Fast Algorithms for ℓ_p-Regression

DEEKSHA ADIL, University of Toronto, Canada
RASMUS KYNG, ETH Zurich, Switzerland
RICHARD PENG, University of Waterloo, Canada, (Part of the work done while at GaTech)
SUSHANT SACHDEVA, University of Toronto, Canada

The ℓ_p-norm regression problem is a classic problem in optimization with wide ranging applications in machine learning and theoretical computer science. The goal is to compute $\boldsymbol{x}^\star = \arg\min_{\boldsymbol{Ax}=\boldsymbol{b}} \|\boldsymbol{x}\|_p^p$, where $\boldsymbol{x}^\star \in \mathbb{R}^n, \boldsymbol{A} \in \mathbb{R}^{d\times n}, \boldsymbol{b} \in \mathbb{R}^d$ and $d \leq n$. Efficient *high-accuracy* algorithms for the problem have been challenging both in theory and practice and the state of the art algorithms require $poly(p) \cdot n^{\frac{1}{2}-\frac{1}{p}}$ linear system solves for $p \geq 2$. In this paper, we provide new algorithms for ℓ_p-regression (and a more general formulation of the problem) that obtain a *high-accuracy* solution in $O(pn^{(p-2)/(3p-2)})$ linear system solves. We further propose a new *inverse maintenance* procedure that speeds-up our algorithm to $\widetilde{O}(n^\omega)$ total runtime, where $O(n^\omega)$ denotes the running time for multiplying $n \times n$ matrices. Additionally, we give the first *Iteratively Reweighted Least Squares (IRLS)* algorithm that is guaranteed to converge to an optimum in a few iterations. Our IRLS algorithm has shown exceptional practical performance, beating the currently available implementations in MATLAB/CVX by 10-50x.

CCS Concepts: • **Theory of computation → Continuous optimization**.

Additional Key Words and Phrases: ℓ_p-Regression, Iterative Refinement, IRLS

ACM Reference Format:
Deeksha Adil, Rasmus Kyng, Richard Peng, and Sushant Sachdeva. 2022. Fast Algorithms for ℓ_p-Regression. *J. ACM* 37, 4, Article 111 (October 2022), 46 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Linear regression in ℓ_p-norm seeks to compute a vector $\boldsymbol{x}^\star \in \mathbb{R}^n$ such that,

$$\boldsymbol{x}^\star = \arg\min_{\boldsymbol{Ax}=\boldsymbol{b}} \|\boldsymbol{x}\|_p^p,$$

where $\boldsymbol{A} \in \mathbb{R}^{d\times n}, \boldsymbol{b} \in \mathbb{R}^d$, $d \leq n$. This is a classic convex optimization problem that captures several well-studied questions including least squares regression ($p = 2$) which is equivalent to solving a system of linear equations, and linear programming ($p = \infty$). The ℓ_p-norm regression problem for $p > 1$ has found use across a wide range of applications in machine learning and theoretical computer science including low rank matrix approximation [18], sparse recovery [14], graph based

Preliminary versions of the results in this paper have appeared as conference publications [1, 3, 4, 6]. This paper unifies and simplifies results from the preliminary versions.

Authors' addresses: Deeksha Adil, deeksha@cs.toronto.edu, University of Toronto, Canada; Rasmus Kyng, ETH Zurich, Switzerland, kyng@inf.ethz.ch; Richard Peng, University of Waterloo, Canada, (Part of the work done while at GaTech), y5peng@uwaterloo.ca; Sushant Sachdeva, sachdeva@cs.toronto.edu, University of Toronto, Canada.

0004-5411/2022/10-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

semi-supervised learning [7, 13, 32, 41], data clustering and learning problems [21, 22, 27]. In this paper, we focus on solving the ℓ_p-norm regression problem for $p \geq 2$. The exact solution to the ℓ_p-norm regression problem for $p \neq 1, 2, \infty$, may not even be expressible using rationals. Thus, the goal is often relaxed to finding an ε-approximate solution to the problem, i.e., find $\hat{\boldsymbol{x}}$ such that $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{b}$ and,

$$\|\hat{\boldsymbol{x}}\|_p^p \leq (1+\varepsilon)\|\boldsymbol{x}^\star\|_p^p,$$

for some small $\varepsilon > 0$. Furthermore, several applications such as graph based semi-supervised learning require that $\hat{\boldsymbol{x}}$ is close to $\boldsymbol{x}^\star$ coordinate-wise and not just in objective value – necessitating a *high-accuracy* solution with $\varepsilon \approx \frac{1}{\mathrm{poly}(n)}$. In order to find such high-accuracy solutions efficiently, we require an algorithm with runtime dependence on ε being $\mathrm{poly}(\log \frac{1}{\varepsilon})$ rather than $\mathrm{poly}(\frac{1}{\varepsilon})$.

Fast, high-accuracy algorithms for ℓ_p-regression are challenging both in theory and practice, due to the lack of smoothness and strong convexity of the objective. The Interior Point Method framework by Nesterov and Nemirovskii [37] can be used to compute a high-accuracy solution for all $p \in [1, \infty]$ in $\widetilde{O}(\sqrt{n})$[1] iterations, with each iteration requiring solving an $n \times n$ system of linear equations. This was the most efficient algorithm for ℓ_p-regression until 2018. In 2018, Bubeck et al. [10] showed that $\Omega(\sqrt{n})$ iterations are necessary for the interior point framework and proposed a new homotopy-based approach that could compute a high-accuracy solution in $\widetilde{O}(n^{\left|\frac{1}{2}-\frac{1}{p}\right|})$ linear system solves for all $p \in (1, \infty)$. Their algorithms improve over the interior point method by $n^{\Omega(1)}$ factors for values of p bounded away from 1 and ∞. However, for p approaching 1 or ∞, the number of linear system solves required by their algorithm approaches $\sqrt{n}$, the same as required by interior point methods. Finding an algorithm for ℓ_p-regression requiring $o(n^{1/2})$ linear system solves has been a long standing open problem.

Among practical implementations for the ℓ_p-norm regression problem, the *Iteratively Reweighted Least Squares (IRLS)* methods stand out due to their simplicity, and have been studied since 1961 [33]. For some range of values for p, IRLS converges rapidly. However, the method is guaranteed to converge only for $p \in (1.5, 3)$ and diverges even for small values of p, e.g. $p = 3.5$ [41]. Over the years, several empirical modifications of the algorithm have been used for various applications in practice (refer to [12] for a full survey). However, an IRLS algorithm that is guaranteed to converge to the optimum in a few iterations for all values of p, has again been a long standing challenge.

1.1 Our Contributions

In this paper, we present the first algorithm for the ℓ_p-regression problem that finds a high-accuracy solution in at most $O(pn^{1/3}) = o(n^{1/2})$ linear system solves, which has been a long sought-after goal in optimization. Our algorithm builds on a new *iterative refinement* framework for ℓ_p-norm objectives that allows us to find a high-accuracy solution using low-accuracy solutions to a subproblem. The iterative refinement framework allows for the subproblems to be solved to an $n^{o(1)}$-approximation and this has been useful in several follow up works on graph optimization (see Section 1.3). We further propose a new *inverse maintenance* framework and show how to speed up our algorithm to solve the ℓ_p-norm problem to a high-accuracy in total time $\widetilde{O}(n^\omega)$. Finally, we give the first IRLS algorithm that provably converges to a high-accuracy solution in a few iterations.

Preliminary versions of the results presented in this paper have appeared in previous conference publications by Adil et al. [1, 3, 4], Adil and Sachdeva [6]. In this paper, we present our results for a more general formulation of the ℓ_p-regression problem,

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \quad f(\boldsymbol{x}) = \boldsymbol{d}^\top \boldsymbol{x} + \|\boldsymbol{M}\boldsymbol{x}\|_2^2 + \|\boldsymbol{N}\boldsymbol{x}\|_p^p \tag{1}$$

[1] $\widetilde{O}$ hides constants, p dependencies, $\log \frac{1}{\varepsilon}$, and $\log n$ factors unless explicitly mentioned

for matrices $\boldsymbol{A} \in \mathbb{R}^{d \times n}, \boldsymbol{M} \in \mathbb{R}^{m_1 \times n}, \boldsymbol{N} \in \mathbb{R}^{m_2 \times n}$, $m_1, m_2 \geq n, d \leq n$. Let $m = \max\{m_1, m_2\}$ and, $\boldsymbol{d} \perp \{ker(\boldsymbol{M}) \cap ker(\boldsymbol{N}) \cap ker(\boldsymbol{A})\}$, $\boldsymbol{b} \in im(\boldsymbol{A})$ so that the above problem has a bounded solution. Our first result is a fast, high-accuracy algorithm for Problem (1).

Theorem 1.1. *Let $\varepsilon > 0$ and $p \geq 2$. There is an algorithm that starting from $\boldsymbol{x}^{(0)}$ satisfying $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$, finds an ε-approximate solution to Problem* (1) *in $O\left(pm^{\frac{p-2}{3p-2}} \log \frac{f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^{\star})}{\varepsilon}\right)$ calls to a linear system solver.*

As a corollary, for the ℓ_p-norm regression problem, i.e., $\boldsymbol{d} = \boldsymbol{M} = 0$ and $\boldsymbol{N} = \boldsymbol{I}$, our algorithm converges in $O(pn^{\frac{p-2}{3p-2}} \log \frac{n}{\varepsilon})$ calls to a linear system solver. This is the first algorithm that converges to a high accuracy solution at an asymptotic rate of convergence $\widetilde{O}(n^{1/3}) = o(n^{1/2})$ for all $p \in [2, \infty)$, and thus faster than all previously known algorithms by at least a factor of $n^{\Omega(1)}$. As a result, we answer the long standing problem in optimization of whether such a rate of convergence could be achieved.

Our next result shows how to speed up our algorithms and solve Problem (1) in time $\widetilde{O}(m^{\omega})$ (or $\widetilde{O}(n^{\omega})$ for ℓ_p-regression), where $\omega \approx 2.37$ and $O(n^{\omega})$ is the current time required for multiplying two $n \times n$ matrices. This is almost as fast as solving a system of linear equations. We achieve this guarantee via a new *inverse maintenance* procedure for ℓ_p-regression and prove the following result.

Theorem 1.2. *If $\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}$ are explicitly given, matrices with polynomially bounded condition numbers, and $p \geq 2$, there is an algorithm for Problem* (1) *that can be implemented to run in total time $\widetilde{O}(m^{\omega})$.*

Our inverse maintenance algorithm is presented in Section 5, where we also give a more fine grained dependence on the parameters m_1, m_2, n and p in the rate of convergence (Theorem 5.1). Our algorithms and techniques for ℓ_p-regression have motivated a line of work in graph optimization and the study of accelerated width reduced methods which we describe in detail in Section 1.3.

Our next contribution is towards the IRLS approach. For the ℓ_p-regression problem i.e. $\boldsymbol{d} = \boldsymbol{M} = 0$ in (1), we give an IRLS algorithm that globally converges to the optimum in at most $O\left(p^3 m^{\frac{p-2}{2(p-1)}} \log \frac{m}{\varepsilon}\right)$ linear system solves for all $p \geq 2$ (Section 6). This is the first IRLS algorithm that is guaranteed to converge to the optimum for all values of $p \geq 2$, with a quantitative bound on the runtime. Our IRLS algorithm has proven to be very fast and robust in practice and is faster than existing implementations in MATLAB/CVX by 10-50x. These speed-ups are demonstrated in experiments performed in [4] and we present these results along with our algorithm in Section 6.

Theorem 1.3. *Let $p \geq 2$. Algorithm 10 returns $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\|\boldsymbol{N}\boldsymbol{x}\|_p^p \leq (1+\varepsilon)\|\boldsymbol{N}\boldsymbol{x}^{\star}\|_p^p$, in at most $O\left(p^3 m^{\frac{(p-2)}{2(p-1)}} \log\left(\frac{m}{\varepsilon}\right)\right)$ calls to a linear system solver.*

The analysis of our IRLS algorithm fits into the overall framework of this paper. Such an algorithm first appeared in the conference paper by Adil et al. [4], where they also ran some experiments to demonstrate the performance of their IRLS algorithm in practice. We include some of their experimental results to show that the rate of convergence in practice is even better than the theoretical bounds.

1.2 Technical Overview

Overall $\log \frac{1}{\varepsilon}$ *Convergence.* Our algorithm follows an overall *iterative refinement* approach for $p \geq 2$, which implies $\boldsymbol{f}(\boldsymbol{x}+\delta) - \boldsymbol{f}(\boldsymbol{x})$ can be upper bounded by the function $\boldsymbol{res}_p = \boldsymbol{g}^{\top}\delta + \|\boldsymbol{R}\delta\|_2^2 + \|\boldsymbol{N}\delta\|_p^p$, and lower bounded by a similar function. Here, the vector $\boldsymbol{g}$ and matrix $\boldsymbol{R}$ depend on $\boldsymbol{x}$, and the matrix $\boldsymbol{N}$ is as defined in Problem (1). We prove that if we can solve $\min_{\boldsymbol{A}\delta=0} \boldsymbol{res}_p(\delta)$ to a κ-approximation, $O(p\kappa \log {}^{(f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^{\star}))}/_{\varepsilon})$ such solves (iterations) suffice to obtain an ε-approximate solution to Problem (1) (Theorem 2.1). We call this problem the *Residual Problem* and this process *Iterative Refinement for ℓ_p-norms.*

Solving the Residual Problem. We next perform a binary search on the linear term of the residual problem and reduce it to solving $O(\log p)$ problems of the form, $\min_{\boldsymbol{A}\delta=\boldsymbol{c}} \|\boldsymbol{R}\delta\|_2^2 + \|\boldsymbol{N}\delta\|_p^p$ (Lemma 3.1). In order to solve these new problems, we use a multiplicative weight update routine that returns a constant approximate solution in $O(pm^{(p-2)/(3p-2)})$ calls to a linear system solver (Theorem 3.2). We can thus find a constant approximate solution to the residual problem in $O(pm^{(p-2)/(3p-2)} \log p)$ calls to a linear system solver (Corollary 3.7). Combined with iterative refinement, we obtain an algorithm that converges in $O\left(p^2 m^{\frac{p-2}{3p-2}} \log p \log \frac{f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^{\star})}{\varepsilon}\right) \le \widetilde{O}(p^2 m^{1/3} \log \frac{1}{\varepsilon})$ linear system solves.

Improving p Dependence. Furthermore, we prove that for any $q \neq p$, given a p-norm residual problem, we can construct a corresponding q-norm residual problem such that β-approximate solution to the q-norm residual problem roughly gives a $O(\beta^2) m^{\left|\frac{1}{p}-\frac{1}{q}\right|}$ approximate solution to the p-norm residual problem (Theorem 4.3). As a consequence, if p is large, i.e. $p \ge \log m$, a constant approximate solution to the corresponding $\log m$-norm residual problem will give an $O(m^{\frac{1}{\log m}}) \le O(1)$-approximate solution to the p-norm residual problem in at most $O(\log m \cdot m^{\frac{\log m-2}{3\log m-2}}) \le \widetilde{O}(m^{\frac{p-2}{3p-2}})$ calls to a linear system solver. Combining this with the algorithm described in the previous paragraph, we obtain our final guarantees as described in Theorem 1.1.

ℓ_p-Regression in Matrix Multiplication Time. We next describe how to obtain the guarantees of Theorem 1.2. While solving the residual problem, the algorithm solves a system of linear equations at every iteration. The key observation for obtaining improved running times is that the weights determining these linear systems change slowly. Thus, we can maintain a spectral approximation to the linear system via a sequence of *lazy* low-rank updates. The Sherman-Morrison-Woodbury formula then allows us to update the inverse quickly. We can use the spectral approximation as a preconditioner for solving the linear system quickly at each iteration. Thus, we obtain a speed-up since the linear systems do not need to be solved from scratch at each iteration, giving Theorem 1.2.

Good Starting Solution. For ℓ_p-norm objectives, i.e., $\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \|\boldsymbol{N}\boldsymbol{x}\|_p^p$, we further show how to find a starting solution $\boldsymbol{x}^{(0)}$ such that $\|\boldsymbol{N}\boldsymbol{x}^{(0)}\|_p^p \le O(m)\|\boldsymbol{N}\boldsymbol{x}^{\star}\|_p^p$. The key idea is that for any k, a constant approximate solution to the k-norm problem is an $O(m)$-approximate solution to the $2k$-norm problem (Lemma 2.9). This inspires a homotopy approach, where we first solve an ℓ_2 norm problem followed by $\ell_{2^2}, \ell_{2^3}, \cdots, \ell_{2^{\lceil \log p \rceil}}$-norm problems to constant approximations. We can thus obtain the required starting solution in at most $O\left(pm^{\frac{p-2}{3p-2}} \log m \log^2 p\right)$ calls to a linear system solver.

IRLS Algorithm. For the IRLS algorithm, given the residual problem at an iteration, we show how to construct a weighted least squares problem, the solution of which is an $O\left(p^2 m^{\frac{p-2}{2(p-1)}}\right)$-approximate solution to the residual problem (Lemma 6.1). This result along with the overall iterative refinement culminates in our IRLS algorithm where we directly solve these weighted least squares problems in every iteration.

1.3 Related Works

ℓ_p-Regression. Until 2018, the fastest high-accuracy algorithms for ℓ_p-regression, including the Nesterov and Nemirovskii [37] Interior Point Method framework and Bubeck et al. [10] homotopy method, asymptotically required $\approx O(\sqrt{n})$ linear system solves. The first algorithm for ℓ_p-regression to beat the $\sqrt{n}$ iteration bound was the algorithm by Adil et al. [3], which was faster than all known algorithms and asymptotically required at most $\approx O(p^{O(p)}n^{1/3})$ iterations , for all $p > 1$. Concurrently Bullins [11] used tools from convex optimization to give an algorithm for $p = 4$ which matches the rates of Adil et al. [3] up to logarithmic factors. Subsequent works have improved the p dependence [1, 6] and proposed alternate methods for obtaining matching rates (upto logarithmic and p factors) [15]. A recent work by Jambulapati et al. [28] shows how to solve ℓ_p-regression in $\approx n + poly(p) \cdot d^{\frac{p-2}{3p-2}}$ iterations where d is the smaller dimension of the constraint matrix $\boldsymbol{A}$.

Width Reduced MWU Algorithms. Width reduction is a technique that has been used repeatedly in multiplicative weight update algorithms to speed up rates of convergence from $m^{1/2}$ to $m^{1/3}$, where m is the size of the input. This technique was first seen in the work of Christiano et al. [20], in the context of the maximum flow problem where for a graph with n vertices and m edges to improve the iteration complexity from $\widetilde{O}(m^{1/2})$ to $\widetilde{O}(m^{1/3})$. A similar improvement was further seen in algorithms for ℓ_1, ℓ_∞-regression by Chin et al. [19], Ene and Vladu [23], ℓ_p-regression ($p \geq 2$) Adil et al. [3] and, algorithms for matrix scaling [8]. In a recent work Adil et al. [2] extend this technique to improve iteration complexities for all *quasi-self-concordant* objectives which includes soft-max and logistic regression among others.

Inverse Maintenance. Inverse Maintenance is a technique used to speed up algorithms and was first introduced by Vaidya [45] in the context of minimum cost and multicommodity flows and has further been used for interior point methods Lee and Sidford [34], Lee et al. [36]. In 2019, Adil et al. [3] developed a method for ℓ_p-regression that utilized the idea of reusing inverses due to controllable rates of change of underlying variables.

IRLS Algorithms. Iteratively Reweighted Least Squares Algorithms are simple to implement and have thus been used in a wide range of applications including sparse signal reconstruction [24], compressive sensing [16] and Chebyshev approximation in FIR filter design [9]. Refer to Burrus [12] for a full survey. The works by Osborne [38] and Karlovitz [29] show convergence in the limit and with certain assumptions on the starting solution. For ℓ_1-regression, Straszak and Vishnoi [42, 43, 44] show quantitative convergence bounds. In 2019, Adil et al. [4] give the first IRLS algorithm with quantitative bounds that is guaranteed to converge with no conditions on the starting point. Their algorithm also works well in practice as suggested by their experiments.

Follow-up Work in Graph Optimization. The ℓ_p-norm flow problem, which asks to minimize the ℓ_p-norm of a flow vector while satisfying certain demand constraints, is modeled via the ℓ_p-regression problem. The maximum flow problem is the special case of $p = \infty$. For graphs with n vertices and m edges, the ℓ_p-norm regression algorithm of Adil et al. [3] when combined with fast laplacian solvers, directly gives an $\approx \widetilde{O}(p^{O(p)}m^{4/3})$ time algorithm for the ℓ_p-norm flow problem. Building on their work, specifically the iterative refinement framework, which allows to solve these problems to a high-accuracy while only requiring an $m^{o(1)}$-aproximate solution to an ℓ_p-norm subproblem, Kyng et al. [31] give an algorithm for unweighted graphs that runs in time $\exp(p^{3/2})m^{1+\frac{7}{\sqrt{p-1}}+o(1)}$. We note that their algorithm runs in time $m^{1+o(1)}$ for $p = \sqrt{\log m}$. Further works including Adil et al. [1] also utilize the iterative refinement guarantees to give an algorithm with runtime $p(m^{1+o(1)} + n^{4/3+o(1)})$

for weighted ℓ_p-norm flow problems by designing new sparsification algorithms that preserve ℓ_p-norm objectives of the subproblem to an $m^{o(1)}$-approximation. For the maximum flow problem, Adil and Sachdeva [6] give an $m^{1+o(1)}\varepsilon^{-1}$ time algorithm for the approximate maximum flow problem on unweighted graphs. Kathuria et al. [30] build on these works further and give an algorithm that computes maximum s-t flow problem where each edge has integer capacities at most U, in time $m^{4/3+o(1)}U^{1/3}$. In a recent breakthrough result by Chen et al. [17], the authors give an algorithm for the maximum flow problem and the ℓ_p-norm flow problem that runs in almost linear time, $m^{1+o(1)}$.

1.4 Organization of Paper

Section 2 describes the overall iterative refinement framework, first for $p \geq 2$, and then for $p \in (1,2)$. In the end, we show how to find good starting solutions for pure ℓ_p-norm objectives for $p \geq 2$. Section 3 describes the width reduced multiplicative weight update routine used to solve the residual problem. In Section 4 we show how to solve p-norm residual problems using q-norm residual problems and give our overall algorithm (Algorithm 6). Section 5 contains our new inverse maintenance algorithm that allows us to solve ℓ_p-regression almost as fast as linear regression. Finally in Section 6 we give an IRLS algorithm and present some experimental results from Adil et al. [4].

2 ITERATIVE REFINEMENT FOR ℓ_p-NORMS

Recall that we would like to find a high-accuracy solution for the problem,

$$\min_{Ax=b} \quad f(x) = d^\top x + \|Mx\|_2^2 + \|Nx\|_p^p$$

for matrices $A \in \mathbb{R}^{d\times n}, M \in \mathbb{R}^{m_1\times n}, N \in \mathbb{R}^{m_2\times n}$, $m_1, m_2 \geq n, d \leq n$.

A common approach in smooth, convex optimization is upper bounding the function using a first order Taylor expansion plus a quadratic function (smoothness), and minimizing this bound repeatedly to converge to the optimum. Additionally, when the function has a similar quadratic lower bound (strong convexity) it can be shown that minimizing this upper bound $O(\log \frac{1}{\varepsilon})^2$ times is sufficient to converge to an ε-approximate solution. The ℓ_p-norm function satisfies no such quadratic upper bound since it has a very steep growth, or lower bound since it is too flat around 0. In this section we show that we can instead upper and lower bound the ℓ_p function for $p \geq 2$ by a second order Taylor expansion plus an ℓ_p^p term. We show that it is sufficient to minimize such a bound to a κ-approximation $O(p\kappa \log \frac{1}{\varepsilon})$ times. Such an iterative refinement method was previously only known for $p = 2$, and we thus call this algorithm *Iterative Refinement for ℓ_p-norms*. In further sections, we show different ways to minimize this upper bound approximately to obtain fast algorithms.

For $p \in (1,2)$, we use a smoothed function which is quadratic in a small range around 0 and grows as ℓ_p^p otherwise. We use this function to give upper and lower bounds and a similar iterative refinement scheme.

We further show how to obtain a good starting solution for Problem (1) in the special case when the vector d and matrix M are zero, i.e., the objective function is only the ℓ_p-norm function.

These sections are based on the results and proofs from Adil et al. [1, 3, 4], Adil and Sachdeva [6].

2.1 Iterative Refinement

We will prove that the following algorithm can be used to obtain a high-accuracy solution, i.e., $\log \frac{1}{\varepsilon}$ rate of convergence for ℓ_p-regression.

[2] hiding problem dependent parameters

Algorithm 1 Iterative Refinement

```
procedure MAIN-SOLVER($\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{d}, \boldsymbol{b}, p, \varepsilon$)
    $\boldsymbol{x} \leftarrow \boldsymbol{x}^{(0)}$
    $\nu \leftarrow$ Bound on $f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^\star)$        ▷ If $f(\boldsymbol{x}^\star) \geq 0$, then $\nu \leftarrow f(\boldsymbol{x}^{(0)})$
    while $\nu > \varepsilon$ do
        $\widetilde{\Delta} \leftarrow$ RESIDUALSOLVER($\boldsymbol{x}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{A}, \boldsymbol{d}, \boldsymbol{b}, \nu, p$)
        if $\boldsymbol{res}_p(\widetilde{\Delta}) \geq \frac{\nu}{32p\kappa}$ then
            $\boldsymbol{x} \leftarrow \boldsymbol{x} - \frac{\widetilde{\Delta}}{p}$
        else
            $\nu \leftarrow \frac{\nu}{2}$
    return $\boldsymbol{x}$
```

Specifically, we will prove,

Theorem 2.1. *Let $p \geq 2$, and $\kappa \geq 1$. Let the initial solution $\boldsymbol{x}^{(0)}$ satisfy $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$. Algorithm 1 returns an ε-approximate solution $\boldsymbol{x}$ of Problem* (1) *in at most $O\left(p\kappa \log\left(\frac{f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^\star)}{\varepsilon}\right)\right)$ calls to a κ-approximate solver for the residual problem (Definition 2.3).*

Before we prove the above result, we will define some of the terms used in the above statement.

2.1.1 Preliminaries.

Definition 2.2 (ε-Approximate Solution). *Let $\boldsymbol{x}^\star$ denote the optimizer of Problem* (1)*. We say $\widetilde{\boldsymbol{x}}$ is an ε-approximate solution to* (1) *if $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \boldsymbol{b}$ and*

$$f(\widetilde{\boldsymbol{x}}) \leq f(\boldsymbol{x}^\star) + \varepsilon.$$

Definition 2.3 (Residual Problem). *For any $p \geq 2$, we define the residual problem $res_p(\Delta)$, for* (1) *at a feasible $\boldsymbol{x}$ as,*

$$\max_{\boldsymbol{A}\Delta = 0} \quad \boldsymbol{res}_p(\Delta) \stackrel{\text{def}}{=} \boldsymbol{g}^\top \Delta - \Delta^\top \boldsymbol{R} \Delta - \|\boldsymbol{N}\Delta\|_p^p, \text{ where,}$$

$$\boldsymbol{g} = \frac{1}{p}\boldsymbol{d} + \frac{2}{p}\boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} + \boldsymbol{N}^\top Diag(|\boldsymbol{N}\boldsymbol{x}|^{p-2})\boldsymbol{N}\boldsymbol{x} \quad \text{and} \quad \boldsymbol{R} = \frac{2}{p^2}\boldsymbol{M}^\top \boldsymbol{M} + 2\boldsymbol{N}^\top Diag(|\boldsymbol{N}\boldsymbol{x}|^{p-2})\boldsymbol{N}.$$

Definition 2.4 (Approximation to Residual Problem). *Let $p \geq 2$ and $\Delta^\star$ be the optimum of the residual problem. $\widetilde{\Delta}$ is a κ-approximation to the residual problem if $\boldsymbol{A}\widetilde{\Delta} = 0$ and,*

$$res_p(\widetilde{\Delta}) \geq \frac{1}{\kappa} res_p(\Delta^\star).$$

2.1.2 Bounding Change in Objective. In order to prove our result, we first show that we can upper and lower bound the change in our ℓ_p-objective by a linear term plus a quadratic term plus an ℓ_p-norm term.

Lemma 2.5. *For any $\boldsymbol{x}, \Delta$ and $p \geq 2$, we have for vectors $\boldsymbol{r}, \boldsymbol{g}$ defined coordinate wise as $\boldsymbol{r} = |\boldsymbol{x}|^{p-2}$ and $\boldsymbol{g} = p|\boldsymbol{x}|^{p-2}\boldsymbol{x}$,*

$$\frac{p}{8}\sum_i \boldsymbol{r}_i \Delta_i^2 + \frac{1}{2^{p+1}}\|\Delta\|_p^p \leq \|\boldsymbol{x} + \Delta\|_p^p - \|\boldsymbol{x}\|_p^p - \boldsymbol{g}^\top \Delta \leq 2p^2 \sum_i \boldsymbol{r}_i \Delta_i^2 + p^p \|\Delta\|_p^p.$$

Proof. To show this, we show that the above holds for all coordinates. For a single coordinate, the above expression is equivalent to proving,

$$\frac{p}{8}|x|^{p-2}\Delta^2 + \frac{1}{2^{p+1}}|\Delta|^p \leq |x+\Delta|^p - |x|^p - p|x|^{p-1}sgn(x)\Delta \leq 2p^2|x|^{p-2}\Delta^2 + p^p|\Delta|^p.$$

Let $\Delta = \alpha x$. Since the above clearly holds for $x = 0$, it remains to show for all α,

$$\frac{p}{8}\alpha^2 + \frac{1}{2^{p+1}}|\alpha|^p \leq |1+\alpha|^p - 1 - p\alpha \leq 2p^2\alpha^2 + p^p|\alpha|^p.$$

(1) $\alpha \geq 1$:
In this case, $1 + \alpha \leq 2\alpha \leq p \cdot \alpha$. So, $|1+\alpha|^p \leq p^p|\alpha|^p$ and the right inequality directly holds. To show the other side, let

$$h(\alpha) = (1+\alpha)^p - 1 - p\alpha - \frac{p}{8}\alpha^2 - \frac{1}{2^{p+1}}\alpha^p.$$

We have,

$$h'(\alpha) = p(1+\alpha)^{p-1} - p - \frac{p}{4}\alpha - \frac{p}{2^{p+1}}\alpha^{p-1}$$

and

$$h''(\alpha) = p(p-1)(1+\alpha)^{p-2} - \frac{p}{4} - \frac{p(p-1)}{2^{p+1}}\alpha^{p-2} \geq 0.$$

Since $h''(\alpha) \geq 0$, $h'(\alpha) \geq h'(1) \geq 0$. So h is an increasing function in α and $h(\alpha) \geq h(1) \geq 0$.

(2) $\alpha \leq -1$:
Now, $|1+\alpha| \leq 1 + |\alpha| \leq p \cdot |\alpha|$, and $2\alpha^2p^2 - |\alpha|p \geq 0$. As a result,

$$|1+\alpha|^p \leq -|\alpha|p + 2\alpha^2p^2 + p^p \cdot |\alpha|^p$$

which gives the right inequality. Consider,

$$h(\alpha) = |1+\alpha|^p - 1 - p\alpha - \frac{p}{8}\alpha^2 - \frac{1}{2^{p+1}}|\alpha|^p.$$

$$h'(\alpha) = -p|1+\alpha|^{p-1} - p - \frac{p}{4}\alpha + p\frac{1}{2^{p+1}}|\alpha|^{p-1}.$$

Let $\beta = -\alpha$. The above expression now becomes,

$$-p(\beta-1)^{p-1} - p + \frac{p}{4}\beta + p\frac{1}{2^{p+1}}\beta^{p-1}.$$

We know that $\beta \geq 1$. When $\beta \geq 2$, $\frac{\beta}{2} \leq \beta - 1$ and $\frac{\beta}{2} \leq \left(\frac{\beta}{2}\right)^{p-1}$. This gives us,

$$\frac{p}{4}\beta + p\frac{1}{2^{p+1}}\beta^{p-1} \leq \frac{p}{2}\left(\frac{\beta}{2}\right)^{p-1} + \frac{p}{2}\left(\frac{\beta}{2}\right)^{p-1} \leq p(\beta-1)^{p-1}$$

giving us $h'(\alpha) \leq 0$ for $\alpha \leq -2$. When $\beta \leq 2$, $\frac{\beta}{2} \geq \left(\frac{\beta}{2}\right)^{p-1}$ and $\frac{\beta}{2} \leq 1$.

$$\frac{p}{4}\beta + p\frac{1}{2^{p+1}}\beta^{p-1} \leq \frac{p}{2}\cdot\frac{\beta}{2} + \frac{p}{2}\cdot\frac{\beta}{2} \leq p$$

giving us $h'(\alpha) \leq 0$ for $-2 \leq \alpha \leq -1$. Therefore, $h'(\alpha) \leq 0$ giving us, $h(\alpha) \geq h(-1) \geq 0$, thus giving the left inequality.

(3) $|\alpha| \leq 1$:
Let $s(\alpha) = 1 + p\alpha + 2p^2\alpha^2 + p^p|\alpha|^p - (1+\alpha)^p$. Now,

$$s'(\alpha) = p + 4p^2\alpha + p^{p+1}|\alpha|^{p-1}sgn(\alpha) - p(1+\alpha)^{p-1}.$$

When $\alpha \leq 0$, we have,

$$s'(\alpha) = p + 4p^2\alpha - p^{p+1}|\alpha|^{p-1} - p(1+\alpha)^{p-1}.$$

and

$$s''(\alpha) = 4p^2 + p^{p+1}(p-1)|\alpha|^{p-2} - p(p-1)(1+\alpha)^{p-1} \geq 2p^2 + p^{p+1}(p-1)|\alpha|^{p-2} - p(p-1) \geq 0.$$

So s' is an increasing function of α which gives us, $s'(\alpha) \leq s'(0) = 0$. Therefore s is a decreasing function, and the minimum is at 0 which is 0. This gives us our required inequality for $\alpha \leq 0$. When $\alpha \geq \frac{1}{p-1}$, $1+\alpha \leq p\cdot\alpha$ and $s'(\alpha) \geq 0$. We are left with the range $0 \leq \alpha \leq \frac{1}{p-1}$. Again, we have,

$$\begin{aligned} s''(\alpha) &= 4p^2 + p^{p+1}(p-1)|\alpha|^{p-2} - p(p-1)(1+\alpha)^{p-1} \\ &\geq 4p^2 + p^{p+1}(p-1)|\alpha|^{p-2} - p(p-1)(1+\frac{1}{p-1})^{p-1} \\ &\geq 4p^2 + p^{p+1}(p-1)|\alpha|^{p-2} - p(p-1)e, \text{When } p \text{ gets large the last term approaches } e \\ &\geq 0. \end{aligned}$$

Therefore, s' is an increasing function, $s'(\alpha) \geq s'(0) = 0$. This implies s is an increasing function, giving, $s(\alpha) \geq s(0) = 0$ as required.
To show the other direction,

$$h(\alpha) = (1+\alpha)^p - 1 - p\alpha - \frac{p}{8}\alpha^2 - \frac{1}{2^{p+1}}|\alpha|^p \geq (1+\alpha)^p - 1 - p\alpha - \frac{p}{8}\alpha^2 - \frac{p}{8}\alpha^2 = (1+\alpha)^p - 1 - p\alpha - \frac{p}{4}\alpha^2.$$

Now, since $p \geq 2$,

$$\begin{aligned} &\big((1+\alpha)^{p-2} - 1\big)sgn(\alpha) \geq 0 \\ \Rightarrow &\big((1+\alpha)^{p-1} - 1 - \alpha\big)sgn(\alpha) \geq 0 \\ \Rightarrow &\left(p(1+\alpha)^{p-1} - p - \frac{p}{2}\alpha\right)sgn(\alpha) \geq 0 \end{aligned}$$

We thus have, $h'(\alpha) \geq 0$ when α is positive and $h'(\alpha) \leq 0$ when α is negative. The minimum of h is at 0 which is 0. This concludes the proof of this case.

□

2.1.3 *Proof of Iterative Refinement.* In this section we will prove our main result. We start by proving the following lemma which relates the objective of the residual problem defined in the preliminaries to the change in objective value when $\boldsymbol{x}$ is updated by Δ/p.

Lemma 2.6. *For any $\boldsymbol{x}, \Delta$ and $p \geq 2$ and $\lambda = 16p$,*

$$\boldsymbol{res}_p(\Delta) \leq \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}\left(\boldsymbol{x} - \frac{\Delta}{p}\right),$$

and

$$\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}\left(\boldsymbol{x} - \lambda\frac{\Delta}{p}\right) \leq \lambda \cdot \boldsymbol{res}_p(\Delta).$$

Proof. We note,

$$\begin{aligned}
\boldsymbol{f}\left(\boldsymbol{x}-\frac{\Delta}{p}\right) =& \boldsymbol{d}^\top\left(\boldsymbol{x}-\frac{\Delta}{p}\right)+\left\|\boldsymbol{M}\left(\boldsymbol{x}-\frac{\Delta}{p}\right)\right\|_2^2+\left\|\boldsymbol{N}\left(\boldsymbol{x}-\frac{\Delta}{p}\right)\right\|_p^p \\
=& \boldsymbol{d}^\top\boldsymbol{x}+\|\boldsymbol{Mx}\|_2^2+\left\|\boldsymbol{N}\left(\boldsymbol{x}-\frac{\Delta}{p}\right)\right\|_p^p-\frac{1}{p}\boldsymbol{d}^\top\Delta-\frac{2}{p}\boldsymbol{x}^\top\boldsymbol{M}^\top\boldsymbol{M}\Delta+\frac{1}{p^2}\|\boldsymbol{M}\Delta\|_2^2 \\
\leq& \boldsymbol{d}^\top\boldsymbol{x}+\|\boldsymbol{Mx}\|_2^2+\|\boldsymbol{Nx}\|_p^p-p|\boldsymbol{Nx}|^{p-2}(\boldsymbol{Nx})^\top\frac{\boldsymbol{N}\Delta}{p}+2p^2\frac{(\boldsymbol{N}\Delta)^\top}{p}(\boldsymbol{Nx})^{p-2}\frac{\boldsymbol{N}\Delta}{p} \\
&+p^p\left\|\frac{\boldsymbol{N}\Delta}{p}\right\|_p^p-\frac{1}{p}\boldsymbol{d}^\top\Delta-\frac{2}{p}\boldsymbol{x}^\top\boldsymbol{M}^\top\boldsymbol{M}\Delta+\frac{1}{p^2}\|\boldsymbol{M}\Delta\|_2^2 \\
&\text{(From right inequality of Lemma 2.5)} \\
=& \boldsymbol{f}(\boldsymbol{x})-\left(\frac{1}{p}\boldsymbol{d}+\frac{2}{p}\boldsymbol{M}^\top\boldsymbol{Mx}+\boldsymbol{N}^\top|\boldsymbol{Nx}|^{p-2}\boldsymbol{Nx}\right)^\top\Delta \\
&-\Delta^\top\left(\frac{2}{p^2}\boldsymbol{M}^\top\boldsymbol{M}+2\boldsymbol{N}^\top Diag(|\boldsymbol{Nx}|^{p-2})\boldsymbol{N}\right)\Delta+\|\boldsymbol{N}\Delta\|_p^p \\
=& \boldsymbol{f}(\boldsymbol{x})-\boldsymbol{res}_p(\Delta),\ \text{From Definition 2.3.}
\end{aligned}$$

Let $\boldsymbol{g}$ and $\boldsymbol{R}$ be as defined in Definition 2.3. We now use a similar calculation and the left inequality of Lemma 2.5 to get,

$$\boldsymbol{f}\left(\boldsymbol{x}-\lambda\frac{\Delta}{p}\right)\geq\boldsymbol{f}(\boldsymbol{x})-\lambda\boldsymbol{g}^\top\Delta-\frac{\lambda^2}{16p}\Delta^\top\boldsymbol{R}\Delta-\frac{\lambda^p}{p^p2^{p+1}}.$$

For $\lambda=16p$,

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x})-\lambda\boldsymbol{g}^\top\Delta-\frac{\lambda^2}{16p}\Delta^\top\boldsymbol{R}\Delta-\frac{\lambda^p}{p^p2^{p+1}} &\geq \boldsymbol{f}(\boldsymbol{x})-\lambda\left(\boldsymbol{g}^\top\Delta-\frac{\lambda}{16p}\Delta^\top\boldsymbol{R}\Delta-\frac{\lambda^{p-1}}{p^p2^{p+1}}\right) \\
&\geq \boldsymbol{f}(\boldsymbol{x})-\lambda\boldsymbol{res}_p(\Delta),
\end{aligned}$$

thus concluding the proof of the lemma. □

We now track the value of $\boldsymbol{f}(\boldsymbol{x}^{(t)})-\boldsymbol{f}(\boldsymbol{x}^\star)$ with a parameter ν. We will first show that, if we have a κ approximate solver for the residual problem, we can either take a step to obtain $\boldsymbol{x}^{(t+1)}$ such that

$$\boldsymbol{f}(\boldsymbol{x}^{(t+1)})-\boldsymbol{f}(\boldsymbol{x}^\star)\leq\left(1-\frac{1}{32p\kappa}\right)\left(\boldsymbol{f}(\boldsymbol{x}^{(t)})-\boldsymbol{f}(\boldsymbol{x}^\star)\right), \tag{2}$$

or we need to reduce the value of ν by a factor of 2 since $\boldsymbol{f}(\boldsymbol{x}^{(t)})-\boldsymbol{f}(\boldsymbol{x}^\star)$ is less than $\nu/2$.

Lemma 2.7. *Consider an iterate t. Let $\boldsymbol{res}_p$ denote the residual problem at $\boldsymbol{x}^{(t)}$ and ν be as defined in Algorithm 1. Let $\widetilde{\Delta}$ denote the solution returned by a κ-approximate solver to the residual problem. Then,*

(1) *either $\boldsymbol{f}(\boldsymbol{x}^{(t)})-\boldsymbol{f}(\boldsymbol{x}^\star)\leq\nu$ and, $\boldsymbol{x}^{(t+1)}=\boldsymbol{x}^{(t)}-\frac{\widetilde{\Delta}}{p}$ satisfies (2),*
(2) *or, $\boldsymbol{f}(\boldsymbol{x}^{(t)})-\boldsymbol{f}(\boldsymbol{x}^\star)\leq\frac{\nu}{2}$ and Line 9 in the algorithm is executed.*

Proof. We will first prove that $\boldsymbol{f}(\boldsymbol{x}^{(t)})-\boldsymbol{f}(\boldsymbol{x}^\star)\leq\nu$ by induction. For $t=0,\boldsymbol{f}(\boldsymbol{x}^{(0)})-\boldsymbol{f}(\boldsymbol{x}^\star)\leq\nu$ by definition. Now, let us assume this is true for iteration t. Note that, if the algorithm updates $\boldsymbol{x}$ in line 7, since $\boldsymbol{f}(\boldsymbol{x}^{(t+1)})\leq\boldsymbol{f}(\boldsymbol{x}^{(t)})$ (solution of the residual problem is always non-negative), the relation holds for $t+1$. Otherwise, the algorithm reduces ν to $\nu/2$ and $\boldsymbol{res}_p(\widetilde{\Delta}) < \frac{\nu}{32p\kappa}$. For $\bar{\Delta}$ such that $\boldsymbol{x}^\star = \boldsymbol{x}^{(t)} - \lambda\frac{\bar{\Delta}}{p}$, and from Lemma 2.6,

$$\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) = \boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}\left(\boldsymbol{x}^{(t)} - \lambda\frac{\bar{\Delta}}{p}\right) \leq \lambda\boldsymbol{res}_p(\bar{\Delta}) \leq \lambda\boldsymbol{res}_p(\Delta^\star).$$

Since $\widetilde{\Delta}$ is a κ-approximate solution to the residual problem,

$$\lambda\boldsymbol{res}_p(\Delta^\star) \leq \lambda\kappa\boldsymbol{res}_p(\widetilde{\Delta}) < 16p\kappa\frac{\nu}{32p\kappa} \leq \frac{\nu}{2}.$$

We have thus shown that $\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) \leq \nu$ for all iterates t and whenever Line 9 of the algorithm is executed, 2 from the lemma statement holds. It remains to prove that if $\boldsymbol{res}_p(\widetilde{\Delta}) \geq \frac{\nu}{32p\kappa}$, then $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \frac{\widetilde{\Delta}}{p}$ satisfies (2). Since, $\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) \leq \nu$,

$$\boldsymbol{res}_p(\widetilde{\Delta}) \geq \frac{\nu}{32p\kappa} \geq \frac{1}{32p\kappa}\left(\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star)\right).$$

Now, from Lemma 2.6,

$$\begin{aligned}\boldsymbol{f}\left(\boldsymbol{x}^{(t+1)}\right) - \boldsymbol{f}(\boldsymbol{x}^\star) &\leq \boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{res}_p(\widetilde{\Delta}) - \boldsymbol{f}(\boldsymbol{x}^\star)\\ &\leq \left(\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star)\right) - \frac{1}{32p\kappa}\left(\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star)\right)\\ &= \left(1 - \frac{1}{32p\kappa}\right)\left(\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star)\right).\end{aligned}$$

□

Corollary 2.8. *The vector $\boldsymbol{x}$ returned by Algorithm 1 is an ε-approximate solution to Problem* (1).

Proof. Our starting solution $\boldsymbol{x}^{(0)}$ satisfies $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$ and the solutions $\widetilde{\Delta}$ of the residual problem added in each iteration satisfy $\boldsymbol{A}\widetilde{\Delta} = 0$. Therefore, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. For the second part, note that we always have $\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) \leq \nu$. When we stop, $\nu \leq \varepsilon$. Thus,

$$\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) \leq \varepsilon.$$

□

We are now ready to prove our main result.

Theorem 2.1. *Let $p \geq 2$, and $\kappa \geq 1$. Let the initial solution $\boldsymbol{x}^{(0)}$ satisfy $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$. Algorithm 1 returns an ε-approximate solution $\boldsymbol{x}$ of Problem* (1) *in at most $O\left(p\kappa\log\left(\frac{f(x^{(0)})-f(x^\star)}{\varepsilon}\right)\right)$ calls to a κ-approximate solver for the residual problem (Definition 2.3).*

Proof. From Corollary 2.8, the solution returned by the algorithm is as required. We next need to bound the runtime. From Lemma 2.7, the algorithm, either reduces ν or Equation (2) holds. The number of times we can reduce ν is bounded by $\log\frac{f(x^{(0)})-f(x^\star)}{\varepsilon}$. The number of times Equation (2) holds can be bounded as follows,

$$\frac{\varepsilon}{2} \leq \boldsymbol{f}\left(\boldsymbol{x}^{(t+1)}\right) - \boldsymbol{f}(\boldsymbol{x}^\star) \leq \left(1 - \frac{1}{32p\kappa}\right)^t\left(\boldsymbol{f}(\boldsymbol{x}^{(0)}) - \boldsymbol{f}(\boldsymbol{x}^\star)\right).$$

Therefore, the total number of iterations T is bounded as $T \leq 32p\kappa\log\left(\frac{f(x^{(0)})-f(x^\star)}{\varepsilon}\right)$. □

2.2 Starting Solution and Homotopy for pure ℓ_p Objectives

In this section, we consider the case where $\boldsymbol{f}(\boldsymbol{x}) = \|\boldsymbol{N}\boldsymbol{x}\|_p^p$, i.e., $\boldsymbol{d} = 0$ and $\boldsymbol{M} = 0$.

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \|\boldsymbol{N}\boldsymbol{x}\|_p^p \tag{3}$$

For such cases, we show how to find a good starting solution. We note that we can solve the following problem since it is equivalent to solving a system of linear equations,

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \|\boldsymbol{N}\boldsymbol{x}\|_2^2.$$

Refer to Appendix A for details on how the above is equivalent to solving a system of linear equations.

We next consider a homotopy on p. Specifically, we want to find a starting solution for the ℓ_p-norm problem by first solving an ℓ_2-norm problem, followed by $\ell_{2^2}, \ell_{2^3}, ..., \ell_{2^{\lfloor \log p-1 \rfloor}}$-norm problems to a constant approximation. The following lemma relates these solutions.

Lemma 2.9. *Let $\boldsymbol{x}_k^\star$ denote the optimum of the k-norm and $\boldsymbol{x}_{2k}^\star$ the optimum of the $2k$-norm problem (3). Let $\widetilde{\boldsymbol{x}}$ be an $O(1)$-approximate solution to the k-norm problem. The following relation holds,*

$$\left\|\boldsymbol{x}_{2k}^\star\right\|_{2k}^{2k} \leq \|\widetilde{\boldsymbol{x}}\|_{2k}^{2k} \leq O(m)\left\|\boldsymbol{x}_{2k}^\star\right\|_{2k}^{2k}.$$

In other words, $\widetilde{\boldsymbol{x}}$ is a $O(m)$-approximate solution to the $2k$-norm problem.

Proof. The left side follows from optimality of $\boldsymbol{x}_{2k}^\star$. For the other side, we have the following relation,

$$\|\widetilde{\boldsymbol{x}}\|_{2k}^{2k} \leq \|\widetilde{\boldsymbol{x}}\|_k^{2k} \leq O(1)\left\|\boldsymbol{x}_k^\star\right\|_k^{2k} \leq O(1)\left\|\boldsymbol{x}_{2k}^\star\right\|_k^{2k} \leq O(1)m^{2k\left(\frac{1}{k}-\frac{1}{2k}\right)}\left\|\boldsymbol{x}_{2k}^\star\right\|_{2k}^{2k} = O(m)\left\|\boldsymbol{x}_{2k}^\star\right\|_{2k}^{2k}.$$

□

Consider the following procedure to obtain a starting point $\boldsymbol{x}^{(0)}$ for the ℓ_p-norm problem.

Algorithm 2 Homotopy on p for Starting Solution

```
procedure StartSolution($\boldsymbol{A}, \boldsymbol{N}, \boldsymbol{b}, p$)
    $\boldsymbol{x}^{(0)} \leftarrow 0, k \leftarrow 2$
    while $k \leq 2^{\lfloor \log p-1 \rfloor}$ do
        $\boldsymbol{x}^{(0)} \leftarrow$ Main-Solver $(\boldsymbol{A}, 0, \boldsymbol{N}, 0, \boldsymbol{b}, k, 1)$        ▷ 2-approximate solution to the $k$-norm Problem
        $k \leftarrow 2k$
    return $\boldsymbol{x}^{(0)}$
```

Lemma 2.10. *Let $\boldsymbol{x}^{(0)}$ be as returned by Algorithm 2. Suppose there exists an oracle that solves the residual problem for any norm ℓ_k, i.e., $\boldsymbol{res}_k$ to a κ_k-approximation in time $T(k, \kappa_k)$. We can then compute $\boldsymbol{x}^{(0)}$ which is a $O(m)$-approximation to the ℓ_p-norm problem, in time at most*

$$O(p \log m) \sum_{k=2^i, i=2}^{i=\lfloor \log p-1 \rfloor} \kappa_k T(k, \kappa_k).$$

Proof. For any k, we have an $O(1)$-approximation solution to the $k/2$-norm solution. From Lemma 2.9, this is a $O(m)$-approximate solution to the k-norm problem. We now have from

Theorem 2.1, that we require $O(k\kappa_k T(k,\kappa_k)\log m)$ time to solve the k-norm problem to a constant approximation. Summing over all k, we have total runtime,

$$T = \sum_{k=2^i, i=2}^{i=\lfloor \log p - 1\rfloor} O(k\kappa_k T(k,\kappa_k)\log m) \leq O(p\log m) \sum_{k=2^i, i=2}^{i=\lfloor \log p - 1\rfloor} \kappa_k T(k,\kappa_k).$$

□

In later sections, we will describe an oracle that will have $\kappa_k = O(1)$ for all values of k and $T(k,\kappa_k)$ depends on k linearly.

2.3 Iterative Refinement for $p \in (1,2)$

We will consider the following pure ℓ_p problem here,

$$\min_{\boldsymbol{Ax}=\boldsymbol{b}} \|\boldsymbol{Nx}\|_p, \tag{4}$$

where $p \in (1,2)$. In the previous sections we saw an iterative refinement framework that worked for $p \geq 2$. In this section, we will show a similar iterative refinement for $p \in (1,2)$. In particular, we will prove the following result from [3].

Theorem 2.11. *Let $p \in (1,2)$, and $\kappa \geq 1$. Given an initial solution $\boldsymbol{x}^{(0)}$ satisfying $\boldsymbol{Ax}^{(0)} = \boldsymbol{b}$, we can find $\widetilde{\boldsymbol{x}}$ such that $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \boldsymbol{b}$ and $\|\boldsymbol{N}\widetilde{\boldsymbol{x}}\|_p^p \leq (1+\varepsilon)\|\boldsymbol{x}^\star\|_p^p$ in $O\left(\left(\frac{p}{p-1}\right)^{\frac{1}{p-1}} \kappa \log \frac{m}{\varepsilon}\right)$ calls to a κ-approximate solver to the residual problem (Definition 2.13).*

The key idea in the algorithm for $p \geq 2$ was an upper and lower bound on the function that was an $\ell_2^2 + \ell_p^p$-norm term (Lemma 2.5). Such a bound does not hold when $p < 2$, however, we will show that a smoothed ℓ_p-norm function can be used for providing such bounds. Specifically, we use the following smoothed ℓ_p-norm function defined in [10].

Definition 2.12. *(Smoothed ℓ_p Function.) Let $p \in (1,2)$, and $x \in \mathbb{R}$, $t \geq 0$. We define,*

$$\gamma_p(t,x) = \begin{cases} \frac{p}{2} t^{p-2} x^2 & \text{if } |x| \leq t, \\ |x|^p - \left(1 - \frac{p}{2}\right) t^p & \text{otherwise.} \end{cases}$$

For any vector $\boldsymbol{x}$ and $\boldsymbol{t} \geq 0$, we define $\gamma_p(\boldsymbol{t},\boldsymbol{x}) = \sum_i \gamma_p(\boldsymbol{t}_i, \boldsymbol{x}_i)$.

We define the following residual problem for this section.

Definition 2.13. *For $p \in (1,2)$, we define the residual problem at any feasible $\boldsymbol{x}$ to be,*

$$\max_{\boldsymbol{A}\Delta=0} \boldsymbol{res}_p(\Delta) \stackrel{\text{def}}{=} \boldsymbol{g}^\top \Delta - 2^p \gamma_p(|\boldsymbol{Nx}|, \boldsymbol{N}\Delta),$$

where $\boldsymbol{g} = p\boldsymbol{N}^\top |\boldsymbol{Nx}|^{p-2} \boldsymbol{Nx}$.

We will follow a similar structure as Section 2.1. We begin by proving analogues of Lemma 2.6 and Lemma 2.5.

Lemma 2.14. *Let $p \in (1,2)$. For any x and Δ,*

$$|x|^p + p|x|^{p-2}x\Delta + \frac{p-1}{p2^p}\gamma_p(|x|,\Delta) \leq |x+\Delta|^p \leq |x|^p + p|x|^{p-2}x\Delta + 2^p\gamma_p(|x|,\Delta)$$

Proof. We first show the following inequality holds for $|\alpha| \leq 1$

$$1 + \alpha p + \frac{(p-1)}{4}\alpha^2 \leq (1+\alpha)^p \leq 1 + \alpha p + p2^{p-1}\alpha^2. \tag{5}$$

Let us first show the left inequality, i.e. $1 + \alpha p + \frac{p-1}{4}\alpha^2 \leq (1+\alpha)^p$. Define the following function,

$$h(\alpha) = (1+\alpha)^p - 1 - \alpha p - \frac{p-1}{4}\alpha^2.$$

When $\alpha = 1, -1$, $h(\alpha) \geq 0$. The derivative of h with respect to α is, $h'(\alpha) = p(1+\alpha)^{p-1} - p - \frac{(p-1)}{2}\alpha$. Next let us see what happens when $|\alpha| < 1$.

$$h''(\alpha) = p(p-1)(1+\alpha)^{p-2} - \frac{p-1}{2} = (p-1)\left(\frac{p}{(1+\alpha)^{2-p}} - \frac{1}{2}\right) \geq 0$$

This implies that $h'(\alpha)$ is an increasing function of α and α_0 for which $h'(\alpha_0) = 0$ is where h attains its minimum value. The only point where h' is 0 is $\alpha_0 = 0$. This implies $h(\alpha) \geq h(0) = 0$. This concludes the proof of the left inequality. For the right inequality, define:

$$s(\alpha) = 1 + \alpha p + p2^{p-1}\alpha^2 - (1+\alpha)^p.$$

Note that $s(0) = 0$ and $s(1), s(-1) \geq 0$. We have,

$$s'(\alpha) = p + p2^p\alpha - p(1+\alpha)^{p-1},$$

and

$$(1+\alpha)^{p-1}sign(\alpha) \leq (1+\alpha)sign(\alpha).$$

Using this, we get, $s'(\alpha)sign(\alpha) \geq p|\alpha|(2^p - 1) \geq 0$ which says $s'(\alpha)$ is positive for α positive and negative for α negative. Thus the minima of s is at 0 which is 0. So $s(\alpha) \geq 0$.

Before we prove the lemma, we will prove the following inequality for $\beta \geq 1$,

$$(\beta - 1)^{p-1} + 1 \geq \frac{1}{2^p}\beta^{p-1}. \tag{6}$$

$(\beta - 1) \geq \frac{\beta}{2}$ for $\beta \geq 2$. So the claim clearly holds for $\beta \geq 2$ since $(\beta-1)^{p-1} \geq \left(\frac{\beta}{2}\right)^{p-1}$. When $1 \leq \beta \leq 2$, $1 \geq \frac{\beta}{2}$, so the claim holds since, $1 \geq \left(\frac{\beta}{2}\right)^{p-1}$
We now prove the lemma.

Let $\Delta = \alpha x$. The term $p|x|^{p-1}sign(x) \cdot \alpha x = \alpha p|x|^{p-1}|x| = \alpha p|x|^p$. Let us first look at the case when $|\alpha| \leq 1$. We want to show,

$$|x|^p + \alpha p|x|^p + c\frac{p}{2}|x|^{p-2}|\alpha x|^2 \leq |x + \alpha x|^p \leq |x|^p + \alpha p|x|^p + C\frac{p}{2}|x|^{p-2}|\alpha x|^2$$
$$\Leftrightarrow (1+\alpha p) + c\frac{p}{2}\alpha^2 \leq (1+\alpha)^p \leq (1+\alpha p) + C\frac{p}{2}\alpha^2.$$

This follows from Equation (5) and the facts $\frac{cp}{2} \leq \frac{p-1}{4}$ and $\frac{Cp}{2} \geq p2^{p-1}$. We next look at the case when $|\alpha| \geq 1$. Now, $\gamma_p(|f|, \Delta) = |\Delta|^p + (\frac{p}{2} - 1)|f|^p$. We need to show

$$|x|^p(1+\alpha p) + \frac{|x|^p(p-1)}{p2^p}(|\alpha|^p + \frac{p}{2} - 1) \leq |x|^p|1+\alpha|^p \leq |x|^p(1+\alpha p) + 2^p|x|^p(|\alpha|^p + \frac{p}{2} - 1).$$

When $|x| = 0$ it is trivially true. When $|x| \neq 0$, let

$$h(\alpha) = |1+\alpha|^p - (1+\alpha p) - \frac{(p-1)}{p2^p}(|\alpha|^p + \frac{p}{2} - 1).$$

Now, taking the derivative with respect to α we get,

$$h'(\alpha) = p\left(|1+\alpha|^{p-1} sign(\alpha) - 1 - \frac{(p-1)}{p2^p}|\alpha|^{p-1} sign(\alpha)\right).$$

We use the mean value theorem to get for $|\alpha| \geq 1$,

$$\begin{aligned}(1+\alpha)^{p-1} - 1 &= (p-1)\alpha(1+z)^{p-2}, z \in (0,\alpha)\\ &\geq (p-1)\alpha(2\alpha)^{p-2}\\ &\geq \frac{p-1}{2}\alpha^{p-1}\end{aligned}$$

which implies $h'(\alpha) \geq 0$ in this range as well. When $\alpha \leq -1$ it follows from Equation (6) that $h'(\alpha) \leq 0$. So the function h is increasing for $\alpha \geq 1$ and decreasing for $\alpha \leq -1$. The minimum value of h is $min\{h(1), h(-1)\} \geq 0$. It follows that $h(\alpha) \geq 0$ which gives us the left inequality. The other side requires proving,

$$|1+\alpha|^p \leq 1 + \alpha p + 2^p(|\alpha|^p + \frac{p}{2} - 1).$$

Define:

$$s(\alpha) = 1 + \alpha p + 2^p(|\alpha|^p + \frac{p}{2} - 1) - |1+\alpha|^p.$$

The derivative $s'(\alpha) = p + \left(p2^p|\alpha|^{p-1} - p|1+\alpha|^{p-1}\right) sign(\alpha)$ is non negative for $\alpha \geq 1$ and non positive for $\alpha \leq -1$. The minimum value taken by s is $\min\{s(1), s(-1)\}$ which is non negative. This gives us the right inequality.

□

Lemma 2.15. *Let $p \in (1,2)$ and $\lambda^{p-1} = \frac{p4^p}{p-1}$. Then for any Δ,*

$$\boldsymbol{res}_p(\Delta) \leq \|\boldsymbol{Nx}\|_p^p - \|\boldsymbol{N}(\boldsymbol{x}-\Delta)\|_p^p,$$

and

$$\|\boldsymbol{Nx}\|_p^p - \|\boldsymbol{N}(\boldsymbol{x}-\lambda\Delta)\|_p^p \leq \lambda \boldsymbol{res}_p(\Delta).$$

Proof. Applying Lemma 2.14 to all coordinates,

$$-\boldsymbol{g}^\top\Delta + \frac{p-1}{p2^p}\gamma_p(|\boldsymbol{Nx}|, \boldsymbol{N}\Delta) \leq \|\boldsymbol{N}(\boldsymbol{x}-\Delta)\|_p^p - \|\boldsymbol{Nx}\|_p^p \leq -\boldsymbol{g}^\top\Delta + 2^p\gamma_p(|\boldsymbol{Nx}|, \boldsymbol{N}\Delta).$$

From the definition of the residual problem and the above equation, the first inequality of our lemma directly follows. To see the other inequality, from the above equation,

$$\begin{aligned}\|\boldsymbol{Nx}\|_p^p - \|\boldsymbol{N}(\boldsymbol{x}-\lambda\Delta)\|_p^p &\leq \lambda\boldsymbol{g}^\top\Delta - \frac{p-1}{p2^p}\gamma_p(|\boldsymbol{Nx}|, \lambda\boldsymbol{N}\Delta)\\ &\leq \lambda\left(\boldsymbol{g}^\top\Delta - \lambda^{p-1}\frac{p-1}{p2^p}\gamma_p(|\boldsymbol{Nx}|, \boldsymbol{N}\Delta)\right)\\ &= \lambda \cdot \boldsymbol{res}_p(\Delta).\end{aligned}$$

Here, we are using the following property of γ_p,

$$\gamma_p(t, \lambda\Delta) \geq \min\{\lambda^2, \lambda^p\}\gamma_p(t, \Delta).$$

□

Lemma 2.15 is similar to Lemma 2.6, and we can follow the proof of Theorem 2.1 to obtain Theorem 2.11.

3 FAST MULTIPLICATIVE WEIGHT UPDATE ALGORITHM FOR ℓ_p-NORMS

In this section, we will show how to solve the residual problem for $p \geq 2$ as defined in the previous section (Definition 2.3), to a constant approximation. The core of our approach is a multiplicative weight update routine with *width reduction* that is used to speed up the algorithm. For problem instances of size m, this routine returns a constant approximate solution in at most $O(m^{1/3})$ calls to a linear system solver. Such a width reduced multiplicative weight update algorithm was first seen in the context of the maximum flow problem and ℓ_∞-regression in works by Chin et al. [19], Christiano et al. [20]

The first instance of such a width reduced multiplicative weight update algorithm for ℓ_p-regression appeared in the work of Adil et al. [3]. In a further work, the authors improved the dependence on p in the runtime [1]. The following sections are based on the improved algorithm from Adil et al. [1].

3.1 Algorithm for ℓ_p-norm Regression

Recall that our residual problem for $p \geq 2$ is defined as:

$$\max_{A\Delta=0} \quad \boldsymbol{res}_p(\Delta) \stackrel{\text{def}}{=} \boldsymbol{g}^\top \Delta - \Delta^\top \boldsymbol{R} \Delta - \|\boldsymbol{N}\Delta\|_p^p,$$

for some vector $\boldsymbol{g}$ and matrices $\boldsymbol{R}$ and $\boldsymbol{N}$. Also recall that in Algorithm 1, we used a parameter ν, which was used to track the value of $\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star)$ at any iteration t. We will now use this parameter ν to do a binary search on the linear term in $\boldsymbol{res}_p$ and reduce the residual problem to,

$$\begin{aligned} \min_{\Delta} \quad & \Delta^\top \boldsymbol{R}\Delta + \|\boldsymbol{N}\Delta\|_p^p \\ s.t. \quad & \boldsymbol{A}\Delta = 0 \\ & \boldsymbol{g}^\top \Delta = c, \end{aligned} \tag{7}$$

for some constant c. Further, we will use our multiplicative weight update solver to solve problems of this kind to a constant approximation. We start by proving the binary search results.

3.1.1 Binary Search. We first note that, if ν at iteration t is such that $\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) \in (\nu/2, \nu]$, then from Lemma 2.6, the residual at $\boldsymbol{x}^{(t)}$ has optimum value, $\boldsymbol{res}_p(\Delta^\star) \in (\frac{\nu}{32p}, \nu]$. We now consider a parameter ζ that has value between $\frac{\nu}{16p}$ and ν such that $\boldsymbol{res}_p(\Delta^\star) \in (\frac{\zeta}{2}, \zeta]$. We have the following lemma that relates the optimum of problem of the type (7) with ζ.

Lemma 3.1. *Let ζ be such that the residual problem satisfies $\boldsymbol{res}_p(\Delta^\star) \in (\frac{\zeta}{2}, \zeta]$. The following problem has optimum at most 2ζ.*

$$\begin{aligned} \min_{\Delta} \quad & \Delta^\top \boldsymbol{R}\Delta + \|\boldsymbol{N}\Delta\|_p^p \\ s.t. \quad & \boldsymbol{A}\Delta = 0 \\ & \boldsymbol{g}^\top \Delta = \frac{\zeta}{2}. \end{aligned} \tag{8}$$

Further, let $\widetilde{\Delta}$ be a solution to the above problem such that $\widetilde{\Delta}^\top \boldsymbol{R}\widetilde{\Delta} \leq a^2\zeta$ and $\|\boldsymbol{N}\widetilde{\Delta}\|_p^p \leq a^p\zeta$ for some $a > 1$. Then $\frac{\widetilde{\Delta}}{5a^2}$ is a $100a^2$-approximation to the residual problem.

Proof. We have assumed that,

$$\boldsymbol{res}(\Delta^\star) = \boldsymbol{g}^\top \Delta^\star - {\Delta^\star}^\top \boldsymbol{R}\Delta^\star - \left\|\boldsymbol{N}\Delta^\star\right\|_p^p \in \left(\frac{\zeta}{2}, \zeta\right].$$

Since the last 2 terms are strictly non-positive, we must have, $\boldsymbol{g}^\top \Delta^\star \geq \frac{\zeta}{2}$. Since $\Delta^\star$ is the optimum and satisfies $\boldsymbol{A}\Delta^\star = 0$,

$$\frac{d}{d\lambda}\left(\boldsymbol{g}^\top \lambda \Delta^\star - \lambda^2 \Delta^{\star\top}\boldsymbol{R}\Delta^\star - \lambda^p \left\|\boldsymbol{N}\Delta^\star\right\|_p^p\right)_{\lambda=1} = 0.$$

Thus,

$$\boldsymbol{g}^\top \Delta^\star - \Delta^{\star\top}\boldsymbol{R}\Delta^\star - \left\|\boldsymbol{N}\Delta^\star\right\|_p^p = \Delta^{\star\top}\boldsymbol{R}\Delta^\star + (p-1)\left\|\boldsymbol{N}\Delta^\star\right\|_p^p.$$

Since $p \geq 2$, we get the following

$$\Delta^{\star\top}\boldsymbol{R}\Delta^\star + \left\|\boldsymbol{N}\Delta^\star\right\|_p^p \leq \boldsymbol{g}^\top \Delta^\star - \Delta^{\star\top}\boldsymbol{R}\Delta^\star - \left\|\boldsymbol{N}\Delta^\star\right\|_p^p \leq \zeta.$$

Now, we know that, $\boldsymbol{g}^\top \Delta^\star \geq \frac{\zeta}{2}$ and $\boldsymbol{g}^\top \Delta^\star - \Delta^{\star\top}\boldsymbol{R}\Delta^\star - \left\|\boldsymbol{N}\Delta^\star\right\|_p^p \leq \zeta$. This gives,

$$\frac{\zeta}{2} \leq \boldsymbol{g}^\top \Delta^\star \leq \Delta^{\star\top}\boldsymbol{R}\Delta^\star + \left\|\boldsymbol{N}\Delta^\star\right\|_p^p + \zeta \leq 2\zeta.$$

Now, let $\widetilde{\Delta}$ be as described in the lemma. We have,

$$\begin{aligned} \boldsymbol{res}_p\left(\frac{\widetilde{\Delta}}{5a^2}\right) &= \frac{1}{5a^2}\boldsymbol{g}^\top\widetilde{\Delta} - \frac{\zeta}{25a^2} - \frac{\zeta}{5^p a^p} \\ &\geq \frac{\zeta}{10a^2} - \frac{2\zeta}{25a^2} \\ &\geq \frac{\zeta}{50a^2} \geq \frac{1}{100a^2}\boldsymbol{res}_p(\Delta^\star) \end{aligned}$$

□

Algorithm 3 Algorithm for Solving the Residual Problem

```
procedure ResidualSolver($\boldsymbol{x}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{A}, \boldsymbol{d}, \boldsymbol{b}, \nu, p$)
    $\zeta \leftarrow \nu$
    $(\boldsymbol{g}, \boldsymbol{R}, \boldsymbol{N}) \leftarrow \boldsymbol{res}_p$                                  ▷ Create residual problem at $\boldsymbol{x}$
    while $\zeta > \frac{\nu}{32p}$ do
        $\widetilde{\Delta}_\zeta \leftarrow$ MWU-Solver$\left([\boldsymbol{A}, \boldsymbol{g}^\top], \boldsymbol{R}^{1/2}, \boldsymbol{N}, [0, \frac{\zeta}{2}]^\top, \zeta, p\right)$        ▷ Algorithm 5
        $\zeta \leftarrow \frac{\zeta}{2}$
    return $\arg\min_{\widetilde{\Delta}_\zeta} \boldsymbol{f}\left(\boldsymbol{x} - \frac{\widetilde{\Delta}_\zeta}{p}\right)$
```

3.1.2 Width-Reduced Approximate Solver. We are now finally ready to solve problems of the type (7). In this section, we will give an algorithm to solve the following problem,

$$\begin{aligned} \min_{\Delta} \quad & \Delta^\top \boldsymbol{M}^\top \boldsymbol{M}\Delta + \|\boldsymbol{N}\Delta\|_p^p \\ \text{s.t.} \quad & \boldsymbol{A}\Delta = \boldsymbol{c}. \end{aligned} \tag{9}$$

Here $\boldsymbol{A} \in \mathbb{R}^{d\times n}$, $\boldsymbol{N} \in \mathbb{R}^{m_1\times n}$, $\boldsymbol{M} \in \mathbb{R}^{m_2\times n}$, and vector $\boldsymbol{c} \in \mathbb{R}^d$. Our approach involves a multiplicative weight update method with a *width reduction* step which allows us to solve these problems faster.

3.1.3 Slow Multiplicative Weight Update Solver. We first give an informal analysis of the multiplicative weight update method without width reduction. We will show that this method converges in $\approx m_1^{\frac{p-2}{2(p-1)}} \leq m_1^{1/2}$ iterations. For simplicity, we let $\boldsymbol{M} = 0$ in Problem (9) and assume without loss of generality that the optimum $\Delta^\star$ satisfies, $\|\boldsymbol{N}\Delta^\star\|_p \leq 1$. Consider the following MWU algorithm for parameter α that we will set later:

(1) $\boldsymbol{w}^{(0)} = 1, \boldsymbol{x}^{(0)} = 0, T = \alpha^{-1} m^{1/p}$
(2) for $t = 1, \cdots, T$:
$\Delta^{(t)} = \arg\min_{\boldsymbol{A}\Delta=\boldsymbol{c}} \sum_i (\boldsymbol{w}_i^{(t-1)})^{p-2}(\boldsymbol{N}\Delta)_i^2, \quad \boldsymbol{w}^{(t)} = \boldsymbol{w}^{(t-1)} + \alpha|\boldsymbol{N}\Delta^{(t)}|, \quad \boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \Delta^{(t)}$
(3) Return $\widetilde{\boldsymbol{x}} = \boldsymbol{x}/T$

We claim that the above algorithm returns $\widetilde{\boldsymbol{x}}$ such that $\|\boldsymbol{N}\widetilde{\boldsymbol{x}}\|_p^p \leq O_p(1)$, i.e., a constant approximate solution to the residual problem, in $\approx m_1^{1/2}$ iterations. We will bound the value of the returned solution, $\|\boldsymbol{N}\widetilde{\boldsymbol{x}}\|_p^p$ by looking at how $\|\boldsymbol{w}^{(t)}\|_p^p$ grows with t. From Lemma 2.5,

$$\|\boldsymbol{w}^{(t-1)} + \alpha\boldsymbol{N}\Delta^{(t)}\|_p^p \leq \|\boldsymbol{w}^{(t-1)}\|_p^p + \alpha p \sum_i (\boldsymbol{w}_i^{(t-1)})^{p-1}(\boldsymbol{N}\Delta^{(t)})_i + 2p^2\alpha^2 \sum_i (\boldsymbol{w}_i^{(t-1)})^{p-2}(\boldsymbol{N}\Delta^{(t)})_i^2 + \alpha^p p^p \|\boldsymbol{N}\Delta^{(t)}\|_p^p.$$

Observe that the third term on the right hand side is exactly the objective of the quadratic problem minimized to obtain $\Delta^{(t)}$. Using that $\Delta^{(t)}$ must achieve a lower objective than $\Delta^\star$, i.e., $\sum_i (\boldsymbol{w}_i^{(t-1)})^{p-2}(\boldsymbol{N}\Delta^{(t)})_i^2 \leq \sum_i (\boldsymbol{w}_i^{(t-1)})^{p-2}(\boldsymbol{N}\Delta^\star)_i^2$ along with Hölder's inequality and $\|\boldsymbol{N}\Delta^\star\|_p \leq 1$, we can bound this term by $\|\boldsymbol{w}^{(t-1)}\|_p^{p-2}$. We can further bound the second term in right hand side of the above inequality by the third term using Hölder's inequality (refer to Proof of Lemma 3.3 for details). These bounds give,

$$\|\boldsymbol{w}^{(t)}\|_p^p \leq \|\boldsymbol{w}^{(t-1)}\|_p^p + \alpha p \|\boldsymbol{w}^{(t-1)}\|_p^{p-1} + 2\alpha^2 p^2 \|\boldsymbol{w}^{(t-1)}\|_p^{p-2} + \alpha^p p^p \|\boldsymbol{N}\Delta^{(t)}\|_p^p.$$

Observe that the growth of $\|\boldsymbol{w}^{(t)}\|_p^p$ is controlled by $\|\boldsymbol{N}\Delta^{(t)}\|_p^p$. We next see how large this quantity can be. Assume that, $\|\boldsymbol{w}^{(t)}\|_p \leq 3m_1^{1/p}$ for all t (one may verify in the end that this holds for all $t \leq T$). Since $(\boldsymbol{w}_i^{(t-1)})^{p-2} \geq (\boldsymbol{w}_i^{(0)})^{p-2} = 1$,

$$\|\boldsymbol{N}\Delta^{(t)}\|_2^2 \leq \sum_i (\boldsymbol{w}_i^{(t-1)})^{p-2}(\boldsymbol{N}\Delta^{(t)})_i^2 \overset{(a)}{\leq} \|\boldsymbol{w}^{(t-1)}\|_p^{p-2} \leq 3^{p-2} m_1^{(p-2)/p},$$

where we used Hölder's inequality in (a). This implies, $\|\boldsymbol{N}\Delta^{(t)}\|_p^p \leq 3^{(p-2)p/2} m_1^{(p-2)/2}$. Now, for $\alpha \approx m_1^{-\frac{p^2-4p+2}{2p(p-1)}}$, $\alpha^p p^p \|\boldsymbol{N}\Delta^{(t)}\|_p^p \leq \alpha p m_1^{\frac{p-1}{p}} \leq \alpha p \|\boldsymbol{w}^{(t-1)}\|_p^{p-1}$ and,

$$\|\boldsymbol{w}^{(t)}\|_p^p \leq \|\boldsymbol{w}^{(t-1)}\|_p^p + \alpha p \|\boldsymbol{w}^{(t-1)}\|_p^{p-1} + 2\alpha^2 p^2 + \alpha p \|\boldsymbol{w}^{(t-1)}\|_p^{p-1} \leq \left(\|\boldsymbol{w}^{(t-1)}\|_p + 2\alpha\right)^p.$$

We can thus prove that,

$$\|\boldsymbol{N}\boldsymbol{x}^{(T)}\|_p^p \leq \frac{1}{m_1}\|\boldsymbol{w}^{(T)}\|_p^p \leq \left(\|\boldsymbol{w}^{(0)})\|_p + 2\alpha T\right)^p = \frac{1}{m_1}\left(m_1^{1/p} + 2m_1^{1/p}\right)^p = 3^p,$$

as required. The total number of iterations is $T = \alpha^{-1} m_1^{1/p} \approx m_1^{\frac{p-2}{2(p-1)}}$.

To obtain the improved rates of convergence via width reduction, our algorithm uses a hard threshold on $\|\boldsymbol{N}\Delta^{(t)}\|_p^p$ and performs a *width reduction step* whenever $\|\boldsymbol{N}\Delta^{(t)}\|_p^p$ is larger than the threshold. The analysis now requires to additionally track how $\|\boldsymbol{w}\|_p$ changes with a width reduction step. Our analysis also tracks the value of an additional potential $\Psi = \min_{\boldsymbol{A}\Delta=\boldsymbol{c}} \sum_i \boldsymbol{w}_i \Delta_i^2$.

The interplay of these two potentials and balancing out their changes with respect to primal updates and width reduction steps give the improved rates of convergence.

3.1.4 Fast, Width-Reduced MWU Solver. In the previous section, we showed that a multiplicative weight update algorithm without width-reduction obtains a rate of convergence $\approx m_1^{1/2}$. In this section we will show how width-reduction allows for a faster $\approx m_1^{1/3}$ rate of convergence. We now present the faster width-reduced algorithm. We will prove the following result.

Theorem 3.2. *Let $p \geq 2$. Consider an instance of Problem (9) described by matrices $\boldsymbol{A} \in \mathbb{R}^{d \times n}, \boldsymbol{N} \in \mathbb{R}^{m_1 \times n}, \boldsymbol{M} \in \mathbb{R}^{m_2 \times n}$, and vector $\boldsymbol{c} \in \mathbb{R}^d$. If the optimum of this problem is at most ζ, Procedure* Residual-Solver *(Algorithm 5) returns an $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$, and $\boldsymbol{x}^\top \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} \leq O(1)\zeta$ and $\|\boldsymbol{N}\boldsymbol{x}\|_p^p \leq O(3^p)\zeta$. The algorithm makes $O\left(pm_1^{\frac{p-2}{(3p-2)}}\right)$ calls to a linear system solver.*

The algorithm and analyses of this chapter are based on [3] and [1].

In every iteration of the algorithm, we solve a weighted linear system. The solution returned is used to update the current iterate if it has a small ℓ_p norm. Otherwise, we do not update the solution, but update the weights corresponding to the coordinates with large value by a constant factor. This step is refered to as the “width reduction step". The analysis is based on a potential function argument for specially defined potentials.

The following is the oracle used in the algorithm, i.e., the linear system we need to solve. We show in the Appendix A how to implement the oracle using a linear system solver.

Algorithm 4 Oracle

```
procedure Oracle($\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{c}, \boldsymbol{w}, \zeta$)
    $\boldsymbol{r}_e \leftarrow \boldsymbol{w}_e^{p-2}$
    $\widetilde{\boldsymbol{M}} \leftarrow \zeta^{-\frac{p-2}{2p}} \boldsymbol{M}$
    Compute,
        $$\Delta = \arg\min_{\boldsymbol{A}\Delta' = \boldsymbol{c}} \quad m_1^{\frac{p-2}{p}} \Delta'^\top \widetilde{\boldsymbol{M}}^\top \widetilde{\boldsymbol{M}} \Delta' + \frac{1}{3^{p-2}} \sum_e \boldsymbol{r}_e \left(\boldsymbol{N}\Delta'\right)_e^2$$
    return $\Delta$
```

We now have the following multiplicative weight update algorithm given in Algorithm 5.

Notation. We will use $\Delta^\star$ to denote the optimum of (9). Since we assume that the optimum value of (9) is at most ζ,

$$\Delta^{\star\top} \boldsymbol{M}^\top \boldsymbol{M} \Delta^\star \leq \zeta \quad \text{and} \quad \|\boldsymbol{N}\Delta^*\|_p^p \leq \zeta \tag{10}$$

3.1.5 Analysis of Algorithm 5. Our analysis is based on tracking the following two potential functions. We will show how these potentials change with a primal step (Line 13) and a width reduction step (18) in the algorithm. The proofs of these lemmas appear later in the section.

$$\Phi\left(\boldsymbol{w}^{(i)}\right) \stackrel{\text{def}}{=} \|\boldsymbol{w}\|_p^p$$

$$\Psi(\boldsymbol{r}) \stackrel{\text{def}}{=} \min_{\Delta : \boldsymbol{A}\Delta = \boldsymbol{c}} m_1^{\frac{p-2}{p}} \Delta^\top \widetilde{\boldsymbol{M}}^\top \widetilde{\boldsymbol{M}} \Delta + \frac{1}{3^{p-2}} \sum_e \boldsymbol{r}_e (\boldsymbol{N}\Delta)_e^2.$$

Algorithm 5 Width Reduced MWU Algorithm

```
procedure MWU-Solver($\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{c}, \zeta, p$)
    $\boldsymbol{w}_e^{(0,0)} \leftarrow 1$
    $\boldsymbol{x} \leftarrow 0$
    $\rho \leftarrow m_1^{\frac{(p^2-4p+2)}{p(3p-2)}}$                                ▷ width parameter
    $\beta \leftarrow 3^{p-1} \cdot m_1^{\frac{p-2}{3p-2}}$                           ▷ resistance threshold
    $\alpha \leftarrow 3^{-\frac{p-1}{p}} \cdot p^{-1} m_1^{-\frac{p^2-5p+2}{p(3p-2)}}$     ▷ step size
    $\tau \leftarrow 3^p \cdot m_1^{\frac{(p-1)(p-2)}{(3p-2)}}$                        ▷ $\ell_p$ threshold
    $T \leftarrow \alpha^{-1} m_1^{1/p} = 3^{\frac{p-1}{p}} \left(p m^{\frac{p-2}{3p-2}}\right)$
    $i \leftarrow 0, k \leftarrow 0$
    while $i < T$ do
        $\Delta \leftarrow$ Oracle($\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{c}, \boldsymbol{w}^{(i,k)}, \zeta$)
        $\boldsymbol{r} \leftarrow \left(\boldsymbol{w}^{(i,k)}\right)^{p-2}$
        if $\|\boldsymbol{N}\Delta\|_p^p \leq \tau\zeta$ then                           ▷ primal step
            $\boldsymbol{w}^{(i+1,k)} \leftarrow \boldsymbol{w}^{(i,k)} + \alpha \frac{|\boldsymbol{N}\Delta|}{\zeta^{1/p}}$
            $\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta$
            $i \leftarrow i + 1$
        else
            For all coordinates $e$ with $|\boldsymbol{N}\Delta|_e \geq \rho\zeta^{\frac{1}{p}}$ and $\boldsymbol{r}_e \leq \beta$    ▷ width reduction step
                $\boldsymbol{w}_e^{(i,k+1)} \leftarrow 2^{\frac{1}{p-2}} \boldsymbol{w}_e$
                $k \leftarrow k + 1$
    return $\frac{\boldsymbol{x}}{T}$
```

Finally, to prove our runtime bound, we will first show that if the total number of width reduction steps K is not too large, then Φ is bounded. We then prove that the number of width reduction steps cannot be too large by using the relation between Φ and Ψ and their respective changes throughout the algorithm.

We now begin our analysis. The next two lemmas show how our potentials change with every iteration of the algorithm.

Lemma 3.3. *After i primal steps, and k width-reduction steps, provided $p^p \alpha^p \tau \leq p\alpha m_1^{\frac{p-1}{p}}$, the potential Φ is bounded as follows:*

$$\Phi\left(\boldsymbol{w}^{(i,k)}\right) \leq \left(2\alpha i + m_1^{1/p}\right)^p \left(1 + \frac{2^{\frac{p}{p-2}}}{\rho^2 m_1^{2/p} \beta^{-\frac{2}{p-2}}}\right)^k.$$

Lemma 3.4. *After i primal steps and k width reduction steps, if,*

(1) $\tau^{2/p}\zeta^{2/p} \geq 4 \cdot 3^{p-2} \frac{\Psi(\boldsymbol{r})}{\beta}$, *and*

(2) $\tau\zeta^{2/p} \geq 2 \cdot 3^{p-2} \Psi(\boldsymbol{r}) \rho^{p-2}$,

then,

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) \geq \Psi\left(\boldsymbol{r}^{(0,0)}\right) + \frac{k}{4} \cdot \tau^{2/p}\zeta^{2/p}.$$

The next lemma gives a lower bound on the energy in the beginning and an upper bound on the energy at each step.

Lemma 3.5. *Let i denote the number of primal steps and k the number of width reduction steps. For any $i, k \geq 0$, we have,*

$$\Psi\left(\boldsymbol{r}^{(i,k)}\right) \leq \zeta^{2/p}\left(m_1^{\frac{p-2}{p}} + \frac{1}{3^{p-2}}\Phi(i,k)^{\frac{p-2}{p}}\right).$$

3.1.6 *Proof of Theorem 3.2.*

Proof. Let $\frac{\boldsymbol{x}}{T}$ be the solution returned by Algorithm 5. We first note that this satisfies the linear constraint required. We next bound the objective value at $\frac{\boldsymbol{x}}{T}$, i.e., $\frac{1}{T^2}\boldsymbol{x}^\top \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x}$ and $\frac{1}{T^p}\|\boldsymbol{N}\boldsymbol{x}\|_p^p$.

Suppose the algorithm terminates in $T = \alpha^{-1}m_1^{1/p}$ primal steps and $K \leq 2^{-\frac{p}{p-2}}\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}}$ width reduction steps. We next note that our parameter values α and τ are such that $p^p\alpha^p\tau \leq p\alpha m_1^{\frac{p-1}{p}}$. We can now apply Lemma 3.3 to get,

$$\Phi\left(\boldsymbol{w}^{(T,K)}\right) \leq 3^p m_1 e^1 = e \cdot 3^p m_1$$

We next observe from the weight and $\boldsymbol{x}$ update steps in our algorithm that, $\zeta^{1/p}m_1^{-1/p}\boldsymbol{w}^{(T,K)} \geq |\boldsymbol{N}\boldsymbol{x}|$. Thus,

$$\frac{1}{T}\|\boldsymbol{N}\boldsymbol{x}\|_p^p \leq \frac{\zeta}{m_1}\left\|\boldsymbol{w}^{(T,K)}\right\|_p^p = \frac{\zeta}{m_1}\Phi\left(\boldsymbol{w}^{(T,K)}\right) \leq e \cdot 3^p\zeta.$$

We next bound the quadratic term. Let $\widetilde{\Delta}^{(t)}$ denote the solution returned by the oracle in iteration t. Since $\Phi \leq e \cdot 3^p m_1$ for all iterations, we always have from Lemma 3.5 that, $\Psi(\boldsymbol{r}) \leq 4m_1^{\frac{p-2}{p}}\zeta^{2/p}$. We will first bound $\left(\widetilde{\Delta}^{(t)}\right)^\top \boldsymbol{M}^\top \boldsymbol{M}\widetilde{\Delta}^{(t)}$ for every t.

$$\left(\widetilde{\Delta}^{(t)}\right)^\top \boldsymbol{M}^\top \boldsymbol{M}\widetilde{\Delta}^{(t)} = \zeta^{\frac{p-2}{p}}\left(\widetilde{\Delta}^{(t)}\right)^\top \widetilde{\boldsymbol{M}}^\top \widetilde{\boldsymbol{M}}\widetilde{\Delta}^{(t)} \leq \zeta^{\frac{p-2}{p}} m_1^{-\frac{p-2}{p}}\Psi(\boldsymbol{r}) \leq 4\zeta.$$

Now from convexity of $\|\boldsymbol{x}\|_2^2$, we get

$$\left\|\boldsymbol{M}\frac{\boldsymbol{x}}{T}\right\|_2^2 \leq \frac{1}{T^2} \cdot T\sum_t \|\boldsymbol{M}\widetilde{\Delta}^{(t)}\|_2^2 \leq 4\zeta.$$

We have shown that if the number of width reduction steps is bounded by K then our algorithm returns the required solution. We will next prove that we cannot have more than K width reduction steps.

Suppose to the contrary, the algorithm takes a width reduction step starting from step (i,k) where $i < T$ and $k = 2^{-\frac{p}{p-2}}\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}}$. Since the conditions for Lemma 3.3 hold for all preceding steps, we must have $\Phi\big(\boldsymbol{w}^{(i,k)}\big) \leq e \cdot 3^p m_1$ which combined with Lemma 3.5 implies $\Psi \leq 4m_1^{\frac{p-2}{p}}\zeta^{2/p}$. Using this bound on Ψ, we note that our parameter values satisfy the conditions of Lemma 3.4. From lemma 3.4,

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) \geq \Psi\left(\boldsymbol{r}^{(0,0)}\right) + \frac{1}{4}\tau^{2/p}\zeta^{2/p}k.$$

Since our parameter choices ensure $\tau^{2/p}k > \frac{1}{4}m_1$,

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) - \Psi\left(\boldsymbol{r}^{(0,0)}\right) > \frac{m_1}{16}\zeta^{2/p}.$$

Since $\Phi\left(\boldsymbol{w}^{(i,k)}\right) \leq O(3^p)m_1$ and $\Psi \geq 0$, from Lemma 3.5,

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) - \Psi\left(\boldsymbol{r}^{(0,0)}\right) \leq 4m_1^{\frac{p-2}{p}}\zeta^{2/p},$$

which is a contradiction. We can thus conclude that we can never have more than $K = 2^{\frac{-p}{p-2}}\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}}$ width reduction steps, thus concluding the correctness of the returned solution. We next bound the number of oracle calls required. The total number of iterations is at most,

$$T + K \leq \alpha^{-1}m_1^{1/p} + 2^{-p/(p-2)}\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}} \leq O\left(pm_1^{\frac{p-2}{3p-2}}\right).$$

□

3.1.7 Proof of Lemma 3.3. We first prove a simple lemma about the solution $\widetilde{\Delta}$ returned by the oracle, that we will use in our proof.

Lemma 3.6. *Let $p \geq 2$. For any $\boldsymbol{w}$, let $\widetilde{\Delta}$ be the solution returned by Algorithm 4. Then,*

$$\sum_e (\boldsymbol{N}\widetilde{\Delta})_e^2 \leq \sum_e \boldsymbol{r}_e(\boldsymbol{N}\widetilde{\Delta})_e^2 \leq \zeta^{\frac{2}{p}}\|\boldsymbol{w}\|^{p-2}$$

Proof. Since $\widetilde{\Delta}$ is the solution returned by Algorithm 4, and $\Delta^\star$ satisfies the constraints of the oracle, we have,

$$\sum_e \boldsymbol{r}_e(\boldsymbol{N}\widetilde{\Delta})_e^2 \leq \sum_e \boldsymbol{r}_e(\boldsymbol{N}\Delta^*)_e^2 = \sum_e \boldsymbol{w}_e^{p-2}(\boldsymbol{N}\Delta^*)_e^2 \leq \zeta^{2/p}\|\boldsymbol{w}\|_p^{p-2}.$$

In the last inequality we use,

$$\begin{aligned}\sum_e \boldsymbol{w}_e(\boldsymbol{N}\Delta^\star)_e^2 &\leq \left(\sum_e (\boldsymbol{N}\Delta^\star)_e^{2\cdot\frac{p}{2}}\right)^{2/p}\left(\sum_e |\boldsymbol{w}_e|^{(p-2)\cdot\frac{p}{p-2}}\right)^{(p-2)/p}\\ &= \|\boldsymbol{N}\Delta^\star\|_p^2\|\boldsymbol{w}\|_p^{(p-2)/p}\\ &\leq \zeta^{2/p}\|\boldsymbol{w}\|_p^{(p-2)/p}, \text{since } \|\boldsymbol{N}\Delta^*\|_p^p \leq \zeta\ .\end{aligned}$$

Finally, using $\boldsymbol{r}_e \geq 1$, we have $\sum_e(\boldsymbol{N}\Delta)_e^2 \leq \sum_e \boldsymbol{r}_e(\boldsymbol{N}\Delta)_e^2$, concluding the proof. □

Lemma 3.3. *After i primal steps, and k width-reduction steps, provided $p^p\alpha^p\tau \leq p\alpha m_1^{\frac{p-1}{p}}$, the potential Φ is bounded as follows:*

$$\Phi\left(\boldsymbol{w}^{(i,k)}\right) \leq \left(2\alpha i + m_1^{1/p}\right)^p\left(1 + \frac{2^{\frac{p}{p-2}}}{\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}}}\right)^k.$$

Proof. We prove this claim by induction. Initially, $i = k = 0$, and $\Phi\left(\boldsymbol{w}^{(0,0)}\right) = m_1$, and thus, the claim holds trivially. Assume that the claim holds for some $i, k \geq 0$. We will use Φ as an abbreviated notation for $\Phi\left(\boldsymbol{w}^{(i,k)}\right)$ below.

Primal Step. For brevity, we use $\boldsymbol{w}$ to denote $\boldsymbol{w}^{(i,k)}$. If the next step is a *primal* step,

$$\Phi\left(\boldsymbol{w}^{(i+1,k)}\right) = \left\|\boldsymbol{w}^{(i,k)} + \alpha\frac{\left|\boldsymbol{N}\widetilde{\Delta}\right|}{\zeta^{1/p}}\right\|_p^p$$

$$\le \|\boldsymbol{w}\|_p^p + \zeta^{-1/p}\alpha\left|(\boldsymbol{N}\widetilde{\Delta})\right|^\top\left|\nabla\|\boldsymbol{w}\|_p^p\right| + 2p^2\alpha^2\zeta^{-2/p}\sum_e |\boldsymbol{w}_e|^{p-2}\left|\boldsymbol{N}\widetilde{\Delta}\right|_e^2 + \alpha^p p^p \zeta^{-1}\|\boldsymbol{N}\widetilde{\Delta}\|_p^p$$

by Lemma 2.5

We next bound $\left|(\boldsymbol{N}\widetilde{\Delta})\right|^\top\left|\nabla\|\boldsymbol{w}\|_p^p\right|$ by $\zeta^{1/p}p\|\boldsymbol{w}\|_p^{p-1}$. Using Cauchy Schwarz's inequality,

$$\begin{aligned}\left(\sum_e\left|\boldsymbol{N}\widetilde{\Delta}\right|_e\left|\nabla_e\|\boldsymbol{w}\|_p^p\right|\right)^2 =& p^2\left(\sum_e\left|\boldsymbol{N}\widetilde{\Delta}\right|_e|\boldsymbol{w}_e|^{p-2}|\boldsymbol{w}_e|\right)^2\\ \le& p^2\left(\sum_e|\boldsymbol{w}_e|^{p-2}\boldsymbol{w}_e^2\right)\left(\sum_e|\boldsymbol{w}_e|^{p-2}(\boldsymbol{N}\widetilde{\Delta})_e^2\right)\\ =& p^2\|\boldsymbol{w}\|_p^p\sum_e \boldsymbol{r}_e(\boldsymbol{N}\widetilde{\Delta})_e^2\\ \le& p^2\|\boldsymbol{w}\|_p^{2p-2}\zeta^{2/p}, \text{ From Lemma 3.6.}\end{aligned}$$

We thus have,

$$\sum_e\left|\boldsymbol{N}\widetilde{\Delta}\right|_e\left|\nabla_e\|\boldsymbol{w}\|_p^p\right| \le p\|\boldsymbol{w}\|_p^{p-1}\zeta^{1/p}.$$

Using the above bound, we now have,

$$\begin{aligned}\Phi\left(\boldsymbol{w}^{(i+1,k)}\right) \le& \|\boldsymbol{w}\|_p^p + p\alpha\|\boldsymbol{w}\|_p^{p-1} + 2p^2\alpha^2\|\boldsymbol{w}\|_p^{p-2} + p^p\alpha^p\|\boldsymbol{N}\widetilde{\Delta}\|_p^p\\ \le& \|\boldsymbol{w}\|_p^p + p\alpha\|\boldsymbol{w}\|_p^{p-1} + 2p^2\alpha^2\|\boldsymbol{w}\|_p^{p-2} + p\alpha m_1^{\frac{p-1}{p}},\\ &\text{(since } p^p\alpha^p\tau \le p\alpha m_1^{\frac{p-1}{p}}\text{)}\end{aligned}$$

Recall $\|\boldsymbol{w}\|_p^p = \Phi(\boldsymbol{w})$. Since $\Phi \ge m_1$, we have,

$$\Phi\left(\boldsymbol{w}^{(i+1,k)}\right) \le \Phi(\boldsymbol{w}) + p\alpha\Phi(\boldsymbol{w})^{\frac{p-1}{p}} + 2p^2\alpha^2\Phi(\boldsymbol{w})^{\frac{p-2}{p}} + p\alpha\Phi(\boldsymbol{w})^{\frac{p-1}{p}} \le (\Phi(\boldsymbol{w})^{1/p} + 2\alpha)^p.$$

From the inductive assumption, we have

$$\Phi(\boldsymbol{w}) \le \left(2\alpha i + m_1^{1/p}\right)^p\left(1 + \frac{2^{\frac{p}{p-2}}}{\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}}}\right)^k.$$

Thus,

$$\Phi(i+1,k) \le (\Phi(\boldsymbol{w})^{1/p} + 2\alpha)^p \le \left(2\alpha(i+1) + m_1^{1/p}\right)^p\left(1 + \frac{2^{\frac{p}{p-2}}}{\rho^2 m_1^{2/p}\beta^{-\frac{2}{p-2}}}\right)^k$$

proving the inductive claim.

Width Reduction Step. Let $\widetilde{\Delta}$ be the solution returned by the oracle and H denote the set of indices j such that $|\boldsymbol{N}\widetilde{\Delta}|_j \ge \rho\zeta^{1/p}$ and $\boldsymbol{r}_j \le \beta$, i.e., the set of indices on which the algorithm performs width reduction. We have the following:

$$\sum_{j\in H}\boldsymbol{r}_j \le \rho^{-2}\zeta^{-2/p}\sum_{j\in H}\boldsymbol{r}_e(\boldsymbol{N}\Delta)_j^2 \le \rho^{-2}\zeta^{-2/p}\sum_j \boldsymbol{r}_j(\boldsymbol{N}\Delta)_e^2 \le \rho^{-2}\|\boldsymbol{w}\|_p^{p-2} \le \rho^{-2}\Phi^{\frac{p-2}{p}},$$

where we use Lemma 3.6 for the second last inequality. Also,

$$\Phi(\boldsymbol{w}^{(i,k+1)}) \leq \Phi + \sum_{j \in H} \left| \boldsymbol{w}_j^{k+1} \right|^p \leq \Phi + 2^{\frac{p}{p-2}} \sum_{j \in H} |\boldsymbol{w}_j|^p \leq \Phi + 2^{\frac{p}{p-2}} \sum_j \boldsymbol{r}_j^{\frac{p}{p-2}}$$
$$\leq \Phi + 2^{\frac{p}{p-2}} \left(\sum_{j \in H} \boldsymbol{r}_j \right) \left(\max_{j \in H} \boldsymbol{r}_j \right)^{\frac{p}{p-2}-1} \leq \Phi + 2^{\frac{p}{p-2}} \rho^{-2} \Phi^{\frac{p-2}{p}} \beta^{\frac{2}{p-2}}.$$

Again, since $\Phi(\boldsymbol{w}) \geq m_1$,

$$\Phi(\boldsymbol{w}^{(i,k+1)}) \leq \Phi \left(1 + 2^{\frac{p}{p-2}} \rho^{-2} m_1^{-\frac{2}{p}} \beta^{\frac{2}{p-2}} \right) \leq \left(2\alpha i + m_1^{1/p} \right)^p \left(1 + \frac{2^{\frac{p}{p-2}}}{\rho^2 m_1^{2/p} \beta^{-\frac{2}{p-2}}} \right)^k$$

proving the inductive claim. □

3.1.8 *Proof of Lemma 3.4.*

Lemma 3.4. *After i primal steps and k width reduction steps, if,*

(1) $\tau^{2/p}\zeta^{2/p} \geq 4 \cdot 3^{p-2} \frac{\Psi(\boldsymbol{r})}{\beta}$, *and*

(2) $\tau\zeta^{2/p} \geq 2 \cdot 3^{p-2} \Psi(\boldsymbol{r}) \rho^{p-2}$,

then,

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) \geq \Psi\left(\boldsymbol{r}^{(0,0)}\right) + \frac{k}{4} \cdot \tau^{2/p}\zeta^{2/p}.$$

Proof. It will be helpful for our analysis to split the index set into three disjoint parts:

- $S = \left\{ e : |\boldsymbol{N}\Delta_e| \leq \rho\zeta^{1/p} \right\}$
- $H = \left\{ e : |\boldsymbol{N}\Delta_e| > \rho\zeta^{1/p} \text{ and } \boldsymbol{r}_e \leq \beta \right\}$
- $B = \left\{ e : |\boldsymbol{N}\Delta_e| > \rho\zeta^{1/p} \text{ and } \boldsymbol{r}_e > \beta \right\}$.

Firstly, we note

$$\sum_{e \in S} |\boldsymbol{N}\Delta|_e^p \leq \rho^{p-2} \zeta^{\frac{p-2}{p}} \sum_{e \in S} |\boldsymbol{N}\Delta|_e^2 \leq \rho^{p-2} \zeta^{\frac{p-2}{p}} \sum_{e \in S} \boldsymbol{r}_e |\boldsymbol{N}\Delta|_e^2 \leq \rho^{p-2} \zeta^{\frac{p-2}{p}} 3^{p-2} \Psi(\boldsymbol{r}).$$

hence, using Assumption 2

$$\sum_{e \in H \cup B} |\boldsymbol{N}\Delta|_e^p \geq \sum_e |\boldsymbol{N}\Delta|_e^p - \sum_{e \in S} |\boldsymbol{N}\Delta|_e^p \geq \tau\zeta - \rho^{p-2} \zeta^{\frac{p-2}{p}} 3^{p-2} \Psi(\boldsymbol{r}) \geq \frac{1}{2} \tau\zeta.$$

This means,

$$\sum_{e \in H \cup B} (\boldsymbol{N}\Delta)_e^2 \geq \left(\sum_{e \in H \cup B} |\boldsymbol{N}\Delta|_e^p \right)^{2/p} \geq \frac{\tau^{2/p}\zeta^{2/p}}{2}.$$

Secondly we note that,

$$\sum_{e \in B} (\boldsymbol{N}\Delta)_e^2 \leq \beta^{-1} \sum_{e \in B} \boldsymbol{r}_e (\boldsymbol{N}\Delta)_e^2 \leq \beta^{-1} 3^{p-2} \Psi(\boldsymbol{r}).$$

So then, using Assumption 1,

$$\sum_{e \in H} (\boldsymbol{N}\Delta)_e^2 = \sum_{e \in H \cup B} (\boldsymbol{N}\Delta)_e^2 - \sum_{e \in B} (\boldsymbol{N}\Delta)_e^2 \geq \frac{\tau^{2/p}\zeta^{2/p}}{2} - \beta^{-1} 3^{p-2} \Psi(\boldsymbol{r}) \geq \frac{\tau^{2/p}\zeta^{2/p}}{4}.$$

As $\boldsymbol{r}_e \geq 1$, this implies $\sum_{e\in H} \boldsymbol{r}_e (\boldsymbol{N}\Delta)_e^2 \geq \frac{\tau^{2/p}\zeta^{2/p}}{4}$. We note that in a width reduction step, the resistances change by a factor of 2. Thus, combining our last two observations, and applying Lemma C.1, we get

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) \geq \Psi\left(\boldsymbol{r}^{(i,k)}\right) + \frac{1}{4}\tau^{2/p}\zeta^{2/p}.$$

Finally, for the "primal step" case, we use the trivial bound from Lemma C.1, ignoring the second term,

$$\Psi\left(\boldsymbol{r}^{(i,k+1)}\right) \geq \Psi\left(\boldsymbol{r}^{(i,k)}\right).$$

□

3.1.9 *Proof of Lemma 3.5.*

Lemma 3.5. *Let i denote the number of primal steps and k the number of width reduction steps. For any $i, k \geq 0$, we have,*

$$\Psi\left(\boldsymbol{r}^{(i,k)}\right) \leq \zeta^{2/p}\left(m_1^{\frac{p-2}{p}} + \frac{1}{3^{p-2}}\Phi(i,k)^{\frac{p-2}{p}}\right).$$

Proof. Lemma 3.6 implies that,

$$\begin{aligned}\Psi\left(\boldsymbol{r}^{(i,k)}\right) &= \zeta^{-(p-2)/p} m_1^{\frac{p-2}{p}} \widetilde{\Delta}^\top \boldsymbol{M}^\top \boldsymbol{M}\widetilde{\Delta} + \frac{1}{3^{p-2}}\sum_e \boldsymbol{r}_e(\boldsymbol{N}\widetilde{\Delta})_e^2 \\ &\leq \zeta^{-(p-2)/p} m_1^{\frac{p-2}{p}} \Delta^{\star\top} \boldsymbol{M}^\top \boldsymbol{M}\Delta^\star + \frac{1}{3^{p-2}}\sum_e \boldsymbol{r}_e(\boldsymbol{N}\Delta^\star)_e^2 \\ &\leq \zeta^{2/p} m_1^{\frac{p-2}{p}} + \zeta^{2/p}\frac{1}{3^{p-2}}\|\boldsymbol{w}\|_p^{p-2} \\ &\leq \zeta^{2/p} m_1^{\frac{p-2}{p}} + \zeta^{2/p}\frac{1}{3^{p-2}}\Phi(i,k)^{\frac{p-2}{p}}.\end{aligned}$$

□

3.2 Complete Algorithm for ℓ_p-Regression

Recall our problem, (1),

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \quad \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{d}^\top \boldsymbol{x} + \|\boldsymbol{M}\boldsymbol{x}\|_2^2 + \|\boldsymbol{N}\boldsymbol{x}\|_p^p.$$

We will now use all the tools and algorithms described so far to give a complete algorithm for the above problem. We will assume we have a starting solution $\boldsymbol{x}^{(0)}$ satisfying $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$ and for purely ℓ_p objectives, we will use the homotopy analysis from Section 2.2.

Our overall algorithm reduces the problem to solving the residual problem (Definition 2.3) approximately. In Sections 3.1.1 and 3.1.2, we give an algorithm to solve the residual problem by first doing a binary search on the linear term and then applying a multiplicative weight update routine to minimize these problems. We have the following result which follows from Lemma 3.1 and Theorem 3.2.

Corollary 3.7. *Consider the residual problem at iteration t of Algorithm 1. Algorithm 3 using Algorithm 5 as a subroutine finds a $O(1)$-approximate solution to the corresponding residual problem in $O\left(pm^{\frac{p-2}{3p-2}}\log p\right)$ calls to a linear system solver.*

Proof. Let ν be such that $\boldsymbol{f}(\boldsymbol{x}^{(t)}) - \boldsymbol{f}(\boldsymbol{x}^\star) \in (\nu/2, \nu]$. Refer to Lemma 2.7 to see that this is the case in which we use the solution of the residual problem. Now, from Lemma 2.6 we know that the optimum of the residual problem satisfies $\boldsymbol{res}_p(\Delta^\star) \in (\nu/32p, \nu]$. Since we vary ζ to take all such values in the range $(\nu/16p, \nu]$ for one such ζ we must have $\boldsymbol{res}_p(\Delta^\star) \in (\zeta/2, \zeta]$. For such a ζ, consider problem (8). Using Algorithm 5 for this problem, from Theorem 3.2 we are guaranteed to find a solution $\widetilde{\Delta}$ such that $\widetilde{\Delta}^\top \boldsymbol{R}\widetilde{\Delta} \leq O(1)\zeta$ and $\|\boldsymbol{N}\widetilde{\Delta}\|_p^p \leq O(3^p)\zeta$. Now from Lemma 3.1, we note that $\widetilde{\Delta}$ is an $O(1)$-approximate solution to the residual problem. Since Algorithm 5 requires $O\left(pm^{\frac{p-2}{3p-2}}\right)$ calls to a linear system solver, and Algorithm 3 calls this algorithm $\log p$ times, we obtain the required runtime. □

We are now ready to prove our main result.

Theorem 3.8. *Let $p \geq 2$, and $\kappa \geq 1$. Let the initial solution $\boldsymbol{x}^{(0)}$ satisfying $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$. Algorithm 1 using Algorithm 3 as a subroutine returns an ε-approximate solution $\boldsymbol{x}$ to* (1) *in at most $O\left(p^2 m^{\frac{p-2}{3p-2}} \log p \log\left(\frac{f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^\star)}{\varepsilon}\right)\right)$ calls to a linear system solver.*

Proof. Follows from Theorem 2.1 and Corollary 3.7. □

3.3 Complete Algorithm for Pure ℓ_p Objectives

Consider the special case when our problem is only the ℓ_p-norm, i.e., Problem (3),

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \|\boldsymbol{N}\boldsymbol{x}\|_p^p.$$

In Section 2.2 we described how to find a good starting point for such problems. Combining this algorithm with our algorithm for solving the residual problem we can obtain a complete algorithm for finding a good starting point. Specifically, we prove the following result.

Corollary 3.9. *Algorithm 2 using Algorithm 3 returns $\boldsymbol{x}^{(0)}$ such that $\boldsymbol{A}\boldsymbol{x}^{(0)} = \boldsymbol{b}$ and $\|\boldsymbol{N}\boldsymbol{x}^{(0)}\|_p^p \leq O(m)\|\boldsymbol{N}\boldsymbol{x}^\star\|_p^p$ in $O\left(p^2 m^{\frac{p-2}{3p-2}} \log^2 p \log m\right)$ calls to a linear system solver.*

Proof. From Lemma 2.9 Algorithm 2 finds such a solution in time $O(p\log m)\sum_{k=2^i,i=2}^{i=\lfloor \log p-1\rfloor} \kappa_k T(k,\kappa_k)$, where κ_k and $T(k,\kappa_k)$ denote the approximation and time to solve a ℓ_k norm problem. Now consider Algorithm 3 with Algorithm 5 as a subroutine. From Corollary 3.7, we can solve any ℓ_k-norm residual problem to a $O(1)$-approximation in $O\left(km^{\frac{k-2}{3k-2}}\log k\right)$ calls to a linear system solver. We thus have $\kappa_k = O(1)$ for all k and $T(k,\kappa_k) = O\left(km^{\frac{k-2}{3k-2}}\log k\right) \leq O\left(pm^{\frac{p-2}{3p-2}}\log p\right)$. Using these values, we obtain a runtime of,

$$O(p\log m)\sum_{k=2^i,i=2}^{i=\lfloor \log p-1\rfloor} \kappa_k T(k,\kappa_k) \leq O(p\log m)\cdot \log p \cdot O\left(pm^{\frac{p-2}{3p-2}}\log p\right) \leq O\left(p^2 m^{\frac{p-2}{3p-2}}\log^2 p\log m\right).$$

□

The following theorem gives a complete runtime for pure ℓ_p objectives.

Corollary 3.10. *Let $p \geq 2$, and $\kappa \geq 1$. Let $\boldsymbol{x}^{(0)}$ be the solution returned by Algorithm 2. Algorithm 1 using Algorithm 3 as a subroutine returns $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\|\boldsymbol{N}\boldsymbol{x}\|_p^p \leq (1+\varepsilon)\|\boldsymbol{N}\boldsymbol{x}^\star\|_p^p$, in at most $O\left(p^2 m^{\frac{p-2}{3p-2}}\log^2 p\log\left(\frac{m}{\varepsilon}\right)\right)$ calls to a linear system solver.*

Proof. Follows directly from Corollary 3.9 and Theorem 3.8. □

4 SOLVING p-NORM PROBLEMS USING q-NORM ORACLES

In this section, we propose a new technique that allows us to solve ℓ_p-norm residual problems by instead solving an ℓ_q-norm residual problem without adding much to the runtime. Such a technique is unknown for pure ℓ_p objectives without a large overhead in the runtime. As a consequence we also obtain an algorithm for ℓ_p-regression with a linear runtime dependence on p instead of the p^2 dependence in the algorithms from previous sections. The p^2 dependence in algorithms had one p factor resulting from solving the p-norm residual problem. At a high level, we show that it is sufficient to solve a $\log m$-norm residual problem when p is large, thus replacing a p-factor with $\log m$. We prove the following results which are based on the proofs and results of Adil and Sachdeva [6].

Theorem 4.1. *Let $\varepsilon > 0$, $2 \leq p \leq poly(m)$ and consider an instance of Problem* (1),

$$\min_{Ax=b} \quad f(x) = d^\top x + \|Mx\|_2^2 + \|Nx\|_p^p.$$

Algorithm 6 finds an ε-approximate solution to (1) *in $O\left(pm^{\frac{p-2}{3p-2}}\log p \log m \log \frac{f(x^{(0)})-f(x^\star)}{\varepsilon}\right)$ calls to a linear system solver.*

Theorem 4.2. *Let $\varepsilon > 0$, $2 \leq p \leq poly(m)$ and consider a pure ℓ_p instance,*

$$\min_{Ax=b} \quad \|Nx\|_p^p.$$

Let $x^{(0)}$ be the output of Algorithm 2. Algorithm 6 using $x^{(0)}$ as a starting solution finds x such that $Ax = b$ and $\|Nx\|_p^p \leq (1+\varepsilon)\min_{Ax=b}\|Nx\|_p^p$ in $O\left(pm^{\frac{p-2}{3p-2}}\log^2 p \log m \log \frac{m}{\varepsilon}\right)$ calls to a linear system solver.

4.1 Relation between Residual Problems for ℓ_p and ℓ_q Norms

In this section we prove how q-norm residual problems can be used to solve p-norm residual problems. This idea first appeared in the work of Adil and Sachdeva [6], where they also apply the results to the maximum flow problem. In this paper, we provide a much simpler proof for the main techncial content and unify the cases of $p < q$ and $p > q$ that were presented separately in previous works. We also unify the case of relating the decision versions of the residual problems (without the linear term) and the entire objective. The results for the maximum flow problem and ℓ_p-norm flow problem as described in the original paper still follow and we refer the reader to the original paper for these applications. The main result of the section is as follows.

Theorem 4.3. *Let $p, q \geq 2$ and ζ be such that $\boldsymbol{res}_p(\Delta^\star) \in (\zeta/2, \zeta]$, where $\Delta^\star$ is the optimum of the ℓ_p-norm residual problem (Definition 2.3). The following ℓ_q-norm residual problem has optimum at least $\frac{\zeta}{4}$,*

$$\max_{A\Delta=0} g^\top\Delta - \Delta^\top R\Delta - \frac{1}{4}\zeta^{1-\frac{q}{p}} m^{\min\left\{\frac{q}{p}-1,0\right\}}\|N\Delta\|_q^q. \tag{11}$$

Let $\beta \geq 1$ and $\widetilde{\Delta}$ denote a feasible solution to the above ℓ_q-norm residual problem with objective value at least $\frac{\zeta}{16\beta}$. For $\alpha = \frac{1}{256\beta}m^{-\frac{p}{p-1}\left|\frac{1}{p}-\frac{1}{q}\right|}$, $\alpha\widetilde{\Delta}$ gives a $O(\beta^2)m^{\frac{p}{p-1}\left|\frac{1}{p}-\frac{1}{q}\right|}$-approximate solution to the ℓ_p-norm residual problem $\boldsymbol{res}_p$.

Proof. Consider $\Delta^\star$, the optimum of the ℓ_p-norm residual problem. Note that $\lambda\Delta^\star$ is a feasible solution for all λ since $A(\lambda\Delta^\star) = 0$. We know that the objective is optimum for $\lambda = 1$. Thus,

$$\left[\frac{d}{d\lambda}\boldsymbol{res}_p(\lambda\Delta^\star)\right]_{\lambda=1} = 0,$$

which gives us,

$$\boldsymbol{g}^\top \Delta^\star - 2{\Delta^\star}^\top \boldsymbol{R}\Delta^\star - p\|\boldsymbol{N}\Delta^\star\|_p^p = 0.$$

Rearranging,

$${\Delta^\star}^\top \boldsymbol{R}\Delta^\star + (p-1)\|\boldsymbol{N}\Delta^\star\|_p^p = \boldsymbol{g}^\top \Delta^\star - {\Delta^\star}^\top \boldsymbol{R}\Delta^\star - \|\boldsymbol{N}\Delta^\star\|_p^p \leq \zeta.$$

Since $p \geq 2$, $\|\boldsymbol{N}\Delta^\star\|_p \leq \zeta^{1/p}$ which implies

$$\|\boldsymbol{N}\Delta^\star\|_q \leq \begin{cases} \zeta^{1/p} & \text{if, } p \leq q \\ m^{\frac{1}{q}-\frac{1}{p}}\zeta^{1/p} & \text{otherwise.} \end{cases}$$

We also note that,

$$\boldsymbol{g}^\top \Delta^\star - {\Delta^\star}^\top \boldsymbol{R}\Delta^\star > \frac{\zeta}{2} + \|\boldsymbol{N}\Delta^\star\|_p^p > \frac{\zeta}{2}.$$

Combining these bounds, we obtain the optimum of (11) is at least,

$$\boldsymbol{g}^\top \Delta^\star - {\Delta^\star}^\top \boldsymbol{R}\Delta^\star - \frac{1}{4}\zeta^{1-\frac{q}{p}} m^{\min\left\{\frac{q}{p}-1,0\right\}} \|\boldsymbol{N}\Delta^\star\|_q^q > \frac{\zeta}{2} - \frac{1}{4}\zeta^{1-\frac{q}{p}}\zeta^{q/p} > \frac{\zeta}{4}.$$

Since the optimum of (11) is at least $\zeta/4$, there exists a feasible $\widetilde{\Delta}$ with objective value at least $\zeta/16\beta$. We now prove the second part, that a scaling of $\widetilde{\Delta}$ gives a good approximation to the ℓ_p-norm residual problem. First, let us assume $|\boldsymbol{g}^\top\widetilde{\Delta}| \leq \zeta$. Since $\widetilde{\Delta}$ has objective value at least $\zeta/16\beta$,

$$\widetilde{\Delta}^\top \boldsymbol{R}\widetilde{\Delta} + \frac{1}{4}\zeta^{1-\frac{q}{p}} m^{\min\left\{\frac{q}{p}-1,0\right\}} \|\boldsymbol{N}\widetilde{\Delta}\|_q^q \leq \boldsymbol{g}^\top\widetilde{\Delta} - \frac{\zeta}{16\beta} \leq \zeta.$$

Thus, $m^{\min\left\{\frac{1}{p}-\frac{1}{q},0\right\}}\|\boldsymbol{N}\widetilde{\Delta}\|_q \leq 4^{\frac{1}{q}}\zeta^{\frac{1}{p}}$, and $\|\boldsymbol{N}\widetilde{\Delta}\|_p^p \leq 4^{\frac{p}{q}}\zeta m^{\left|1-\frac{p}{q}\right|}$. Let $\bar{\Delta} = \alpha\widetilde{\Delta}$, where $\alpha = \frac{1}{256\beta}m^{-\frac{p}{p-1}\left|\frac{1}{p}-\frac{1}{q}\right|}$. We will show that $\alpha\bar{\Delta}$ is a good solution to the ℓ_p-norm residual problem.

$$\begin{aligned} \boldsymbol{res}_p(\alpha\bar{\Delta}) &= \alpha\left(\boldsymbol{g}^\top\widetilde{\Delta} - \alpha\widetilde{\Delta}^\top \boldsymbol{R}\widetilde{\Delta} - \alpha^{p-1}\|\boldsymbol{N}\widetilde{\Delta}\|_p^p\right) \\ &\geq \alpha\left(\frac{\zeta}{16\beta} - \frac{1}{256\beta}\zeta - \alpha^{p-1}4^{\frac{p}{q}}\zeta m^{\left|1-\frac{p}{q}\right|}\right) \\ &\geq \alpha\left(\frac{\zeta}{16\beta} - \frac{\zeta}{256\beta} - \frac{\zeta}{64\beta}\right) \\ &\geq \frac{\alpha}{64\beta}\boldsymbol{res}_p(\Delta^\star). \end{aligned}$$

For the case $|\boldsymbol{g}^\top\widetilde{\Delta}| \geq \zeta$, consider the vector $z\widetilde{\Delta}$ where $z = \frac{\zeta}{2|\boldsymbol{g}^\top\widetilde{\Delta}|} \leq \frac{1}{2}$. This vector is still feasible for Problem (11) and $\boldsymbol{g}^\top z\widetilde{\Delta} = \frac{\zeta}{2}$ and,

$$z\boldsymbol{g}^\top\widetilde{\Delta} - z^2\widetilde{\Delta}^\top \boldsymbol{R}\widetilde{\Delta} - z^q\frac{1}{4}\zeta^{1-\frac{q}{p}} m^{\min\left\{\frac{q}{p}-1,0\right\}} \|\boldsymbol{N}\widetilde{\Delta}\|_q^q \geq \frac{\zeta}{2} - z^2\zeta \geq \frac{\zeta}{4}.$$

We can now repeat the same argument as above. □

4.2 Faster Algorithm for ℓ_p-Regression

In this section, we will combine the tools developed in previous chapters and combine it with Section 4.1 to obtain an algorithm for Problem 1 that requires $O\left(pm^{\frac{p-2}{3p-2}}\log p \log m \log\frac{f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^\star)}{\varepsilon}\right)$ calls to a linear systems solver. For pure ℓ_p objectives we can combine our algorithm with the algorithm in Section 2.2 to obtain a convergence rate of $O\left(pm^{\frac{p-2}{3p-2}}\log^2 p \log m \log\frac{m}{\varepsilon}\right)$ linear systems solves.

Algorithm 6 Complete Algorithm with Linear p-dependence

```
procedure $\ell_p$-Solver($A, M, N, d, b, p, \varepsilon$)
    $x \leftarrow x^{(0)}$
    $\nu \leftarrow$ Upper bound on $f(x^{(0)}) - f(x^{\star})$        ▹ If $f(x^{\star}) \geq 0$, then $\nu \leftarrow f(x^{(0)})$
    while $\nu > \varepsilon$ do
        if $p \geq \log m$ then
            $\widetilde{\Delta} \leftarrow \log m$-ResidualSolver($x, M, N, A, d, b, \nu, p$)
        else
            $\widetilde{\Delta} \leftarrow$ ResidualSolver($x, M, N, A, d, b, \nu, p$)
        if $res_p(\widetilde{\Delta}) \geq \frac{\nu}{32p\kappa}$ then
            $x \leftarrow x - \frac{\widetilde{\Delta}}{p}$
        else
            $\nu \leftarrow \frac{\nu}{2}$
    return $x$
```

Algorithm 7 Residual Solver using $\log m$-norm

```
procedure $\log m$-ResidualSolver($x, M, N, A, d, b, \nu, p$)
    $\zeta \leftarrow \nu$
    $\alpha \leftarrow m^{-\frac{1}{p-1}}$
    $(g, R, N) \leftarrow res_p$        ▹ Create residual problem at $x$
    while $\zeta > \frac{\nu}{32p}$ do
        $\widetilde{N} \leftarrow \frac{1}{4^{1/\log m}} \zeta^{\frac{1}{\log m} - \frac{1}{p}} m^{\min\left\{\frac{1}{p} - \frac{1}{\log m}, 0\right\}} N$
        $\widetilde{\Delta}_{\zeta} \leftarrow$ MWU-Solver$\left([A, g^{\top}], R^{1/2}, \widetilde{N}, [0, \frac{\zeta}{2}]^{\top}, \zeta, \log m\right)$        ▹ Algorithm 5
        $\zeta \leftarrow \frac{\zeta}{2}$
    return $\alpha\widetilde{\Delta} \leftarrow \arg\min_{\widetilde{\Delta}_{\zeta}} f\left(x - \frac{\alpha\widetilde{\Delta}_{\zeta}}{p}\right)$
```

Lemma 4.4. *Let $poly(m) \geq p \geq \log m$. Algorithm 7 returns an $O(m^{\frac{1}{p-1}})$-approximate solution to the ℓ_p-residual problem res_p at x in at most $O\left(m^{\frac{p-2}{3p-2}} \log m \log p\right)$ calls to a linear system solver.*

Proof. Let ν be such that $f(x^{(t)}) - f(x^{\star}) \in (\nu/2, \nu]$. Refer to Lemma 2.7 to see that this is the case in which we use the solution of the residual problem. Now, from Lemma 2.6 we know that the optimum of the residual problem satisfies $res_p(\Delta^{\star}) \in (\nu/32p, \nu]$. Since we vary ζ to take all such values in the range $(\nu/16p, \nu]$ for one such ζ we must have $res_p(\Delta^{\star}) \in (\zeta/2, \zeta]$. For such a ζ, consider the $\log m$-norm residual problem (11). Using Algorithm 5 for this problem, from Theorem 3.2 we are guaranteed to find a solution $\widetilde{\Delta}$ such that $\widetilde{\Delta}^{\top} R \widetilde{\Delta} \leq O(1)\zeta$ and $\|\widetilde{N}\widetilde{\Delta}\|_{\log m}^{\log m} \leq O(3^p)\zeta$. Now from Lemma 3.1, we note that $\widetilde{\Delta}$ is an $O(1)$-approximate solution to the $\log m$-residual problem. We now use Theorem 4.3, which states that $\alpha\widetilde{\Delta}$ is a $O\left(m^{\frac{1}{p-1}}\right)$-approximate solution to the required residual problem res_p.

Since for $p \geq \log m$, Algorithm 5 requires $O\left(m^{\frac{\log m - 2}{3\log m - 2}} \log m\right) \leq O\left(m^{\frac{p-2}{3p-2}} \log m\right)$ calls to a linear system solver, and Algorithm 3 calls this algorithm $\log p$ times, we obtain the required runtime. □

Theorem 4.1. *Let $\varepsilon > 0$, $2 \leq p \leq poly(m)$ and consider an instance of Problem* (1),

$$\min_{\boldsymbol{Ax}=\boldsymbol{b}} \quad f(\boldsymbol{x}) = \boldsymbol{d}^{\top}\boldsymbol{x} + \|\boldsymbol{Mx}\|_2^2 + \|\boldsymbol{Nx}\|_p^p.$$

Algorithm 6 finds an ε-approximate solution to (1) *in $O\left(pm^{\frac{p-2}{3p-2}} \log p \log m \log \frac{f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^{\star})}{\varepsilon}\right)$ calls to a linear system solver.*

Proof. We note that Algorithm 6 is essentially Algorithm 1 which calls different residual solvers depending on the value of p. If $p \leq \log m$, from Theorem 3.8, we obtain the required solution in $O\left(m^{\frac{p-2}{3p-2}} \log p \log \frac{f(\boldsymbol{x}^{(0)})-f(\boldsymbol{x}^{\star})}{\varepsilon}\right)$ calls to a linear system solver. If $p \geq \log m$, from Lemma 4.4, we obtain an $O(m^{\frac{1}{p-1}}) \leq O(m^{\frac{1}{\log m}}) \leq O(1)$ approximate solution to the residual problem at any iteration in $O\left(m^{\frac{p-2}{3p-2}} \log m \log p\right)$ calls to a linear system solver. Combining this with Theorem 2.1, we obtain our result. □

Theorem 4.2. *Let $\varepsilon > 0$, $2 \leq p \leq poly(m)$ and consider a pure ℓ_p instance,*

$$\min_{\boldsymbol{Ax}=\boldsymbol{b}} \quad \|\boldsymbol{Nx}\|_p^p.$$

Let $\boldsymbol{x}^{(0)}$ be the output of Algorithm 2. Algorithm 6 using $\boldsymbol{x}^{(0)}$ as a starting solution finds $\boldsymbol{x}$ such that $\boldsymbol{Ax} = \boldsymbol{b}$ and $\|\boldsymbol{Nx}\|_p^p \leq (1+\varepsilon)\min_{\boldsymbol{Ax}=\boldsymbol{b}} \|\boldsymbol{Nx}\|_p^p$ in $O\left(pm^{\frac{p-2}{3p-2}} \log^2 p \log m \log \frac{m}{\varepsilon}\right)$ calls to a linear system solver.

Proof. From Lemma 2.9 we can find an $O(m)$-approximation to the above problem in time

$$O(p\log m) \sum_{k=2^i, i=2}^{i=\lfloor \log p-1 \rfloor} \kappa_k T(k, \kappa_k),$$

where κ is the approximation to which we solve the residual problem for the k-norm problem and $T(k,\kappa)$ is the time required to do so. If $k \geq \log m$, we use Algorithm 7 to solve such residual problems. Thus $\kappa_k = m^{\frac{1}{k-1}} \leq m^{\frac{1}{\log m}} \leq O(1)$ and $T(k,\kappa_k) = O\left(m^{\frac{p-2}{3p-2}} \log p\right)$. If $k \leq \log m$, we can use Algorithm 3 and $\kappa_k = O(1)$, $T(k,\kappa_k) = O\left(m^{\frac{p-2}{3p-2}} \log p\right)$. Thus, the total runtime is $O\left(pm^{\frac{p-2}{3p-2}} \log m \log^2 p\right)$. We now combine this with Theorem 4.1 to obtain the required rates of convergence. □

5 SPEEDUPS FOR GENERAL MATRICES VIA INVERSE MAINTENANCE

Inverse maintenance was first introduced by Vaidya in 1990 [45] for speeding up algorithms for minimum cost and multicommodity flow problems. The key idea is to reuse the inverse of matrices, which is possible due to the controllable rates at which variables are updated in some algorithms. In the work by Adil et al. [3], the authors design a new inverse maintenance algorithm for ℓ_p-regression that can solve ℓ_p-regression for any $p > 2$ almost as fast as linear regression. This section is based on Section 6 of [3] and we give a more fine grained and simplified analysis of the original result. In particular, we simplify the proofs and give the result with explicit dependencies on both matrix dimensions as opposed to just the larger dimension.

Our inverse maintenance procedure is based on the same high-level ideas of combining low-rank updates and matrix multiplication as in Vaidya [45] and Lee and Sidford [35]. However, recall that the rate of convergence of our algorithm is controlled by two potentials which change at different rates based on the two different kind of weight update steps in our algorithm. In order to handle these updates, our inverse maintenance algorithm uses a new fine-grained bucketing scheme, inspired by lazy updates in data structures and is different from previous works on inverse maintenance which usually update weights based on fixed thresholds. Our scheme is also simpler than those used in [35, 45]. We now present our algorithm in detail.

Consider the weighted linear system being solved at each iteration of Algorithm 5. Each weighted linear system is of the form,

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{c}} \boldsymbol{x}^\top \left(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\right)\boldsymbol{x}$$

where $\boldsymbol{A} \in \mathbb{R}^{d\times n}, \boldsymbol{N} \in \mathbb{R}^{m_1\times n}, \boldsymbol{M} \in \mathbb{R}^{m_2\times n}$. From Equation (15) in Section 3, the solution of the above linear system is given by,

$$\boldsymbol{x}^\star = \left(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\right)^{-1}\boldsymbol{A}^\top\left(\boldsymbol{A}\left(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\right)^{-1}\boldsymbol{A}^\top\right)^{-1}\boldsymbol{c}.$$

In order to compute the above expression, we require the following products in order. The runtimes are considering the fact $\omega \geq 2$.

- $\boldsymbol{M}^\top\boldsymbol{M}$ and $\boldsymbol{N}^\top\boldsymbol{R}\boldsymbol{N}$: require time $m_2 n^{\omega-1}$ and $m_1 n^{\omega-1}$ respectively
- $(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N})^{-1}$: requires time n^ω
- $(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N})^{-1}\boldsymbol{A}^\top$ and $\boldsymbol{A}(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N})^{-1}\boldsymbol{A}^\top$: require time $n^2 d^{\omega-2}$
- $\left(\boldsymbol{A}(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N})^{-1}\boldsymbol{A}^\top\right)^{-1}$: requires time d^ω
- $(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N})^{-1}\boldsymbol{A}^\top\left(\boldsymbol{A}(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N})^{-1}\boldsymbol{A}^\top\right)^{-1}$: requires time $nd^{\omega-1}$

The cost of solving the above problem is dominated by the first step, and we thus require time $O(mn^{\omega-1})$, where $m = \max\{m_1, m_2\}$. This directly gives the runtime of Algorithm 5 to be $O\left(pm^{\frac{p-2}{(3p-2)}} mn^{\omega-1}\right)$. In this section, we show that we can implement Algorithm 5 in time similar to solving a system of linear equations for all $p \geq 2$. In particular, we prove the following result.

Theorem 5.1. *If $\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}$ are explicitly given, matrices with polynomially bounded condition numbers, and $p \geq 2$, Algorithm 5 as given in Section 3.1.2 can be implemented to run in total time*

$$O\left(mn^{\omega-1} + p^{3-\omega}n^2 m^{\omega-2} + p^{3-\omega}n^2 m^{\frac{p-(10-4\omega)}{3p-2}}\right).$$

5.1 Inverse Maintenance Algorithm

We first note that the weights $\boldsymbol{w}_e^{(i)}$'s, and thus $\boldsymbol{r}_e^{(i)}$'s are monotonically increasing. Our algorithm in Section 3.1.2 updates both in every iteration. Here, we will instead update these gradually when there is a significant increase in the values. We thus give a lazy update scheme. The update can be done via the following consequence of the Woodbury matrix formula. The main idea is that we initially explicitly compute the inverse of the required matrix, and then when we update the coordinates that have significant increases, but are still within a good factor approximation of the original values, and directly use the current matrix inverse as a preconditioner and solve linear systems faster.

5.1.1 Low Rank Update. The following lemma is the same as Lemma 6.2 of Adil et al. [3].

Lemma 5.2. *Given matrices $\boldsymbol{N} \in \mathbb{R}^{m_1\times n}, \boldsymbol{M} \in \mathbb{R}^{m_2\times n}$, and vectors $\boldsymbol{r}$ and $\tilde{\boldsymbol{r}}$ that differ in k entries, as well as the matrix $\widehat{\boldsymbol{Z}} = (\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\boldsymbol{r})\boldsymbol{N})^{-1}$, we can construct $(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\tilde{\boldsymbol{r}})\boldsymbol{N})^{-1}$ in $O(k^{\omega-2}n^2)$ time.*

Proof. Let S denote the entries that differ in $\boldsymbol{r}$ and $\tilde{\boldsymbol{r}}$. Then we have

$$\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\tilde{\boldsymbol{r}})\boldsymbol{N} = \boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\boldsymbol{r})\boldsymbol{N} + \boldsymbol{N}_{:,S}^\top(Diag(\tilde{\boldsymbol{r}}_S) - Diag(\boldsymbol{r}_S))\boldsymbol{N}_{S,:}.$$

This is a low rank perturbation, so by Woodbury matrix identity we get:

$$\left(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\tilde{\boldsymbol{r}})\boldsymbol{N}\right)^{-1} = \widehat{\boldsymbol{Z}} - \widehat{\boldsymbol{Z}}\boldsymbol{N}_{:,S}^\top\left((Diag(\tilde{\boldsymbol{r}}_S) - Diag(\boldsymbol{r}_S))^{-1} + \boldsymbol{N}_{S,:}\widehat{\boldsymbol{Z}}\boldsymbol{N}_{:,S}^\top\right)^{-1}\boldsymbol{N}_{S,:}\widehat{\boldsymbol{Z}},$$

where we use $\widehat{\boldsymbol{Z}}^\top = \widehat{\boldsymbol{Z}}$ because $\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\boldsymbol{r})\boldsymbol{N}$ is a symmetric matrix. To explicitly compute this matrix, we need to:

(1) compute the matrix $\boldsymbol{N}_{S,:}\widehat{\boldsymbol{Z}}$,
(2) compute $\boldsymbol{N}_{:,S}\widehat{\boldsymbol{Z}}\boldsymbol{N}_{:,S}^\top$
(3) invert the middle term.

This cost is dominated by the first term, which can be viewed as multiplying $\lceil n/k \rceil$ pairs of $k \times n$ and $n \times k$ matrices. Each such multiplication takes time $k^{\omega-1}n$, for a total cost of $O(k^{\omega-2}n^2)$. The other terms all involve matrices with dimension at most $k \times n$, and are thus lower order terms. □

5.1.2 Approximation and Fast Linear Systems Solver. We now define the notion of approximation we use and how to solve linear systems fast given a good preconditioner.

Definition 5.3. *We use $a \approx_c b$ for positive numbers a and b iff $c^{-1}a \leq b \leq c \cdot b$, and for vectors and for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ we use $\boldsymbol{a} \approx_c \boldsymbol{b}$ to denote $\boldsymbol{a}_i \approx_c \boldsymbol{b}_i$ entry-wise.*

In our algorithm, we only update k resistances that have increased by a constant factor. We can therefore use a constant factor preconditioner to solve the new linear system. We will use the following result on solving preconditioned systems of linear equations.

Lemma 5.4. *If $\boldsymbol{r}$ and $\tilde{\boldsymbol{r}}$ are vectors such that $\boldsymbol{r} \approx_{\widetilde{O}(1)} \tilde{\boldsymbol{r}}$, and we're given the matrix $\widehat{Z}^{-1} = (\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\boldsymbol{r})\boldsymbol{N})^{-1}$ explicitly, then we can solve a system of linear equations involving $Z = \boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\tilde{\boldsymbol{r}})\boldsymbol{N}$ to $1/poly(n)$ accuracy in $\widetilde{O}(n^2)$ time.*

Proof. Suppose we want to solve the system,

$$Zx = \boldsymbol{b}.$$

We know $\widehat{Z}^{-1}$ and that for some constant c, $\frac{1}{c}\boldsymbol{I} \preceq \widehat{Z}^{-1/2} Z \widehat{Z}^{-1/2} \preceq c\boldsymbol{I}$. The following iterative method (which is essentially gradient descent),

$$\boldsymbol{x}^{(k+1)} \to \boldsymbol{x}^{(k)} - \hat{Z}^{-1}(Zx - \boldsymbol{b})$$

converges to an ε-approximate solution in $O\left(c \log \frac{1}{\varepsilon}\right)$ iterations. Each iteration can be computed via matrix-vector products. Since matrix vector products for $n \times n$ matrices require at most $O(n^2)$ we get the above lemma for $\varepsilon = 1/poly(n)$. □

5.1.3 Algorithm. The algorithm is the same as that in Section 6 of Adil et al. [3]. The algorithm has two parts, an initialization routine InverseInit which is called only at the first iteration, and the inverse maintenance procedure, UpdateInverse which is called from Algorithm 4, Oracle. Algorithm Oracle is called every time the resistances are updated in Algorithm 5. For this section, we will assume access to all variables from these routines, and maintain the following global variables:

(1) $\widehat{\boldsymbol{r}}$: resistances from the last time we updated each entry.
(2) $counter(\eta)_e$: for each entry, track the number of times that it changed (relative to $\widehat{\boldsymbol{r}}$) by a factor of about $2^{-\eta}$ since the previous update.
(3) $\widehat{Z}$, an inverse of the matrix given by $\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\widehat{\boldsymbol{r}})\boldsymbol{N}$.

Algorithm 8 Inverse Maintenance Initialization

```
procedure InverseInit($\boldsymbol{M}, \boldsymbol{N}, \boldsymbol{r}^{(0)}$)
    Set $\widehat{\boldsymbol{r}} \leftarrow \boldsymbol{r}^{(0)}$.
    Set $counter(\eta)_e \leftarrow 0$ for all $0 \leq \eta \leq \log(m)$ and $e$.
    Set $\widehat{\boldsymbol{Z}} \leftarrow (\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\boldsymbol{r})\boldsymbol{N})^{-1}$ by explicitly inverting the matrix.
```

Algorithm 9 Inverse Maintenance Procedure

```
procedure UpdateInverse
    for all entries $e$ do
        Find the least non-negative integer $\eta$ such that
            $$\frac{1}{2^\eta} \leq \frac{\boldsymbol{r}_e^{(i)} - \boldsymbol{r}_e^{(i-1)}}{\widehat{\boldsymbol{r}}_e}.$$
        Increment $counter(\eta)_e$.
    $E_{changed} \leftarrow \cup_{\eta : i \pmod{2^\eta} \equiv 0} \{e : counter(\eta)_e \geq 2^\eta\}$
    $\tilde{\boldsymbol{r}} \leftarrow \widehat{\boldsymbol{r}}$
    for all $e \in E_{changed}$ do
        $\tilde{\boldsymbol{r}}_e \leftarrow \boldsymbol{r}_e^{(i)}$.
        Set $counter(\eta)_e \leftarrow 0$ for all $\eta$.
    $\widehat{\boldsymbol{Z}} \leftarrow$ LowRankUpdate($\widehat{\boldsymbol{Z}}, \widehat{\boldsymbol{r}}, \tilde{\boldsymbol{r}}$).
    $\widehat{\boldsymbol{r}} \leftarrow \tilde{\boldsymbol{r}}$.
```

5.1.4 Analysis. We first verify that the maintained inverse is always a good preconditioner to the actual matrix, $\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top Diag(\boldsymbol{r}^{(i)})\boldsymbol{N}$.

Lemma 5.5 (Lemma 6.5, Adil et al. [3]). *After each call to* UpdateInverse, *the vector* $\widehat{\boldsymbol{r}}$ *satisfies*

$$\widehat{\boldsymbol{r}} \approx_{\widetilde{O}(1)} \boldsymbol{r}^{(i)}.$$

Proof. First, observe that any change in resistance exceeding 1 is reflected immediately. Otherwise, every time we update $counter(j)_e$, $\boldsymbol{r}_e$ can only increase additively by at most

$$2^{-j+1}\widehat{\boldsymbol{r}}_e.$$

Once $counter(j)_e$ exceeds 2^j, e will be added to $E_{changed}$ after at most 2^j steps. So when we start from $\widehat{\boldsymbol{r}}_e$, e is added to $E_{changed}$ after $counter(j)_e \leq 2^j + 2^j = 2^{j+1}$ iterations. The maximum possible increase in resistance due to the bucket j is,

$$2^{-j+1}\widehat{\boldsymbol{r}}_e \cdot 2^{j+1} = 4\widehat{\boldsymbol{r}}_e.$$

Since there are only at most $m^{1/3}$ iterations, the contributions of buckets with $j > \log m$ are negligible. Now the change in resistance is influenced by all buckets j, each contributing at most $4\widehat{\boldsymbol{r}}_e$ increase. The total change is at most $4\widehat{\boldsymbol{r}}_e \log m$ since there are at most $\log m$ buckets. We therefore have

$$\widehat{\boldsymbol{r}}_e \leq \boldsymbol{r}_e^{(i)} \leq 5\widehat{\boldsymbol{r}}_e \log m.$$

for every i. □

It remains to bound the number and sizes of calls made to Lemma 5.2. For this we define variables $k(\eta)^{(i)}$ to denote the number of edges added to $E_{changed}$ at iteration i due to the value of $counter(\eta)_e$. Note that $k(\eta)^{(i)}$ is non-zero only if $i \equiv 0 \pmod{2^\eta}$, and

$$\left|E_{changed}^{(i)}\right| \leq \sum_{\eta} k(\eta)^{(i)}.$$

We divide our analysis into 2 cases, when the relative change in resistance is at least 1 and when the relative change in resistance is at most 1. To begin with, let us first look at the following lemma that relates the change in weights to the relative change in resistance.

Lemma 5.6. *Consider a primal step from Algorithm 5. We have*

$$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \leq \left(1 + \alpha \frac{|\boldsymbol{N}\Delta|_e}{\zeta^{1/p}}\right)^{p-2} - 1$$

where Δ is the solution produced by the oracle Algorithm 4.

Proof. Recall from Algorithm 4 that

$$\boldsymbol{r}_e^{(i)} = \left(\boldsymbol{w}_e^{(i)}\right)^{p-2}.$$

For a primal step of Algorithm 5, we have

$$\boldsymbol{w}_e^{(i+1)} - \boldsymbol{w}_e^{(i)} = \frac{\alpha}{\zeta^{1/p}} |\boldsymbol{N}\Delta|_e.$$

Substituting this in gives

$$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} = \frac{\left(\boldsymbol{w}_e^{(i)} + \frac{\alpha}{\zeta^{1/p}}|\boldsymbol{N}\Delta|_e\right)^{p-2} - \left(\boldsymbol{w}_e^{(i)}\right)^{p-2}}{\left(\boldsymbol{w}_e^{(i)}\right)^{p-2}} \leq \left(1 + \frac{\frac{\alpha}{\zeta^{1/p}}|\boldsymbol{N}\Delta|_e}{\boldsymbol{w}_e^{(i)}}\right)^{p-2} - 1 \leq \left(1 + \frac{\alpha}{\zeta^{1/p}}|\boldsymbol{N}\Delta|_e\right)^{p-2} - 1,$$

where the last inequality utilizes $\boldsymbol{w}_e^{(i)} \geq 1$. □

We now consider the case when the relative change in resistance is at least 1.

Lemma 5.7. *Throughout the course of a run of Algorithm 5, the number of edges added to $E_{changed}$ due to relative resistance increase of at least 1,*

$$\sum_{1 \leq i \leq T} k(0)^{(i)} \leq O\left(m^{\frac{p+2}{3p-2}}\right).$$

Proof. From Lemma C.1, we know that the change in energy over one iteration is at least,

$$\sum_e (\boldsymbol{N}\Delta)_e^2 \left(1 - \frac{\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i+1)}}\right).$$

Over all iterations, the change in energy is at least,

$$\sum_i \sum_e (\boldsymbol{N}\Delta)_e^2 \left(1 - \frac{\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i+1)}}\right)$$

which is upper bounded by $O(m^{\frac{p-2}{p}})\zeta^{2/p}$. When iteration i is a width reduction step, the relative resistance change is always at least 1. In this case $|\boldsymbol{N}\Delta| \geq \rho\zeta^{1/p}$. When we have a primal step, Lemma 5.6 implies that when the relative change in resistance is at least 1 then,

$$|\boldsymbol{N}\Delta|_e \geq \Omega(1)\alpha^{-1}\zeta^{1/p}.$$

Using the bound $|\boldsymbol{N}\Delta|_e \geq \Omega(p^{-1})\alpha^{-1}\zeta^{1/p}$ is sufficient since $\rho > \Omega(p^{-1}\alpha^{-1})$ and both kinds of iterations are accounted for. The total change in energy can now be bounded.

$$
\begin{aligned}
& p^{-2}\alpha^{-2}\zeta^{2/p}\sum_i\sum_e \mathbb{1}_{\left[\frac{r_e^{(i+1)}-r_e^{(i)}}{r_e^{(i)}}\geq 1\right]} \leq O(m^{\frac{p-2}{p}})\zeta^{2/p}\\
& \Leftrightarrow p^{-2}\alpha^{-2}\sum_i k(0)^{(i)} \leq O(m^{\frac{p-2}{p}})\\
& \Leftrightarrow \sum_i k(0)^{(i)} \leq O(p^2 m^{(p-2)/p}\alpha^2).
\end{aligned}
$$

The Lemma follows by substituting $\alpha = \Theta\left(p^{-1}m^{-\frac{p^2-5p+2}{p(3p-2)}}\right)$ in the above equation. □

Lemma 5.8. *Throughout the course of a run of Algorithm 5, the number of edges added to $E_{changed}$ due to relative resistance increase between $2^{-\eta}$ and $2^{-\eta+1}$,*

$$
\sum_{1\leq i\leq T} k(\eta)^{(i)} \leq \begin{cases} 0 & \text{if } 2^{\eta} \geq T,\\ O\left(m^{\frac{p+2}{3p-2}}2^{2\eta}\right) & \text{otherwise.}\end{cases}
$$

Proof. From Lemma C.1, the total change in energy is at least,

$$
\sum_i\sum_e (\boldsymbol{N}\Delta)_e^2\left(1-\frac{\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i+1)}}\right).
$$

We know that $\frac{r_e^{(i+1)}-r_e^{(i)}}{r_e^{(i)}} \geq 2^{-\eta}$. Using Lemma 5.6, we have,

$$
\left(1+\alpha\frac{|\boldsymbol{N}\Delta|_e}{\zeta^{1/p}}\right)^{p-2} - 1 \geq 2^{-\eta}.
$$

We thus obtain,

$$
\left(1+\alpha\frac{|\boldsymbol{N}\Delta|_e}{\zeta^{1/p}}\right)^{p-2} - 1 \leq \begin{cases} \alpha\frac{|\boldsymbol{N}\Delta|_e}{\zeta^{1/p}} & \text{when } \alpha|\Delta_e| \leq \zeta^{1/p} \text{ or } p-2\leq 1\\ \left(2\alpha\frac{|\boldsymbol{N}\Delta|_e}{\zeta^{1/p}}\right)^{p-2} & \text{otherwise.}\end{cases}
$$

Now, in the second case, when $\alpha|\boldsymbol{N}\Delta_e| \geq \zeta^{1/p}$ and $p-2>1$,

$$
\left(2\alpha\frac{|\boldsymbol{N}\Delta|_e}{\zeta^{1/p}}\right)^{p-2} \geq 2^{-\eta} \Rightarrow \alpha|\boldsymbol{N}\Delta|_e \geq \left(\frac{1}{2^{\eta}}\right)^{1/(p-2)+1}\zeta^{1/p} \geq 2^{-\eta-1}\zeta^{1/p}
$$

Therefore, for both cases we have,

$$
\alpha|\boldsymbol{N}\Delta|_e \geq \left(2^{-\eta-1}\right)\zeta^{1/p}.
$$

Using the above bound and the fact that the total change in energy is at most $O(m^{\frac{p-2}{p}})\zeta^{2/p}$, gives,

$$
\begin{aligned}
& \sum_i\sum_e (\boldsymbol{N}\Delta)_e^2\left(1-\frac{\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i+1)}}\right) \leq O(m^{\frac{p-2}{p}})\zeta^{2/p}\\
& \Rightarrow \frac{1}{4}\sum_i\sum_e\left(\alpha^{-1}2^{-\eta}\zeta^{1/p}\right)^2\cdot\left(2^{-\eta}\mathbb{1}_{2^{-\eta+1}\geq\frac{r_e^{(i+1)}-r_e^{(i)}}{r_e^{(i)}}\geq 2^{-\eta}}\right) \leq O(m^{\frac{p-2}{p}})\zeta^{2/p}\\
& \Rightarrow \alpha^{-2}2^{-3\eta}\sum_i 2^{\eta}k(\eta)^{(i)} \leq O(m^{\frac{p-2}{p}})
\end{aligned}
$$

$$\Rightarrow \sum_i k(\eta)^{(i)} \leq O\left(m^{(p-2)/p}\alpha^2 2^{2\eta}\right)$$

The Lemma follows substituting $\alpha = \Theta\left(p^{-1}m^{-\frac{p^2-5p+2}{p(3p-2)}}\right)$ in the above equation. □

We can now use the concavity of $f(z) = z^{\omega-2}$ to upper bound the contribution of these terms.

Corollary 5.9. *Let $k(\eta)^{(i)}$ be as defined. Over all iterations we have,*

$$\sum_i \left(k(0)^{(i)}\right)^{\omega-2} \leq O\left(p^{3-\omega}m^{\frac{p-(10-4\omega)}{3p-2}}\right)$$

and for every η,

$$\sum_i^T \left(k(\eta)^{(i)}\right)^{\omega-2} \leq \begin{cases} 0 & \text{if } 2^\eta \geq T, \\ O\left(p^{3-\omega}m^{\frac{p-2+4(\omega-2)}{3p-2}} \cdot 2^{\eta(3\omega-7)}\right) & \text{otherwise.} \end{cases}$$

Proof. Due to the concavity of the $\omega - 2 \approx 0.3727 < 1$ power, this total is maximized when it's equally distributed over all iterations. In the first sum, the number of terms is equal to the number of iterations, i.e., $O(pm^{\frac{p-2}{3p-2}})$. In the second sum the number of terms is $O(pm^{\frac{p-2}{3p-2}})2^{-\eta}$. Distributing the sum equally over the above numbers give,

$$\sum_i^T \left(k(0)^{(i)}\right)^{\omega-2} \leq \left(O\left(p^{-1}m^{\frac{p+2}{3p-2}-\frac{p-2}{3p-2}}\right)\right)^{\omega-2} \cdot O\left(pm^{\frac{p-2}{3p-2}}\right) = O\left(p^{3-\omega}m^{\frac{p-2+4(\omega-2)}{3p-2}}\right) = O\left(p^{3-\omega}m^{\frac{p-(10-4\omega)}{3p-2}}\right)$$

and

$$\begin{aligned} \sum_i^T \left(k(\eta)^{(i)}\right)^{\omega-2} &\leq O\left(pm^{\frac{p-2}{3p-2}}2^{-\eta}\right) \cdot \left(p^{-1}\frac{m^{\frac{p+2}{3p-2}}2^{2\eta}}{m^{\frac{p-2}{3p-2}}2^{-\eta}}\right)^{\omega-2} \\ &= O\left(p^{3-\omega}m^{\frac{p-2+4(\omega-2)}{3p-2}}2^{-\eta} \cdot 2^{3\eta(\omega-2)}\right) \\ &= O\left(p^{3-\omega}m^{\frac{p-2+4(\omega-2)}{3p-2}}2^{\eta(3\omega-7)}\right). \end{aligned}$$

□

5.2 Proof of Theorem 5.1

Theorem 5.1. *If $\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}$ are explicitly given, matrices with polynomially bounded condition numbers, and $p \geq 2$, Algorithm 5 as given in Section 3.1.2 can be implemented to run in total time*

$$O\left(mn^{\omega-1} + p^{3-\omega}n^2m^{\omega-2} + p^{3-\omega}n^2m^{\frac{p-(10-4\omega)}{3p-2}}\right).$$

Proof. By Lemma 5.5, the $\widehat{\boldsymbol{r}}$ that the inverse being maintained corresponds to always satisfy $\widehat{\boldsymbol{r}} \approx_{\widetilde{O}(1)} \boldsymbol{r}^{(i)}$. So by the iterative linear systems solver method outlined in Lemma 5.4, we can implement each call to Oracle (Algorithm 4)in time $O(n^2)$ in addition to the cost of performing inverse maintenance. This leads to a total cost of

$$\widetilde{O}\left(pn^2m^{\frac{p-2}{3p-2}}\right).$$

across the $T = \Theta(pm^{\frac{p-2}{3p-2}})$ iterations.

The costs of inverse maintenance is dominated by the calls to the low-rank update procedure outlined in Lemma 5.2. Its total cost is bounded by

$$O\left(\sum_i \left|E^{(i)}_{changed}\right|^{\omega-2} n^2\right) = O\left(n^2 \sum_i \left(\sum_\eta k(\eta)^{(i)}\right)^{\omega-2}\right).$$

Because there are only $O(\log m)$ values of η, and each $k(\eta)^{(i)}$ is non-negative, we can bound the total cost by:

$$\widetilde{O}\left(n^2 \sum_i \sum_\eta \left(k(\eta)^{(i)}\right)^{\omega-2}\right) \leq \widetilde{O}\left(p^{3-\omega} n^2 \sum_{\eta:2^\eta \leq T} m^{\frac{p-2+4(\omega-2)}{3p-2}} \cdot 2^{\eta(3\omega-7)}\right),$$

where the inequality follows from substituting in the result of Lemma 5.9. Depending on the sign of $3\omega - 7$, this sum is dominated either at $\eta = 0$ or $\eta = \log T$. Including both terms then gives

$$\widetilde{O}\left(p^{3-\omega} n^2 \left(m^{\frac{p-2+4(\omega-2)}{3p-2}} + m^{\frac{p-2+4(\omega-2)+(p-2)(3\omega-7)}{3p-2}}\right)\right),$$

with the exponent on the trailing term simplifying to $\omega - 2$ to give,

$$\widetilde{O}\left(p^{3-\omega} n^2 \left(m^{\frac{p-(10-4\omega)}{3p-2}} + m^{\omega-2}\right)\right).$$

□

6 ITERATIVELY REWEIGHTED LEAST SQUARES ALGORITHM

Iteratively Reweighted Least Squares (IRLS) Algorithms are a family of algorithms for solving ℓ_p-regression. These algorithms have been studied extensively for about 60 years [24, 33, 39] and the classical form solves the following version of ℓ_p-regression,

$$\min_{x} \|Ax - b\|_p, \tag{12}$$

where A is a tall thin matrix and b is a vector. The main idea in IRLS algorithms is to solve a weighted least squares problem in every iteration to obtain the next iterate,

$$x^{(t+1)} = \arg\min_{x} (Ax^{(t)} - b)^\top R (Ax^{(t)} - b) \tag{13}$$

starting from $x^{(0)}$ which is usually $\arg\min_x \|Ax - b\|_2^2$. Here R is picked to be $Diag(|Ax^{(t)} - b|^{p-2})$ and note that the above equation now becomes a fixed point iterate for the ℓ_p-regression problem. It is known that the fixed point is unique for $p \in (1, \infty)$.

The basic version of the above IRLS algorithm is guaranteed to converge for $p \in (1.5, 3)$, however, even for small $p \approx 3.5$, the algorithm diverges [41]. Over the years there have been several studies on IRLS algorithms and attempts to show convergence [29, 38], but none of them show quantitative bounds or require starting solutions close enough to the optimum. Refer to Burrus [12] for a complete survey on these methods.

In this section we propose an IRLS algorithm and prove that our algorithm is guaranteed to converge geometrically to the optimum. Our algorithm is based on the algorithm of Adil et al. [4] and present some experimental results from experiments performed in the paper that demonstrate our algorithm works very well in practice. We provide a much simpler analysis and integrate the analysis with the framework we have built so far.

We will focus on the following pure ℓ_p setting for better readability,

$$\min_{Ax=b} \|Nx\|_p.$$

We note that our algorithm also works for the setting described in Equation (12). We will first describe our algorithm in the next section, and then present some experimental results from experiments that were performed in Adil et al. [4].

6.1 IRLS Algorithm

Our IRLS algorithm is based on our overall iterative refinement framework (Algorithm 1) where we will directly use a weighted least squares problem to solve the residual problem. Consider Algorithm 10 and compare it with Algorithm 1. We note that it is same overall, except now we have an extra step LineSearch and we update the solution (Line 7) at every iteration. These steps do not affect the overall convergence guarantees of the iterative refinement framework in Algorithm 1, since these are only ensuring that given a solution from ResidualSolver-IRLS, we are taking a step that reduces the objective value the most as opposed the fixed update defined in Algorithm 1. In other words, we are reducing the objective value in each iteration at least as much as in Algorithm 1. We thus require to prove the guarantees of ResidualSolver-IRLS (Algorithm 11) and combine it with Theorem 2.1 to obtain our final convergence guarantees. We will prove the following result on our IRLS algorithm (Algorithm 10).

Theorem 1.3. *Let $p \geq 2$. Algorithm 10 returns $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\|\boldsymbol{N}\boldsymbol{x}\|_p^p \leq (1+\varepsilon)\|\boldsymbol{N}\boldsymbol{x}^\star\|_p^p$, in at most $O\left(p^3 m^{\frac{(p-2)}{2(p-1)}} \log\left(\frac{m}{\varepsilon}\right)\right)$ calls to a linear system solver.*

The key connection with IRLS algorithms is that we are able to show that it is sufficient to solve a weighted least squares problem to solve the residual problem. The two main differences are, in every iteration we add a small systematic *padding* to $\boldsymbol{R}$ and, we perform a line search. These tricks are common empirical modifications used to avoid ill conditioning of matrices and for a faster convergence [29, 46].

Algorithm 10 Iteratively Reweighted Least Squares

```
 1: procedure IRLS($A, N, b, p, \varepsilon$)
 2:     $x \leftarrow \arg\min_{Ax=b} \|Nx\|_2^2$
 3:     $\nu \leftarrow \|Nx\|_p^p$
 4:     while $\nu > \frac{\varepsilon}{2}\|Nx\|_p^p$ do
 5:         $\widetilde{\Delta}, \kappa \leftarrow$ ResidualSolver-IRLS($x, N, A, b, \nu, p$)
 6:         $\alpha \leftarrow$ LineSearch($N, x, \widetilde{\Delta}$)                ▷ $\alpha = \arg\min_\beta \|N(x - \beta\widetilde{\Delta})\|_p^p$
 7:         $x \leftarrow x - \alpha\frac{\widetilde{\Delta}}{p}$
 8:         if $res_p(\alpha\widetilde{\Delta}) < \frac{\nu}{32p\kappa}$ then
 9:             $\nu \leftarrow \frac{\nu}{2}$
10:     return $x$
```

We will prove the following result about solving the residual problem.

Lemma 6.1. *Let $\boldsymbol{x}$ be the current iterate and ν be such that $\|\boldsymbol{N}\boldsymbol{x}\|_p^p - OPT \in (\nu/2, \nu]$. Let $\widetilde{\Delta}$ be the solution of* (14). *Then for α_0 and α as defined in Algorithm 11 and Algorithm 10 respectively, $\alpha\widetilde{\Delta}$ is a $O(p^2\alpha_0^{-1}) = O\left(p^2 m^{\frac{p-2}{2(p-1)}}\right)$-approximate solution to the residual problem.*

We note that Theorem 1.3 directly follows from Lemma 6.1, Lemma 2.7 and Theorem 2.1. Therefore, in the next section, we will prove Lemma 6.1.

Algorithm 11 Residual Solver for IRLS

```
procedure ResidualSolver-IRLS($\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{A}, \boldsymbol{b}, \nu, p$)
    $\boldsymbol{g} \leftarrow Diag(|\boldsymbol{Nx}|^{p-2})\boldsymbol{Nx}$
    $\boldsymbol{R} \leftarrow 2Diag(|\boldsymbol{Nx}|^{p-2})$
    $\boldsymbol{s} \leftarrow \nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}}$
    $\widetilde{\Delta} \leftarrow \arg\max_{\boldsymbol{A}\Delta=0} \boldsymbol{g}^\top \boldsymbol{N}\Delta - \Delta^\top \boldsymbol{N}^\top (R + \boldsymbol{sI})\boldsymbol{N}\Delta$        ▷ Problem (14)
    $k \leftarrow \frac{\|\boldsymbol{N}\widetilde{\Delta}\|_p^p}{\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R}+s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}}$
    $\alpha_0 \leftarrow \min\left\{\frac{1}{2}, \frac{1}{2k^{1/(p-1)}}\right\}$
    return $\widetilde{\Delta}, 2^{13}p^2\alpha_0^{-1}$
```

6.1.1 Solving the Residual Problem. Recall the residual problem (Definition 2.3),

$$\max_{\boldsymbol{A}\Delta=0} \boldsymbol{g}^\top \boldsymbol{N}\Delta - \Delta^\top \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\Delta - \|\boldsymbol{N}\Delta\|_p^p,$$

with $\boldsymbol{g} = Diag(|\boldsymbol{Nx}|^{p-2})\boldsymbol{Nx}$ and $\boldsymbol{R} = 2Diag(|\boldsymbol{Nx}|^{p-2})$. Let ν be as in Algorithm 1, then we will show that the solution of the following weighted least squares problem is a good approximation to the residual problem,

$$\max_{\boldsymbol{A}\Delta=0} \boldsymbol{g}^\top \boldsymbol{N}\Delta - \Delta^\top \boldsymbol{N}^\top (\boldsymbol{R} + \nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} \boldsymbol{I})\boldsymbol{N}\Delta. \tag{14}$$

6.1.2 Proof of Lemma 6.1.

Proof. Since $\|\boldsymbol{Nx}\|_p^p - OPT \in (\nu/2, \nu]$, from Lemma 2.6, we have the optimum of the residual problem satisfies, $res_p(\Delta^\star) \in (\nu/32p, \nu]$. We will next prove, that the objective of (14) at the optimum is at most ν and at least $\frac{\nu}{2^{13}p^2}$. Before proving the above bound, we will prove how $\alpha\widetilde{\Delta}$ gives the required approximation to the residual problem. We have $\alpha_0 = \min\left\{\frac{1}{2}, \frac{1}{(2k)^{1/p-1}}\right\}$.

$$\begin{aligned}
res_p(\alpha\widetilde{\Delta}) &\geq 16p \cdot res_p(\alpha\widetilde{\Delta}/16p)\\
&\geq \|\boldsymbol{Nx}\|_p^p - \|\boldsymbol{N}(\boldsymbol{x} - \alpha\widetilde{\Delta})\|_p^p\\
&\geq \|\boldsymbol{Nx}\|_p^p - \|\boldsymbol{N}(\boldsymbol{x} - \alpha_0\widetilde{\Delta})\|_p^p\\
&\geq res_p(\alpha_0\widetilde{\Delta})\\
&= \alpha_0\left(\boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta} - \alpha_0\widetilde{\Delta}^\top \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\widetilde{\Delta} - \alpha_0^{p-1}\|\boldsymbol{N}\widetilde{\Delta}\|_p^p\right)\\
&\geq \alpha_0\left(\boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta} - \alpha_0\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta} - \alpha_0^{p-1}k\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}\right)\\
&\geq \alpha_0\left(\boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta} - \frac{1}{2}\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta} - \frac{1}{2}\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}\right)\\
&= \alpha_0\left(\boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta} - \widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}\right)\\
&\geq \frac{\alpha_0\nu}{2^{13}p^2} \geq \frac{\alpha_0}{2^{13}p^2}OPT.
\end{aligned}$$

It remains to prove the bound on the optimal objective of (14) and bound α_0 for which it is sufficient to find an upper bound on k,

$$k = \frac{\|\boldsymbol{N}\widetilde{\Delta}\|_p^p}{\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}}.$$

We will first bound k. Since, $s\boldsymbol{I} \leq \boldsymbol{R} + s\boldsymbol{I}$,

$$\|\boldsymbol{N}\widetilde{\Delta}\|_2^2 \leq \frac{1}{s}\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta},$$

and

$$\|\boldsymbol{N}\widetilde{\Delta}\|_p^p \leq \|\boldsymbol{N}\widetilde{\Delta}\|_2^p \leq \frac{1}{s}\|\boldsymbol{N}\widetilde{\Delta}\|_2^{p-2}\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}.$$

Therefore it is sufficient to bound $\|\boldsymbol{N}\widetilde{\Delta}\|_2$, as

$$k = \frac{\|\boldsymbol{N}\widetilde{\Delta}\|_p^p}{\widetilde{\Delta}^\top \boldsymbol{N}^\top (\boldsymbol{R} + s\boldsymbol{I})\boldsymbol{N}\widetilde{\Delta}} \leq \frac{1}{s}\|\boldsymbol{N}\widetilde{\Delta}\|_2^{p-2}.$$

To bound $\|\boldsymbol{N}\widetilde{\Delta}\|_2$, we start by assuming $|\boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta}| \leq \nu$. Now, since optimal objective of (14) is lower bounded by $\frac{\nu}{2^{13}p^2}$,

$$\widetilde{\Delta}^\top \boldsymbol{N}^\top (R + \nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} \boldsymbol{I})\boldsymbol{N}\widetilde{\Delta} \leq \boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta} - \frac{\nu}{2^{13}p^2} \leq \nu.$$

We thus have,

$$\nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} \|\boldsymbol{N}\widetilde{\Delta}\|_2^2 \leq \nu.$$

Using this we get,

$$k \leq \frac{1}{\nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}}} \frac{\nu^{\frac{p-2}{2}}}{\nu^{\frac{(p-2)^2}{2p}} m^{-\frac{(p-2)^2}{2p}}} = m^{\frac{p-2}{2}}.$$

We thus have α_0 lower bounded by $m^{-\frac{p-2}{2(p-1)}}$, which gives us our result. It remains to give a lower bound to the optimal objective of (14).

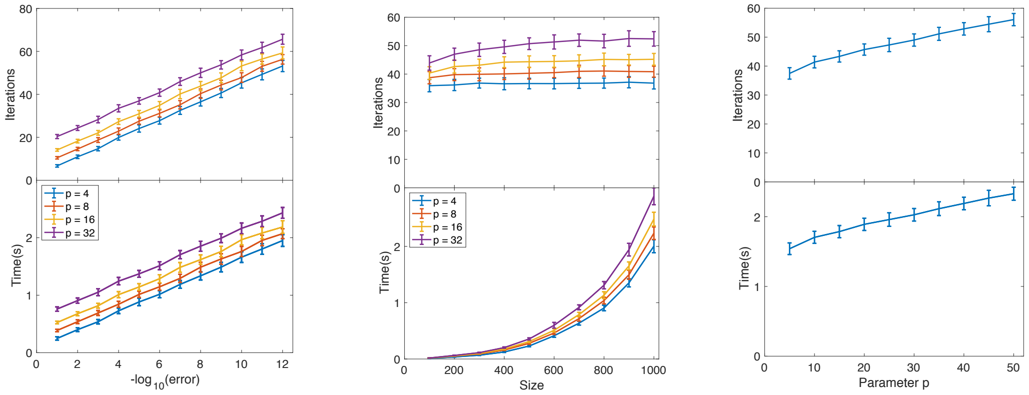
Let $\Delta^\star$ denote the optimum of the residual problem. We know that $\|\boldsymbol{N}\Delta^\star\|_p^p \leq \nu$, $\Delta^{\star\top} \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\Delta^\star \leq \nu$ and $\boldsymbol{g}^\top \boldsymbol{N}\Delta^\star > \nu/32p$. Since $\|\boldsymbol{N}\Delta^\star\|_p^p \leq \nu$ we have $\|\boldsymbol{N}\Delta^\star\|_2^2 \leq m^{(p-2)/p}\nu^{2/p}$. For $a = 1/2^7 p$, $a\Delta^\star$ is a feasible solution for (14).

$$\begin{aligned}
\boldsymbol{g}^\top \boldsymbol{N}\widetilde{\Delta} - \widetilde{\Delta}^\top \boldsymbol{N}^\top (R + \nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} \boldsymbol{I})\boldsymbol{N}\widetilde{\Delta} &\geq a\boldsymbol{g}^\top \boldsymbol{N}\Delta^\star - a^2 \Delta^{\star\top} \boldsymbol{N}^\top (R + \nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} \boldsymbol{I})\boldsymbol{N}\Delta^\star \\
&\geq a\left(\frac{\nu}{32p} - a\Delta^{\star\top} \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\Delta^\star - a\nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} \|\boldsymbol{N}\Delta^\star\|_2^2\right) \\
&\geq a\left(\frac{\nu}{32p} - a\nu - a\nu^{\frac{p-2}{p}} m^{-\frac{p-2}{p}} m^{(p-2)/p}\nu^{2/p}\right) \\
&= a\left(\frac{\nu}{32p} - a\nu - a\nu\right) \\
&= a\frac{\nu}{2^6 p} = \frac{\nu}{2^{13}p^2}
\end{aligned}$$

Thus, the optimal objective of (14) is lower bounded by $\frac{\nu}{2^{13}p^2}$. □

6.2 Experiments

In this section, we include the experimental results from Adil et al. [4] which are based on Algorithm p-IRLS described in the paper. We would like to mention that p-IRLS is similar in spirit to Algorithm 10 and thus we expect a similar performance by an implementation of Algorithm 10. Algorithm p-IRLS is described for setting (12) and is available at https://github.com/fast-algos/pIRLS [5]. We now give a brief summary of the experiments.



(a) Size of $\boldsymbol{A}$ fixed to 1000×850.

(b) Sizes of $\boldsymbol{A}$: $(50 + 100(k-1)) \times 100k$. Error $\varepsilon = 10^{-8}$.

(c) Size of $\boldsymbol{A}$ is fixed to 1000×850. Error $\varepsilon = 10^{-8}$.

Fig. 1. Random Matrix instances. Comparing the number of iterations and time taken by our algorithm with the parameters. Averaged over 100 random samples for $\boldsymbol{A}$ and $\boldsymbol{b}$. Linear solver used : backslash.

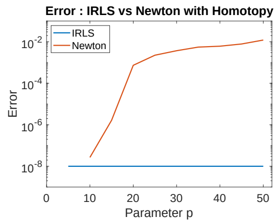


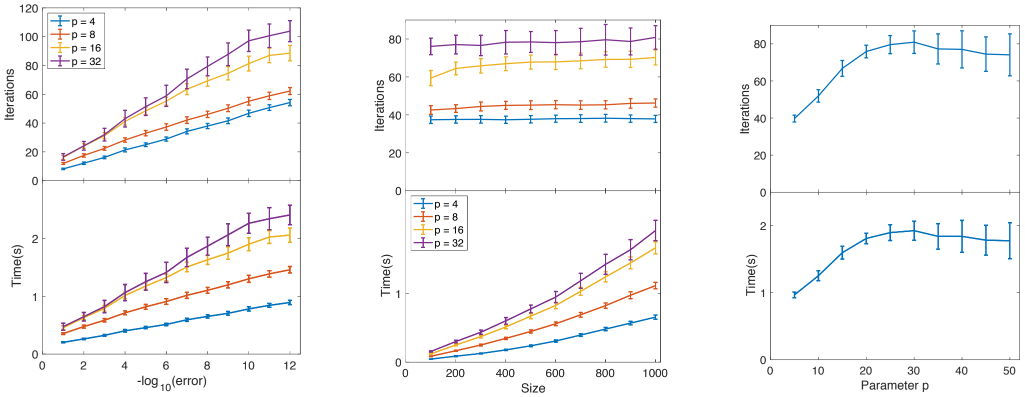
Fig. 2. Averaged over 100 random samples. Graph: 1000 nodes (5000-6000 edges). Solver: PCG with Cholesky preconditioner.

6.2.1 Experiments on p-IRLS. All implementations were done on on MATLAB 2018b on a Desktop ubuntu machine with an Intel Core i5-4570 CPU @ $3.20GHz \times 4$ processor and 4GB RAM. The two kinds of instances considered are *Random Matrices* and *Graph instances* for the problem $\min_x \|\boldsymbol{Ax} - \boldsymbol{b}\|_p$.

(1) **Random Matrices:** Matrices $\boldsymbol{A}$ and $\boldsymbol{b}$ are generated randomly i.e., every entry of the matrix is chosen uniformly at random between 0 and 1.

(2) **Graphs:** Instances are generated as in Rios et al. [41]. Vertices are uniform random vectors in $[0,1]^{10}$ and edges are created by connecting the 10 nearest neighbors. The weight of every edge is determined by a Gaussian function (Eq 3.1,[41]). Around 10 vertices have labels chosen uniformly at random between 0 and 1. The problem is to minimize the ℓ_p laplacian. Appendix B contains details on how to formulate this problem into our standard form. These instances were generated using the code by Rios [40].

The performance of p-IRLS is compared against Matlab/CVX solver [25, 26] and the IRLS/homotopy based implementation from Rios et al. [41]. More details on the experiments are in Adil et al. [4] and the plots and specific details of the implementation are included in Figures 1,2,3 and, 4.

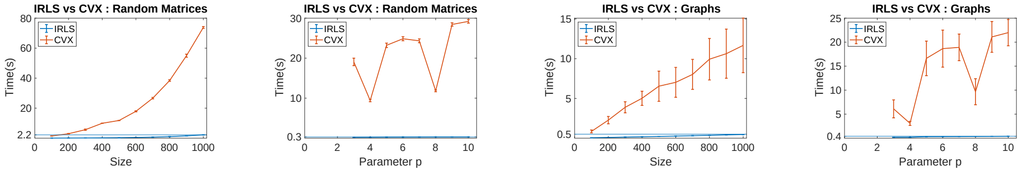


(a) Size of graph fixed to 1000 nodes (around 5000-6000 edges).

(b) Number of nodes: $100k$. Error $\varepsilon = 10^{-8}$.

(c) Size of graph fixed to 1000 nodes (around 5000-6000 edges). Error $\varepsilon = 10^{-8}$.

Fig. 3. Graph Instances. Comparing the number of iterations and time taken by our algorithm with the parameters. Averaged over 100 graph samples. Linear solver used : backslash.



(a) Fixed $p = 8$. Size of matrices: $100k \times (50 + 100(k-1))$.

(b) Size of matrices fixed to 500×450.

(c) Fixed $p = 8$. The number of nodes : $50k, k = 1, 2, ..., 10$.

(d) Size of graphs fixed to 400 nodes (around 2000 edges).

Fig. 4. Averaged over 100 samples. Precision set to $\varepsilon = 10^{-8}$.CVX solver used : SDPT3 for Matrices and Sedumi for Graphs.

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 *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[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*. SIAM, 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*. 14189–14200.

[5] Deeksha Adil, Richard Peng, and Sushant Sachdeva. 2019. pIRLS. https://github.com/fast-algos/pIRLS.

[6] Deeksha Adil and Sushant Sachdeva. 2020. Faster p-norm minimizing flows, via smoothed q-norm problems. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 892–910.

[7] Morteza Alamgir and Ulrike Luxburg. 2011. Phase transition in the family of p-resistances. *Advances in neural information processing systems* 24 (2011), 379–387.

[8] Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. 2017. Much faster algorithms for matrix scaling. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 890–901.

[9] Jose Antonio Barreto and C Sidney Burrus. 1994. L/sub p/-complex approximation using iterative reweighted least squares for FIR digital filters. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3. IEEE, III–545.

[10] 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*. 1130–1137.

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

[12] C Burrus. 2012. Iterative re-weighted least-squares OpenStax-CNX.

[13] Jeff Calder. 2019. Consistency of Lipschitz learning with infinite unlabeled data and finite labeled data. *SIAM Journal on Mathematics of Data Science* 1, 4 (2019), 780–812.

[14] Emmanuel J Candes and Terence Tao. 2005. Decoding by linear programming. *IEEE transactions on information theory* 51, 12 (2005), 4203–4215.

[15] 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*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 19052–19063. https://proceedings.neurips.cc/paper/2020/file/dba4c1a117472f6aca95211285d0587e-Paper.pdf

[16] Rick Chartrand and Wotao Yin. 2008. Iteratively reweighted algorithms for compressive sensing. In *2008 IEEE international conference on acoustics, speech and signal processing*. IEEE, 3869–3872.

[17] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. 2022. Maximum flow and minimum-cost flow in almost-linear time. *arXiv preprint arXiv:2203.00671* (2022).

[18] Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P Woodruff. 2017. Algorithms for ℓ_p low-rank approximation. In *International Conference on Machine Learning*. PMLR, 806–814.

[19] Hui Han Chin, Aleksander Madry, Gary L Miller, and Richard Peng. 2013. Runtime guarantees for regression problems. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. 269–282.

[20] Paul Christiano, Jonathan A Kelner, Aleksander Madry, 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 forty-third annual ACM symposium on Theory of computing*. 273–282.

[21] Abderrahim Elmoataz, X Desquesnes, and M Toutain. 2017. On the game p-Laplacian on weighted graphs with applications in image processing and data clustering. *European Journal of Applied Mathematics* 28, 6 (2017), 922–948.

[22] Abderrahim Elmoataz, Matthieu Toutain, and Daniel Tenbrinck. 2015. On the p-Laplacian and ∞-Laplacian on graphs with applications in image and data processing. *SIAM Journal on Imaging Sciences* 8, 4 (2015), 2412–2451.

[23] 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.

[24] Irina F Gorodnitsky and Bhaskar D Rao. 1997. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing* 45, 3 (1997), 600–616.

[25] M. Grant and S. Boyd. 2008. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura (Eds.). Springer-Verlag Limited, 95–110. http://stanford.edu/~boyd/graph_dcp.html.

[26] M. Grant and S. Boyd. 2014. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. http://cvxr.com/cvx.

[27] Yosra Hafiene, Jalal Fadili, and Abderrahim Elmoataz. 2018. Nonlocal p-Laplacian Variational problems on graphs. *arXiv preprint arXiv:1810.12817* (2018).

[28] 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*. 529–542.

[29] LA Karlovitz. 1970. Construction of nearest points in the L_p, p even, and L_∞ norms. I. *Journal of Approximation Theory* 3, 2 (1970), 123–127.

[30] 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.

[31] 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*. 902–913.

[32] R. Kyng, A. B. Rao, S. Sachdeva, and D. A Spielman. 2015. Algorithms for Lipschitz learning on graphs. In *COLT*.

[33] Charles Lawrence Lawson. 1961. Contribution to the theory of linear least maximum approximation. *Ph. D. dissertation, Univ. Calif.* (1961).

[34] Yin Tat Lee and Aaron Sidford. 2014. Path Finding Methods for Linear Programming: Solving Linear Programs in $\tilde{O}(\sqrt{rank})$ Iterations and Faster Algorithms for Maximum Flow. In *Foundations of Computer Science*

(FOCS), 2014 IEEE 55th Annual Symposium on. IEEE, 424–433. Available at http://arxiv.org/abs/1312.6677 and http://arxiv.org/abs/1312.6713.

[35] 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. Available at: https://arxiv.org/abs/1503.01752.

[36] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. 2015. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 1049–1065.

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

[38] Michael Robert Osborne. 1985. *Finite algorithms in optimization and data analysis.* John Wiley & Sons, Inc.

[39] John Rischard Rice. 1964. *The approximation of functions*. Vol. 1. Addison-Wesley Reading, Mass.

[40] M. F. Rios. 2019. Laplacian_Lp_Graph_SSL. https://github.com/mauriciofloresML/Laplacian_Lp_Graph_SSL.

[41] Mauricio Flores Rios, Jeff Calder, and Gilad Lerman. 2019. Algorithms for ℓ_p-based semi-supervised learning on graphs. *arXiv preprint arXiv:1901.05031* (2019).

[42] Damian Straszak and Nisheeth K Vishnoi. 2016. IRLS and slime mold: Equivalence and convergence. *arXiv preprint arXiv:1601.02712* (2016).

[43] Damian Straszak and Nisheeth K Vishnoi. 2016. Natural algorithms for flow problems. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1868–1883.

[44] Damian Straszak and Nisheeth K. Vishnoi. 2016. On a Natural Dynamics for Linear Programming. *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science* (2016).

[45] P Vaidya. 1990. *Solving linear equations with diagonally dominant matrices by constructing good preconditioners.* Technical Report. Technical report, Department of Computer Science, University of Illinois.

[46] Ricardo A Vargas and Charles S Burrus. 1999. Adaptive iterative reweighted least squares design of L/sub p/FIR filters. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, Vol. 3. IEEE, 1129–1132.

A SOLVING ℓ_2 PROBLEMS UNDER SUBSPACE CONSTRAINTS

We will show how to solve general problems of the following form using a linear system solver.

$$\begin{aligned}\min_{\boldsymbol{x}} \quad & \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2\\ & \boldsymbol{C}\boldsymbol{x}=\boldsymbol{d}.\end{aligned}$$

We first write the Lagrangian of the problem,

$$L(\boldsymbol{x},\boldsymbol{v}) = \min_{\boldsymbol{x}}\max_{\boldsymbol{v}} \quad (\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^\top(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})+\boldsymbol{v}^\top(\boldsymbol{d}-\boldsymbol{C}\boldsymbol{x})$$

Using Lagrangian duality and noting that strong duality holds, we can write the above as,

$$\begin{aligned}L(\boldsymbol{x},\boldsymbol{v}) &= \min_{\boldsymbol{x}}\max_{\boldsymbol{v}} \quad (\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^\top(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})+\boldsymbol{v}^\top(\boldsymbol{d}-\boldsymbol{C}\boldsymbol{x})\\ &= \max_{\boldsymbol{v}}\min_{\boldsymbol{x}} \quad (\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^\top(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})+\boldsymbol{v}^\top(\boldsymbol{d}-\boldsymbol{C}\boldsymbol{x}).\end{aligned}$$

We first find $\boldsymbol{x}^\star$ that minimizes the above objective by setting the gradient with respect to $\boldsymbol{x}$ to 0. We thus have,

$$\boldsymbol{x}^\star = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\left(\frac{2\boldsymbol{A}^\top\boldsymbol{b}+\boldsymbol{C}^\top\boldsymbol{v}}{2}\right).$$

Using this value of $\boldsymbol{x}$ we arrive at the following dual program.

$$L(\boldsymbol{v}) = \max_{\boldsymbol{v}} \quad -\frac{1}{4}\boldsymbol{v}^\top\boldsymbol{C}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{C}^\top\boldsymbol{v}-\boldsymbol{b}^\top\boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{b}-\boldsymbol{v}^\top\boldsymbol{C}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{b}+\boldsymbol{b}^\top\boldsymbol{b}+\boldsymbol{v}^\top\boldsymbol{d},$$

which is optimized at,

$$\boldsymbol{v}^\star = 2\left(\boldsymbol{C}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{C}^\top\right)^{-1}\left(\boldsymbol{d}-\boldsymbol{C}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{b}\right).$$

Strong duality also implies that $L(\boldsymbol{x},\boldsymbol{v}^\star)$ is optimized at $\boldsymbol{x}^\star$, which gives us,

$$\boldsymbol{x}^\star = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\left(\boldsymbol{A}^\top\boldsymbol{b}+\boldsymbol{C}^\top\left(\boldsymbol{C}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{C}^\top\right)^{-1}\left(\boldsymbol{d}-\boldsymbol{C}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{b}\right)\right).$$

We now note that we can compute $\boldsymbol{x}^\star$ by solving the following linear systems in order:

(1) Find inverse of $\boldsymbol{A}^\top \boldsymbol{A}$
(2) $\big(\boldsymbol{C}(\boldsymbol{A}^\top \boldsymbol{A})^{-1}\boldsymbol{C}^\top\big)x = \big(\boldsymbol{d} - \boldsymbol{C}(\boldsymbol{A}^\top \boldsymbol{A})^{-1}\boldsymbol{A}^\top \boldsymbol{b}\big)$

B CONVERTING ℓ_p-LAPLACIAN MINIMIZATION TO REGRESSION FORM

Define the following terms:

- n denote the number of vertices.
- l denote the number of labels.
- $\boldsymbol{B}$ denote the edge-vertex adjacency matrix.
- $\boldsymbol{g}$ denote the vector of labels for the l labelled vertices.
- $\boldsymbol{W}$ denote the diagonal matrix with weights of the edges.

Set $\boldsymbol{A} = \boldsymbol{W}^{1/p}\boldsymbol{B}$ and $\boldsymbol{b} = -\boldsymbol{B}[:, n : n+l]\boldsymbol{g}$. Now $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_p^p$ is equal to the ℓ_p laplacian and we can use our IRLS algorithm from Chapter 7 to find the $\boldsymbol{x}$ that minimizes this.

C INCREASING RESISTANCES

We first prove the following lemma that shows how much Ψ changes with a change in resistance.

Lemma C.1. *Let $\widetilde{\Delta} = \arg\min_{A\Delta=c} \Delta^\top \boldsymbol{M}^\top \boldsymbol{M}\Delta + \sum_e \boldsymbol{r}_e(\boldsymbol{N}\Delta)_e^2$. Then one has for any $\boldsymbol{r}'$ and $\boldsymbol{r}$ such that $\boldsymbol{r}' \geq \boldsymbol{r}$,*

$$\Psi(\boldsymbol{r}') \geq \Psi(\boldsymbol{r}) + \sum_e \left(1 - \frac{\boldsymbol{r}_e}{\boldsymbol{r}'_e}\right)\boldsymbol{r}_e(\boldsymbol{N}\widetilde{\Delta})_e^2.$$

Proof. For this proof, we use $\boldsymbol{R} = Diag(\boldsymbol{r})$.

$$\Psi(\boldsymbol{r}) = \min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{c}} \boldsymbol{x}^\top \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\boldsymbol{x}.$$

Constructing the Lagrangian and noting that strong duality holds,

$$\begin{aligned}\Psi(\boldsymbol{r}) &= \min_x \max_y \quad \boldsymbol{x}^\top \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\boldsymbol{x} + 2\boldsymbol{y}^\top(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x})\\ &= \max_y \min_x \quad \boldsymbol{x}^\top \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\boldsymbol{x} + 2\boldsymbol{y}^\top(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}).\end{aligned}$$

Optimality conditions with respect to $\boldsymbol{x}$ give us,

$$2\boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x}^\star + 2\boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\boldsymbol{x}^\star = 2\boldsymbol{A}^\top \boldsymbol{y}.$$

Substituting this in Ψ gives us,

$$\Psi(\boldsymbol{r}) = \max_y \quad 2\boldsymbol{y}^\top \boldsymbol{c} - \boldsymbol{y}^\top \boldsymbol{A}\big(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\big)^{-1}\boldsymbol{A}^\top \boldsymbol{y}.$$

Optimality conditions with respect to $\boldsymbol{y}$ now give us,

$$2\boldsymbol{c} = 2\boldsymbol{A}\big(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\big)^{-1}\boldsymbol{A}^\top \boldsymbol{y}^\star,$$

which upon re-substitution gives,

$$\Psi(\boldsymbol{r}) = \boldsymbol{c}^\top \Big(\boldsymbol{A}\big(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\big)^{-1}\boldsymbol{A}^\top\Big)^{-1}\boldsymbol{c}.$$

We also note that

$$\boldsymbol{x}^\star = \big(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\big)^{-1}\boldsymbol{A}^\top \Big(\boldsymbol{A}\big(\boldsymbol{M}^\top \boldsymbol{M} + \boldsymbol{N}^\top \boldsymbol{R}\boldsymbol{N}\big)^{-1}\boldsymbol{A}^\top\Big)^{-1}\boldsymbol{c}. \tag{15}$$

We now want to see what happens when we change $\boldsymbol{r}$. Let $\boldsymbol{R}$ denote the diagonal matrix with entries $\boldsymbol{r}$ and let $\boldsymbol{R}' = \boldsymbol{R} + \boldsymbol{S}$, where $\boldsymbol{S}$ is the diagonal matrix with the changes in the resistances. We will use the following version of the Sherman-Morrison-Woodbury formula multiple times,

$$(\boldsymbol{X} + \boldsymbol{UCV})^{-1} = \boldsymbol{X}^{-1} - \boldsymbol{X}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1} + \boldsymbol{VX}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VX}^{-1}.$$

We begin by applying the above formula for $\boldsymbol{X} = \boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}$, $\boldsymbol{C} = \boldsymbol{I}$, $\boldsymbol{U} = \boldsymbol{N}^\top\boldsymbol{S}^{1/2}$ and $\boldsymbol{V} = \boldsymbol{S}^{1/2}\boldsymbol{N}$. We thus get,

$$\begin{aligned}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{R}'\boldsymbol{N}\right)^{-1} &= \left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1} - \left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{N}^\top\boldsymbol{S}^{1/2} \\ &\quad \left(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{N}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{N}^\top\boldsymbol{S}^{1/2}\right)^{-1}\boldsymbol{S}^{1/2}\boldsymbol{N}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}. \quad (16)\end{aligned}$$

We next claim that,

$$\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{N}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{N}^\top\boldsymbol{S}^{1/2} \preceq \boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2},$$

which gives us,

$$\begin{aligned}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{R}'\boldsymbol{N}\right)^{-1} &\preceq \left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1} - \\ &\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{N}^\top\boldsymbol{S}^{1/2}(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}\boldsymbol{S}^{1/2}\boldsymbol{N}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}. \quad (17)\end{aligned}$$

This further implies,

$$\begin{aligned}\boldsymbol{A}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{R}'\boldsymbol{N}\right)^{-1}\boldsymbol{A}^\top &\preceq \boldsymbol{A}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{A}^\top - \\ &\boldsymbol{A}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{N}^\top\boldsymbol{S}^{1/2}(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}\boldsymbol{S}^{1/2}\boldsymbol{N}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN}\right)^{-1}\boldsymbol{A}^\top. \quad (18)\end{aligned}$$

We apply the Sherman-Morrison formula again for, $\boldsymbol{X} = \boldsymbol{A}(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN})^{-1}\boldsymbol{A}^\top$, $\boldsymbol{C} = -(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}$, $\boldsymbol{U} = \boldsymbol{A}(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN})^{-1}\boldsymbol{N}^\top\boldsymbol{S}^{1/2}$ and $\boldsymbol{V} = \boldsymbol{S}^{1/2}\boldsymbol{N}(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN})^{-1}\boldsymbol{A}^\top$. Let us look at the term $\boldsymbol{C}^{-1} + \boldsymbol{VX}^{-1}\boldsymbol{U}$.

$$-\left(\boldsymbol{C}^{-1} + \boldsymbol{VX}^{-1}\boldsymbol{U}\right)^{-1} = \left(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2} - \boldsymbol{VX}^{-1}\boldsymbol{U}\right)^{-1} \succeq (\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}.$$

Using this, we get,

$$\left(\boldsymbol{A}\left(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{R}'\boldsymbol{N}\right)^{-1}\boldsymbol{A}^\top\right)^{-1} \succeq \boldsymbol{X}^{-1} + \boldsymbol{X}^{-1}\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}\boldsymbol{VX}^{-1},$$

which on multiplying by $\boldsymbol{c}^\top$ and $\boldsymbol{c}$ gives,

$$\Psi(\boldsymbol{r}') \geq \Psi(\boldsymbol{r}) + \boldsymbol{c}^\top\boldsymbol{X}^{-1}\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}\boldsymbol{VX}^{-1}\boldsymbol{c}.$$

We note from Equation (15) that $\boldsymbol{x}^\star = (\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{N}^\top\boldsymbol{RN})^{-1}\boldsymbol{A}^\top\boldsymbol{X}^{-1}\boldsymbol{c}$. We thus have,

$$\begin{aligned}\Psi(\boldsymbol{r}') &\geq \Psi(\boldsymbol{r}) + \left(\boldsymbol{x}^\star\right)^\top\boldsymbol{N}^\top\boldsymbol{S}^{1/2}(\boldsymbol{I} + \boldsymbol{S}^{1/2}\boldsymbol{R}^{-1}\boldsymbol{S}^{1/2})^{-1}\boldsymbol{S}^{1/2}\boldsymbol{Nx}^\star \\ &= \Psi(\boldsymbol{r}) + \sum_e\left(\frac{\boldsymbol{r}_e' - \boldsymbol{r}_e}{\boldsymbol{r}_e'}\right)\boldsymbol{r}_e(\boldsymbol{Nx}^\star)_e^2.\end{aligned}$$

□