{R}^n_{\ge 0}$, let w be the c-regularized ℓ_q Lewis weights. There is an algorithm ApproxRegLewis$(\mathbf{A}, c, q)$ that whp. computes a vector $\widehat{w} \in \mathbb{R}^n_{\ge 0}$ satisfying $\widehat{w}_i/w_i \in [0.9, 1.1]$ for all $i \in [n]$ in $\widetilde{O}(\mathcal{T}_\mathbf{A})$ time.*

We can now give our algorithm to show Lemma 5.1. We can now show Lemma 5.1.

Algorithm 2: OracleSmall$(\mathbf{U}, v, r, q)$. Given matrix $\mathbf{U} \in \mathbb{R}^{n\times d}, v \in \mathbb{R}^n, r \in \mathbb{R}^n, q \le 2$, such that there exists x with $\mathbf{U}^\top x = v$ and $x^\top \mathbf{R} x \le 1$ and $\|x\|_p^p \le 1$, returns y satisfying $\mathbf{U}^\top y = v$, $y^\top \mathbf{R} y = O(1)$ and $\|y\|_p = O(d^{\frac{p-2}{2p-2}})$.

1 $\widehat{w} \leftarrow$ ApproxRegLewis$(\mathbf{U}, dr^{\frac{p}{p-2}}, q)$. ▷ Lemma 5.3

2 Return $y \leftarrow \arg\min_{\mathbf{U}^\top x = v} x^\top \left(d^{1-\frac{2}{p}}\mathbf{R} + \widehat{\mathbf{W}}^{1-\frac{2}{p}}\right) x$.

Proof of Lemma 5.1. Let x satisfy $\mathbf{U}^\top x = v$ and $x^\top \mathbf{R} x \le 1$ and $\|x\|_p \le 1$. By the definition of y in line 2 in OracleSmall (Algorithm 2), we know that

$$y^\top \left(d^{1-\frac{2}{p}}\mathbf{R} + \widehat{\mathbf{W}}^{1-\frac{2}{p}}\right) y \le x^\top \left(d^{1-\frac{2}{p}}\mathbf{R} + \widehat{\mathbf{W}}^{1-\frac{2}{p}}\right) x$$
$$\le d^{1-\frac{2}{p}} + \|w\|_1^{1-\frac{2}{p}} \|x\|_p^2 \le 3d^{1-\frac{2}{p}}$$

where we have used Hölder's inequality and $\|\widehat{w}\|_1 \le 1.1d$ by Lemma 5.3. In particular, this gives us $d^{1-\frac{2}{p}} y^\top \mathbf{R} y \le 3d^{1-\frac{2}{p}}$ so $y^\top \mathbf{R} y \le 3$. Now we bound $\|y\|_p^p$. Note that by the optimality conditions for y (that the gradient of the objective of line 2 of Algorithm 2 is in the kernel of $\mathbf{U}^\top$), there must exist a vector $z \in \mathbb{R}^d$ such that

$$y = \left(d^{1-\frac{2}{p}}\mathbf{R} + \widehat{\mathbf{W}}^{1-\frac{2}{p}}\right)^{-1} \mathbf{U}z.$$

Define $\widehat{y} = \left(d^{1-\frac{2}{p}}\mathbf{R} + \widehat{\mathbf{W}}^{1-\frac{2}{p}}\right)^{-1/2} \mathbf{U}z$, so that

$$\|\widehat{y}\|_2^2 = y^\top \left(d^{1-\frac{2}{p}}\mathbf{R} + \widehat{\mathbf{W}}^{1-\frac{2}{p}}\right) y \le 3d^{1-\frac{2}{p}}.$$

Now we get that

$$\|y\|_p^p = \sum_{i\in[n]} \left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-p} |(\mathbf{U}z)_i|^p$$

$$= \sum_{i\in[n]} \left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-p} \left(dr_i^{\frac{p}{p-2}} + \widehat{w}_i\right)^{\frac{(p-2)^2}{2p}} \cdot \tag{16}$$
$$\left|\left(\left(d\mathbf{R}^{\frac{p}{p-2}} + \widehat{\mathbf{W}}\right)^{\frac{1}{2}-\frac{1}{q}} \mathbf{U}z\right)_i\right|^{p-2} (\mathbf{U}z)_i^2$$

$$\overset{(i)}{\le} \sum_{i\in[n]} \left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-1} \widehat{w}_i^{-\frac{p-2}{2}} \cdot \tag{17}$$
$$\left|\left(\left(d\mathbf{R}^{\frac{p}{p-2}} + \widehat{\mathbf{W}}\right)^{\frac{1}{2}-\frac{1}{q}} \mathbf{U}z\right)_i\right|^{p-2} (\mathbf{U}z)_i^2$$

$$\overset{(ii)}{\le} \left\|\left(d\mathbf{R}^{\frac{p}{p-2}} + \widehat{\mathbf{W}}\right)^{\frac{1}{2}-\frac{1}{q}} \mathbf{U}z\right\|_2^{\frac{p-2}{2}} \cdot \tag{18}$$

$$\sum_{i\in[n]} \left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-1} \widehat{w}_i^{-\frac{p-2}{2}} \sigma\left(\left(d\mathbf{R}^{\frac{p}{p-2}} + \widehat{\mathbf{W}}\right)^{\frac{1}{2}-\frac{1}{q}} \mathbf{U}\right)_i^{\frac{p-2}{2}} (\mathbf{U}z)_i^2 \tag{19}$$

$$\overset{(iii)}{\le} 2^{\frac{p-2}{2}} \left\|\left(d\mathbf{R}^{\frac{p}{p-2}} + \widehat{\mathbf{W}}\right)^{\frac{1}{2}-\frac{1}{q}} \mathbf{U}z\right\|_2^{\frac{p-2}{2}} \cdot \tag{20}$$

$$\sum_{i\in[n]} \left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-1} (\mathbf{U}z)_i^2$$

$$\overset{(iv)}{\le} 4^{\frac{p-2}{2}} \|\widehat{y}\|_2^p \le 4^{\frac{p-2}{2}} \left(3d^{1-\frac{2}{p}}\right)^{\frac{p}{2}} \tag{21}$$

$$\le 4^p d^{\frac{p}{2}-1}. \tag{22}$$

Here, (i) follows from the inequality $a^{1-2/p} + b^{1-2/p} \ge (a+b)^{1-2/p}$, which holds for all $a, b \ge 0$, for $a = d_i r_i^{\frac{p}{p-2}}$ and $b = \widehat{w}_i$, and the trivial bound

$$\left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-p/2} \le \widehat{w}_i^{-\frac{p-2}{2}}.$$

Also, (ii) is shown using Fact 2.2, and (iii) uses the definition of c-regularized Lewis weights (Definition 5.2) for $c = dr^{\frac{p}{p-2}}$ as chosen in line 1 of OracleSmall (Algorithm 2) and that $\widehat{w}$ are 1.1-approximate weights by Lemma 5.3. Finally, (iv) uses the definition of $\widehat{y}$ and

$$(dr_i^{\frac{p}{p-2}} + \widehat{w}_i)^{1-\frac{2}{q}} = (dr_i^{\frac{p}{p-2}} + \widehat{w}_i)^{\frac{2}{p}-1}$$
$$\le \max\left(d^{1-\frac{2}{p}} r_i, \widehat{w}_i^{1-\frac{2}{p}}\right)^{-1} \le 2\left(d^{1-\frac{2}{p}} r_i + \widehat{w}_i^{1-\frac{2}{p}}\right)^{-1}.$$

Taking p-th roots of (22) shows that $\|y\|_p \le O(d^{\frac{p-2}{2p-2}})$ as desired. To bound the cost, note that line 1 and 2 of OracleSmall (Algorithm 2) call $\widetilde{O}(1)$ solves to $\mathbf{U}^\top \mathbf{D} \mathbf{U}$ for diagonal matrices $\mathbf{D}$ by Lemma 5.3. □

ACKNOWLEDGEMENTS

We thank Michael B. Cohen, Yair Carmon, Qijia Jiang, Yujia Jin, Yin Tat Lee, Kevin Tian, and Richard Peng for helpful discussions. We also would like to thank anonymous reviewers for several helpful comments in improving the presentation of the paper.

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. Yang P. Liu was supported by the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship, and NSF CAREER Award CCF-1844855 and NSF Grant CCF-1955039.

REFERENCES

[1] Deeksha Adil, Brian Bullins, Rasmus Kyng, and Sushant Sachdeva. 2021. Almost-Linear-Time Weighted ℓ_p-Norm Solvers in Slightly Dense Graphs via Sparsification. In *ICALP (LIPIcs, Vol. 198)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 9:1–9:15.

[2] Deeksha Adil, Brian Bullins, and Sushant Sachdeva. 2021. Unifying Width-Reduced Methods for Quasi-Self-Concordant Optimization. *arXiv preprint arXiv:2107.02432* (2021).

[3] Deeksha Adil, Rasmus Kyng, Richard Peng, and Sushant Sachdeva. 2019. Iterative Refinement for ℓ_p-norm Regression. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*. 1405–1424.

[4] Deeksha Adil, Richard Peng, and Sushant Sachdeva. 2019. Fast, Provably convergent IRLS Algorithm for p-norm Linear Regression. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 14166–14177. http://papers.nips.cc/paper/9565-fast-provably-convergent-irls-algorithm-for-p-norm-linear-regression

[5] Deeksha Adil and Sushant Sachdeva. 2020. Faster p-norm minimizing flows, via smoothed q-norm problems. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*. 892–910.

[6] Naman Agarwal and Elad Hazan. 2018. Lower bounds for higher-order convex optimization. In *Conference On Learning Theory*. PMLR, 774–792.

[7] Yossi Arjevani, Ohad Shamir, and Ron Shiff. 2019. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming* 178, 1 (2019), 327–360.

[8] Kyriakos Axiotis, Aleksander Mądry, and Adrian Vladu. 2020. Circulation Control for Faster Minimum Cost Flow in Unit-Capacity Graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. 93–104.

[9] Francis Bach. 2010. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics* 4 (2010), 384–414.

[10] Jan van den Brand. 2020. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 259–278.

[11] Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. 2021. Minimum cost flows, MDPs, and ℓ_1-regression in nearly linear time for dense instances. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*. 859–869.

[12] Jan van den Brand, Yin Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. 2020. Bipartite Matching in Nearly-linear Time on Moderately Dense Graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. 919–930. https://doi.org/10.1109/FOCS46700.2020.00090 Available at https://arxiv.org/pdf/2101.05719.pdf.

[13] Jan van den Brand, Yin Tat Lee, Aaron Sidford, and Zhao Song. 2020. Solving Tall Dense Linear Programs in Nearly Linear Time. In *STOC*. https://arxiv.org/pdf/2002.02304.pdf.

[14] Sébastien Bubeck, Michael B. Cohen, Yin Tat Lee, and Yuanzhi Li. 2018. An homotopy method for ℓ_p regression provably beyond self-concordance and in input-sparsity time. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 1130–1137.

[15] Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. 2019. Complexity of Highly Parallel Non-Smooth Convex Optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 13900–13909.

[16] Brian Bullins. 2018. Fast minimization of structured convex quartics. *arXiv preprint arXiv:1812.10349* (2018).

[17] Brian Bullins and Richard Peng. 2019. Higher-order accelerated methods for faster non-smooth optimization. *arXiv preprint arXiv:1906.01621* (2019).

[18] Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. 2020. Acceleration with a Ball Optimization Oracle. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[19] Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. 2021. Thinking inside the ball: Near-optimal minimization of the maximal loss. *arXiv preprint arXiv:2105.01778* (2021).

[20] Hui Han Chin, Aleksander Mądry, Gary L. Miller, and Richard Peng. 2013. Runtime guarantees for regression problems. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, Robert D. Kleinberg (Ed.). ACM, 269–282. https://doi.org/10.1145/2422436.2422469

[21] Paul Christiano, Jonathan A. Kelner, Aleksander Mądry, Daniel A. Spielman, and Shang-Hua Teng. 2011. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, Lance Fortnow and Salil P. Vadhan (Eds.). ACM, 273–282. https://doi.org/10.1145/1993636.1993674

[22] Kenneth Clarkson, Ruosong Wang, and David Woodruff. 2019. Dimensionality reduction for tukey regression. In *International Conference on Machine Learning*. PMLR, 1262–1271.

[23] Kenneth L Clarkson. 2005. Subgradient and sampling algorithms for l 1 regression.

[24] Kenneth L Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, Xiangrui Meng, and David P Woodruff. 2016. The fast cauchy transform and faster robust linear regression. *SIAM J. Comput.* 45, 3 (2016), 763–810.

[25] Kenneth L. Clarkson and David P. Woodruff. 2013. Low rank approximation and regression in input sparsity time. In *STOC*. ACM, 81–90.

[26] Michael B. Cohen, Ben Cousins, Yin Tat Lee, and Xin Yang. 2019. A near-optimal algorithm for approximating the John Ellipsoid. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*. 849–873.

[27] Michael B Cohen, Yin Tat Lee, and Zhao Song. 2019. Solving Linear Programs in the Current Matrix Multiplication Time. In *STOC*. https://arxiv.org/pdf/1810.07896.

[28] Michael B Cohen, Aleksander Mądry, Piotr Sankowski, and Adrian Vladu. 2017. Negative-Weight Shortest Paths and Unit Capacity Minimum Cost Flow in $O(m^{10/7} \log W)$ Time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 752–771.

[29] Michael B. Cohen and Richard Peng. 2015. ℓ_p Row Sampling by Lewis Weights. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, Rocco A. Servedio and Ronitt Rubinfeld (Eds.). ACM, 183–192. https://doi.org/10.1145/2746539.2746567

[30] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W Mahoney. 2009. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.* 38, 5 (2009), 2060–2078.

[31] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. 2012. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.* 13 (2012), 3475–3506. http://dl.acm.org/citation.cfm?id=2503352

[32] David Durfee, Kevin A. Lai, and Saurabh Sawlani. 2018. ℓ_1 Regression using Lewis Weights Preconditioning and Stochastic Gradient Descent. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018 (Proceedings of Machine Learning Research, Vol. 75)*, Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet (Eds.). PMLR, 1626–1656. http://proceedings.mlr.press/v75/durfee18a.html

[33] Alina Ene and Adrian Vladu. 2019. Improved Convergence for ℓ_1 and ℓ_∞ Regression via Iteratively Reweighted Least Squares. In *International Conference on Machine Learning*. PMLR, 1794–1801.

[34] Alexander V. Gasnikov, Pavel E. Dvurechensky, Eduard A. Gorbunov, Evgeniya A. Vorontsova, Daniil Selikhanovych, César A. Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. 2019. Near Optimal Methods for Minimizing Convex Functions with Lipschitz p-th Derivatives. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA (Proceedings of Machine Learning Research, Vol. 99)*, Alina Beygelzimer and Daniel Hsu (Eds.). PMLR, 1392–1393.

[35] Mehrdad Ghadiri, Richard Peng, and Santosh S Vempala. 2021. Sparse Regression Faster than d^ω. *arXiv preprint arXiv:2109.11537* (2021).

[36] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. 2021. A faster algorithm for solving general LPs. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*. 823–832.

[37] Tarun Kathuria. 2020. A Potential Reduction Inspired Algorithm for Exact Max Flow in Almost $\widetilde{O}(m^{4/3})$ Time. *arXiv preprint arXiv:2009.03260* (2020).

[38] Tarun Kathuria, Yang P Liu, and Aaron Sidford. 2020. Unit Capacity Maxflow in Almost $O(m^{4/3})$ Time. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 119–130.
[39] Jonathan A. Kelner, Gary L. Miller, and Richard Peng. 2012. Faster approximate multicommodity flow using quadratically coupled flows. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*. 1–18.
[40] Rasmus Kyng, Richard Peng, Sushant Sachdeva, and Di Wang. 2019. Flows in almost linear time via adaptive preconditioning. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. 902–913.
[41] Yin Tat Lee and Aaron Sidford. 2015. Efficient Inverse Maintenance and Faster Algorithms for Linear Programming. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. 230–249.
[42] Yin Tat Lee and Aaron Sidford. 2019. Solving linear programs with Sqrt (rank) linear system solves. *arXiv preprint arXiv:1910.08033* (2019).
[43] Yin Tat Lee, Zhao Song, and Qiuyi Zhang. 2019. Solving Empirical Risk Minimization in the Current Matrix Multiplication Time. In *COLT*. https://arxiv.org/pdf/1905.04447.
[44] D. Lewis. 1978. Finite dimensional subspaces of L_p. *Studia Mathematica* 63, 2 (1978), 207–212. http://eudml.org/doc/218208
[45] Chih-Jen Lin, Ruby C Weng, and S Sathiya Keerthi. 2008. Trust region Newton method for large-scale logistic regression. *Journal of Machine Learning Research* 9, 4 (2008).
[46] Yang P Liu and Aaron Sidford. 2020. Faster divergence maximization for faster maximum flow. *arXiv preprint arXiv:2003.08929* (2020).
[47] Yang P. Liu and Aaron Sidford. 2020. Faster energy maximization for faster maximum flow. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*. 803–814.
[48] Haihao Lu, Robert M. Freund, and Yurii E. Nesterov. 2018. Relatively Smooth Convex Optimization by First-Order Methods, and Applications. *SIAM J. Optim.* 28, 1 (2018), 333–354.
[49] Aleksander Mądry. 2013. Navigating Central Path with Electrical Flows: From Flows to Matchings, and Back. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. IEEE Computer Society, 253–262. https://doi.org/10.1109/FOCS.2013.35
[50] Aleksander Mądry. 2016. Computing Maximum Flow with Augmenting Electrical Flows. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, Irit Dinur (Ed.). IEEE Computer Society, 593–602. https://doi.org/10.1109/FOCS.2016.70
[51] Xiangrui Meng and Michael W. Mahoney. 2013. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *STOC*. ACM, 91–100.
[52] Xiangrui Meng and Michael W. Mahoney. 2013. Robust Regression on MapReduce. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013 (JMLR Workshop and Conference Proceedings, Vol. 28)*. JMLR.org, 888–896. http://proceedings.mlr.press/v28/meng13b.html
[53] Renato DC Monteiro and Benar Fux Svaiter. 2013. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization* 23, 2 (2013), 1092–1125.
[54] Yurii Nesterov. 2019. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming* (2019), 1–27.
[55] Yurii E Nesterov. 1983. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl. akad. nauk Sssr*, Vol. 269. 543–547.
[56] Yurii E. Nesterov. 2009. Unconstrained Convex Minimization in Relative Scale. *Math. Oper. Res.* 34, 1 (2009), 180–193. https://doi.org/10.1287/moor.1080.0348
[57] James Renegar. 1988. A polynomial-time algorithm, based on Newton's method, for linear programming. *Math. Program.* 40, 1-3 (1988), 59–93.
[58] Daniel A. Spielman and Nikhil Srivastava. 2011. Graph Sparsification by Effective Resistances. *SIAM J. Comput.* 40, 6 (2011), 1913–1926.
[59] Pravin M. Vaidya. 1989. Speeding-Up Linear Programming Using Fast Matrix Multiplication (Extended Abstract). In *30th IEEE Annual Symposium on Foundations of Computer Science, FOCS 1989, Research Triangle Park, NC, USA, October 30 - November 1, 1989*. IEEE Computer Society, 332–337. https://doi.org/10.1109/SFCS.1989.63499
[60] Pravin M Vaidya. 1990. An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations. *Mathematical Programming* 47, 1-3 (1990), 175–201.
[61] Przemyslaw Wojtaszczyk. 1996. *Banach spaces for analysts*. Number 25. Cambridge University Press.
[62] David P. Woodruff and Qin Zhang. 2013. Subspace Embeddings and ℓ_p-Regression Using Exponential Random Variables. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*. 546–567.
[63] Jiyan Yang, Yinlam Chow, Christopher Ré, and Michael W. Mahoney. 2016. Weighted SGD for ℓ_p Regression with Randomized Preconditioning. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, Robert Krauthgamer (Ed.). SIAM, 558–569. https://doi.org/10.1137/1.9781611974331.ch41

A LEWIS WEIGHTS FOR ℓ_∞ REGRESSION

Here, we argue that using ℓ_∞ Lewis weight overestimates (Definition 2.4) along with the computations and framework of [18] directly give an algorithm for ℓ_∞-regression, proving Theorem 2. We use the notion of *quasi-self-concordance* from [18, Definition 10].

Lemma A.1. *Define* $\mathrm{lse}_t : \mathbb{R}^n \to \mathbb{R}$ *as*

$$\mathrm{lse}_t(x) = t \log\left(\sum_{i\in[n]} \exp(x_i/t)\right).$$

If $w \in \mathbb{R}^n_{\geq 0}$ *are* ℓ_∞ *Lewis weight overestimates (Definition 2.4) then* $\mathrm{lse}_t(x)$ *is* $1/t$*-smooth and* $2/t$*-quasi-self-concordant in the norm* $\|\cdot\|_{\mathbf{A}^\top\mathbf{W}\mathbf{A}}$.

Proof. By scaling, it suffices to show the result for $t = 1$. The computations in the proof of [18, Lemma 14] in [18, Appendix G.1] shows that $u^\top \nabla^2 \mathrm{lse}(x) u \leq \|u\|_\infty^2$. Now by Lemma 2.6 for $p = \infty$ we get $\|u\|_\infty \leq \|u\|_{\mathbf{A}^\top\mathbf{W}\mathbf{A}}$ which implies the smoothness claim.

To show quasi-self-concordance, we use the computations in the proof of [18, Lemma 14] in [18, Appendix G.1] to get

$$\left|\nabla^3 \mathrm{lse}(x)[u,u,h]\right| \leq \|u\|^2_{\nabla^2 \mathrm{lse}(x)} \|h\|_\infty.$$

The proof follows from the fact that $\|h\|_\infty \leq \|h\|_{\mathbf{A}^\top\mathbf{W}\mathbf{A}}$ by Lemma 2.6 with $p = \infty$. □

We can plug this bound into [18, Corollary 12] to show Theorem 2.

Proof of Theorem 2. Define $x_\star = \arg\min_x \|\mathbf{A}x - b\|_\infty$ and $\mathrm{OPT} = \|\mathbf{A}x_\star - b\|_\infty$. We assume that we start at a point $x \in \mathbb{R}^n$ with $\|\mathbf{A}x - b\|_\infty \leq 2\mathrm{OPT}$. Otherwise, the same proof shows that given any upper bound on OPT, the algorithm allows us to reduce the error by a constant factor. We can initialize with polynomial error by solving the ℓ_2 problem $\min_{x\in\mathbb{R}^d} \|\mathbf{A}x - b\|_\infty$. We set $t = \frac{\varepsilon \mathrm{OPT}}{20 \log n}$ (which loses $\varepsilon\mathrm{OPT}/2$ additive function accuracy) and minimize $\mathrm{lse}_t(\mathbf{A}x - b)$ to $\varepsilon\mathrm{OPT}/2$ accuracy. In [18, Corollary 12] we set $\mathbf{M} = \mathbf{A}^\top\mathbf{W}\mathbf{A}$ and $M = 2/t = O\left(\frac{20\log n}{\varepsilon \mathrm{OPT}}\right)$ by Lemma A.1.

We show we can set $R = O(\mathrm{OPT}\sqrt{d})$. Indeed note that $\|\mathbf{A}(x - x^*)\|_\infty \leq \|\mathbf{A}x - b\|_\infty + \|\mathbf{A}x_\star - b\|_\infty \leq 3\mathrm{OPT}$. Thus we get

$$R^2 = \|x - x_\star\|^2_{\mathbf{A}^\top\mathbf{W}\mathbf{A}} \leq \sum_{i\in[n]} w_i (\mathbf{A}(x - x_\star))_i^2$$
$$\leq 9\mathrm{OPT}^2 \sum_{i\in[n]} w_i \leq 18 d \mathrm{OPT}^2$$

as $\|w\|_1 \leq 2d$ by the construction in Lemma 2.5. Pluggin in this value of R, M into [18, Corollary 12] gives an iteration bound of

$$\widetilde{O}((RM)^{2/3}) = \widetilde{O}\left(\left(\mathrm{OPT}\sqrt{d} \cdot \frac{20\log n}{\varepsilon \mathrm{OPT}}\right)^{2/3}\right) = \widetilde{O}(d^{1/3}\varepsilon^{-2/3}).$$

□