Distributionally Robust Linear Regression With Block Lewis Weights

Naren Sarayu Manoj NSM@TTIC.EDU
Kumar Kshitij Patel KKPATEL@TTIC.EDU
Toyota Technological Institute Chicago

Abstract

We present an algorithm for the empirical group distributionally robust (GDR) least squares problem. Given m groups, a parameter vector in $\mathbb{R}^d$, and stacked design matrices and responses $\mathbf{A}$ and $\boldsymbol{b}$, our algorithm obtains a $(1+\varepsilon)$-multiplicative optimal solution using $\widetilde{O}(\min(\mathsf{rank}(\mathbf{A}), m)^{1/3}\varepsilon^{-2/3})$ linear system solves of matrices of the form $\mathbf{A}^\top\mathbf{B}\mathbf{A}$ for block-diagonal $\mathbf{B}$. Our technical methods follow from a recent technique that relates the empirical GDR problem to a carefully chosen least squares problem and an application of ball-oracle acceleration. For moderate accuracy regimes, our algorithm improves over all known interior point methods and matches the state-of-the-art guarantees for the special case of ℓ_∞ regression.

1. Introduction

Machine learning algorithms and their training datasets have grown tremendously in the past decade, both in size and complexity. This increased model complexity has made it more challenging to interpret and predict their behavior in unobserved scenarios. Hence, many applications that involve societal decisions still rely on simple, interpretable models like linear regression (often after some feature engineering). Examples of such applications are predicting housing prices across cities, estimating wages across industries, forecasting loan amounts across banks, predicting life insurance premiums for different groups, and projecting energy consumption in various communities [SVWZ24].

A shared safety and sometimes legal concern across the above applications is the potential for unfair outcomes, i.e., outputting a notably worse model for some disadvantaged groups. Specifically, consider fitting a linear model $\boldsymbol{x} \in \mathbb{R}^d$ to make predictions on some task over n groups where group i's dataset consisting of n_i entries is denoted by $S_i = \{(\boldsymbol{a}_i^j, b_i^j)\}_{j\in[n_i]}$. The *utilitarian* or the social-cost-minimizing objective minimizes the weighted prediction error across groups, i.e.,

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \quad \frac{1}{\sum_{j\in[m]} n_j} \sum_{i\in[m]} n_i \cdot \left(\frac{1}{n_i}\left\|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\right\|_2^2\right) \quad , \tag{1.1}$$

where $\mathbf{A}_{S_i} \coloneqq [\boldsymbol{a}_i^i \dots \boldsymbol{a}_i^{n_i}]^\top \in \mathbb{R}^{n_i\times d}$ is the feature matrix and $\boldsymbol{b}_{S_i} \coloneqq [b_i^1 \dots b_i^{n_i}]^\top \in \mathbb{R}^{n_i}$ is the label vector for group $i \in [m]$. Note that the objective (1.1) is equivalent to ignoring any group differences and combining all the datasets into a single large data set of size $n \coloneqq \sum_{j\in[M]} n_j$. In particular, denoting the concatenation of all feature matrices by $\mathbf{A} \coloneqq [\mathbf{A}_{S_1}^\top \dots \mathbf{A}_{S_M}^\top]^\top$ and of all

the label vectors by $\boldsymbol{b} \coloneqq [\boldsymbol{b}_{S_1}^\top \dots \boldsymbol{b}_{S_M}^\top]^\top$, we get the following equivalent problem to (1.1),

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \quad \frac{1}{n}\left\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\right\|_2^2 = \frac{1}{n}\sum_{i\in[m],j\in[n_i]}\left(\left\langle \boldsymbol{a}_i^j, \boldsymbol{x}\right\rangle - b_i^j\right)^2 \quad . \tag{1.2}$$

Due to an imbalance in dataset sizes across groups or the presence of outlier behavior in some groups, the solution obtained by optimizing objective (1.1) might be unfair to some groups. Specifically, the prediction error might be disproportionately higher for those groups. To overcome these limitations, the following *egalitarian* or group Distributionally Robust Optimization (DRO) objective has been considered in several recent works [BDDMR13; DGN16; SKHL19; LCDS20; SGJ22; AAKMRZ22; SVWZ24],

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\max_{i\in[m]}\frac{1}{n_i}\left\|\mathbf{A}_{S_i}\boldsymbol{x}-\boldsymbol{b}_{S_i}\right\|_2^2 \quad . \tag{1.3}$$

Objective (1.3) is the "fairest" objective among all objectives that balance utility and fairness by adding group fairness constraints across demographic groups [KLMR18; CR18; ANSS22; CGSB22; RVFRWYT19], as it optimizes for the worst possible group's utility [GNPS24].

In this paper, we give a new algorithm to approximately optimize (1.3). We will be interested in finding $\widehat{\boldsymbol{x}} \in \mathbb{R}^d$ such that

$$\max_{i\in[m]}\frac{1}{n_i}\left\|\mathbf{A}_{S_i}\widehat{\boldsymbol{x}}-\boldsymbol{b}_{S_i}\right\|_2 \le (1+\varepsilon)\min_{\boldsymbol{x}\in\mathbb{R}^d}\max_{i\in[m]}\frac{1}{n_i}\left\|\mathbf{A}_{S_i}\boldsymbol{x}-\boldsymbol{b}_{S_i}\right\|_2 \quad , \tag{1.4}$$

for some pre-specified multiplicative-accuracy $\varepsilon > 0$. Since (1.3) is a convex problem, it is natural to apply standard black-box optimization techniques to solve the problem. However, we identify several challenges in applying existing methods:

- **Efficient first-order algorithms have geometry-dependent rates.** To our knowledge, using an efficient first-order method (such as sub-gradient descent) will incur a geometry-dependent runtime. In particular, if the matrices $\mathbf{A}_{S_i}$ or if the stacked matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ are poorly conditioned, then this will be reflected accordingly in the convergence rates. This is a drawback of the existing results by Abernethy, Awasthi, Kleindessner, Morgenstern, Russell, and Zhang [AAKMRZ22] and Song, Vakilian, Woodruff, and Zhou [SVWZ24].
- **The objective is not smooth.** As written, objective (1.3) may not be smooth. Since the objective is the pointwise maximum of several continuous functions, the derivative is not well-defined at the points at which the maximizing function changes. Thus, applying subgradient descent to this objective without a customized analysis will result in a rather unimpressive $1/\varepsilon^2$ dependence in the iteration complexity.
- **Min-max optimization approaches have a $1/\varepsilon^2$ dependence on iteration complexity.** Since problem 1.3 is a min-max optimization objective, it is also natural to try to use game theory-inspired approaches that use some oracle (such as gradients) for each group as a black box. Perhaps the most basic such algorithm is casting objective (1.3) as a repeated game between a min player (equipped with a no-regret algorithm) and a max player (equipped with the best response oracle). There are two shortcomings to this approach: first, the guarantee is inherently weaker than (1.3) as the min player must randomize to get low regret (thus

the output $\hat{\boldsymbol{x}}$ is random); second even though the function for each group is smooth, the iteration complexity (to get ε average regret) for smooth online convex optimization still has an unimpressive $1/\varepsilon^2$ dependence (as opposed to $1/\varepsilon$ for smooth convex optimization). Thus, this approach is no better than directly applying sub-gradient descent to objective (1.3). Several works have built upon this idea [SGJ22; ZZZYZ24].

- **Interior point methods have a poor iteration complexity for large m.** Another natural approach (that can partially address the previous two issues), following the discussion by Boyd and Vandenberghe [BV04, Section 6.4], is to rewrite the problem (1.3) in its epigraph form and use an interior point method (IPM) to solve the resulting problem (which is, in this case, a quadratically constrained linear program). Unfortunately, this will give an algorithm whose analysis is only known to yield an iteration complexity of $O(\sqrt{m})$, where each iteration solves a linear system in matrices of the form $\mathbf{A}^\top\mathbf{B}\mathbf{A}$ for a block-diagonal $\mathbf{B}$. A naïve implementation of this algorithm will, therefore, have a superlinear runtime in the number of groups, which is undesirable when the number of groups is large. Furthermore, note that (1.3) is not a linear program when at least one group i is such that $n_i > 1$. So, we cannot immediately apply recent advances in linear programming that get iteration complexities independent of the number of constraints [LS19].

Hence, designing an algorithm without these shortcomings requires novel ideas.

2. Our results

Our main result is an algorithm for optimizing (1.3) up to the guarantee (1.4) that overcomes all of the difficulties mentioned in the previous section. We state our guarantee in the following theorem.

Theorem 1. *Let $\mathbf{A}_{S_1}, \ldots, \mathbf{A}_{S_m}$ be such that $\mathbf{A}_{S_i} \in \mathbb{R}^{n_i \times d}$ and let $\boldsymbol{b}_{S_1}, \ldots, \boldsymbol{b}_{S_m}$ be such that $\boldsymbol{b}_{S_i} \in \mathbb{R}^{|S_i|}$. Let $\mathbf{A} \in \mathbb{R}^{(\sum_{i=1}^m n_i) \times d}$ and $\boldsymbol{b} \in \mathbb{R}^{\sum_{i=1}^m n_i}$ be formed by stacking the $\mathbf{A}_{S_i}$ and $\boldsymbol{b}_{S_i}$. Let $\varepsilon > 0$, then there exists an algorithm (Algorithm 1) that returns $\widehat{\boldsymbol{x}}$ satisfying (1.4) and runs in*

$$O\left(\frac{\min\left(\mathsf{rank}(\mathbf{A}), m\right)^{1/3}\left(\log\left(n\log m/\varepsilon\right)^{14/3} + \log\left(m\right)\right)}{\varepsilon^{2/3}}\right)$$

linear system solves in matrices of the form $\mathbf{A}^\top\mathbf{B}\mathbf{A}$, where $\mathbf{B}$ is a block-diagonal matrix where each block has size $|S_i| \times |S_i|$.

We compare the guarantee of Theorem 1 against those of the other black-box methods that we are aware of in Table 1. From this, we see that unlike the mentioned first-order methods, our algorithm does not have any geometry-dependent terms. Additionally, our algorithm improves over the standard log-barrier IPM when the desired accuracy $\varepsilon \geq m^{-1/4}$ – this improvement is more pronounced when $m \gg \mathsf{rank}\left(\mathbf{A}\right)$, which is the case in which the number of data sources is much larger than the dimension of the parameter vector $\boldsymbol{x}$. Additionally, for $\varepsilon \geq \mathsf{rank}\left(\mathbf{A}\right)^{-1/4}$, our guarantee matches the best known guarantee for ℓ_∞ regression [LS19; JLS22].

Algorithm	Iteration Complexity	Each Iteration
Subgradient Descent	$\frac{\|\boldsymbol{x}^\star\|_2 \max_{1\le i\le m}\|\mathbf{A}_{S_i}\|_{\mathrm{op}}}{\varepsilon^2}$	Evaluate $\nabla f(\boldsymbol{x})$
Nesterov Acceleration on Smoothened Objective	$\frac{\|\boldsymbol{x}^\star\|_2 \left(\max_{1\le i\le m}\|\mathbf{A}_{S_i}\|_{\mathrm{op}}\right)^{1/2}}{\varepsilon}$	Evaluate $\nabla \widetilde{f}_{\beta,\delta}(\boldsymbol{x})$
[AAKMRZ22]	$\frac{\|\boldsymbol{x}^\star\|_2 \max_{1\le i\le m}\|\mathbf{A}_{S_i}\|_{\mathrm{op}}}{\varepsilon}$	Evaluate $\nabla \widetilde{f}_{\beta,\delta}(\boldsymbol{x})$
Interior Point with Log-Barrier [BV04]	$m^{1/2}\log\left(\frac{1}{\varepsilon}\right)$	Linear system solve in $\mathbf{A}^\top\mathbf{B}\mathbf{A}$
Naïve application of [CJJJLST20]	$\frac{m^{1/3}}{\varepsilon^{2/3}}$	Linear system solve in $\mathbf{A}^\top\mathbf{B}\mathbf{A}$
ℓ_∞ Regression with Lewis Weights [JLS22]	$\frac{\mathsf{rank}(\mathbf{A})^{1/3}}{\varepsilon^{2/3}}$	Linear system solve in $\mathbf{A}^\top\mathbf{D}\mathbf{A}$
ℓ_∞ Regression with IPM [LS19]	$\mathsf{rank}\,(\mathbf{A})^{1/2}\log\left(\frac{1}{\varepsilon}\right)$	Linear system solve in $\mathbf{A}^\top\mathbf{D}\mathbf{A}$
This Paper (Theorem 1)	$\frac{\min(\mathsf{rank}(\mathbf{A}),m)^{1/3}}{\varepsilon^{2/3}}$	Linear system solve in $\mathbf{A}^\top\mathbf{B}\mathbf{A}$

Table 1: Here, we list the complexities of algorithms for optimizing (1.3) or for ℓ_∞ regression, assuming $\mathsf{OPT} = 1$ (the first three guarantees are additive approximations) and ignoring $\mathrm{polylog}(n, m)$ terms. We write $\mathbf{D}$ to be a diagonal matrix and $\mathbf{B}$ to be a block-diagonal matrix where each block has size $|S_i| \times |S_i| + O(|S_i|)$. To explain why we describe the second-order method results in terms of linear system solve complexity, see the discussion in [JLS22, Section 1.2]. We remark that in the special case where $|S_i| = 1$, our algorithm exactly recovers that of [JLS22].

3. Our algorithm and technical overview

For the rest of the paper, for $\boldsymbol{c} \in \mathbb{R}^d$, let $f(\boldsymbol{x}) \coloneqq \max_{1\le i\le m} \|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\|_2$ (we drop the scalings n_i as they can simply be folded into $\mathbf{A}_{S_i}$ and $\boldsymbol{b}_{S_i}$).

Without loss of generality (by rescaling), let $\mathsf{OPT} \ge 1$, where $\mathsf{OPT} \coloneqq \min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. So, it is enough to get an ε-additive optimal solution $\widehat{\boldsymbol{x}}$.

Our algorithm (Algorithm 1) and analysis use the ball oracle acceleration framework of Carmon, Jambulapati, Jiang, Jin, Lee, Sidford, and Tian [CJJJLST20], which itself exploits an acceleration framework due to Monteiro and Svaiter [MS13]. At a high level, the ball oracle acceleration framework breaks the problem of optimizing a smooth convex function f into subproblems of the form

$$\operatorname*{argmin}_{\|\boldsymbol{x}-\overline{\boldsymbol{x}}\|_{\mathbf{M}}\le r} f(\boldsymbol{x}), \tag{3.1}$$

where $\|\boldsymbol{x}\|_{\mathbf{M}} \coloneqq \sqrt{\boldsymbol{x}^\top\mathbf{M}\boldsymbol{x}}$ for positive semidefinite $\mathbf{M}$. The main result of [CJJJLST20] is that if we can identify $\mathbf{M}$ for which we can implement an approximate solver for (3.1) (called a "ball

optimization oracle") and for which we can identify an initialization $\boldsymbol{x}_0$ with $f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star) \le \varepsilon_0$, then we can optimize f up to ε additive accuracy in $\widetilde{O}\left(\left(\|\boldsymbol{x}^\star - \boldsymbol{x}_0\|_{\mathbf{M}}/r\right)^{2/3} \log\left(\varepsilon_0/\varepsilon\right)\right)$ calls to the ball oracle.

To apply this result in this form, we need to choose $\mathbf{M}$ and some surrogate objective $\widetilde{f}$ so that:

1. The approximation error $\left\|\widetilde{f} - f\right\|_\infty$ is small;
2. The surrogate objective $\widetilde{f}$ is smooth in $\|\cdot\|_{\mathbf{M}}$;
3. We can find an initialization $\boldsymbol{x}_0$ that witnesses both a small $\|\boldsymbol{x}^\star - \boldsymbol{x}_0\|_{\mathbf{M}}$ and $\widetilde{f}(\boldsymbol{x}_0) - \widetilde{f}(\boldsymbol{x}^\star)$;
4. With an appropriate choice of r, the ball oracle subproblems (3.1) (but using the surrogate $\widetilde{f}$ in place of f) can be implemented efficiently.

To smoothen $f(\boldsymbol{x})$, we consider the family of objectives parameterized by β, δ

$$\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) \coloneqq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{\sqrt{\delta^2 + \|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\|_2^2} - \delta}{\beta}\right)\right).$$

It is straightforward to show that for all $\boldsymbol{x} \in \mathbb{R}^d$, $\left|\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) - f(\boldsymbol{x})\right| \le \beta \log m + \delta$, so setting $\beta = \varepsilon/4\log m$ and $\delta = \varepsilon/4$, it is sufficient to optimize $\widetilde{f}_{\beta,\delta}$ up to $\varepsilon/2$ additive error to get an ε-additive suboptimal solution to our original objective. Furthermore, $\widetilde{f}$ as written can be shown to be $O(1/\beta + 1/\delta)$-smooth in the norm $\|\mathbf{A}\boldsymbol{x}\|_{\mathcal{G}_\infty} \coloneqq \max_{1\le i\le m}\|\mathbf{A}\boldsymbol{x}\|_2$. This gives us our first requirement on $\mathbf{M}$ – namely, that for all $\boldsymbol{x} \in \mathbb{R}^d$, we have $\|\mathbf{A}\boldsymbol{x}\|_{\mathcal{G}_\infty} \le \|\boldsymbol{x}\|_{\mathbf{M}}$ (so that we get that $\widetilde{f}$ is smooth in $\|\cdot\|_{\mathbf{M}}$).

For the next desideratum, it will be enough to let $\mathbf{M} = \mathbf{A}^\top\mathbf{W}\mathbf{A}$ for positive diagonal $\mathbf{W}$ for which

$$\text{for all } \boldsymbol{x} \in \mathbb{R}^d\text{:} \quad f(\boldsymbol{x}) \le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x} - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2 \le Cf(\boldsymbol{x}).$$

Then, setting $\boldsymbol{x}_0$ to the optimal point for the least-squares objective $\min_{\boldsymbol{x}_0\in\mathbb{R}^d}\left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0 - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2$, we get by optimality that $f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star) \le C - 1$ and $\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_{\mathbf{M}} \le 2C$.

For the final item, we leverage the analysis of [CJJJLST20], which shows that if $\widetilde{f}$ satisfies a higher-order smoothness condition called *quasi self concordance* with respect to the norm $\|\cdot\|_{\mathbf{M}}$, then we can choose $r = \widetilde{\Theta}(\varepsilon)$ and implement (3.1) with low linear system solve iteration complexity. We prove that $\widetilde{f}$ is $O(1/\beta + 1/\delta)$-quasi self concordant in the norm $\max_{i\in[m]}\|\mathbf{A}_{S_i}\boldsymbol{x}\|_2$, so once again we need for all $\boldsymbol{x} \in \mathbb{R}^d$ that $\max_{i\in[m]}\|\mathbf{A}_{S_i}\boldsymbol{x}\|_2 \le \left\|\mathbf{M}^{1/2}\boldsymbol{x}\right\|_2$.

At this point, it remains how to choose $\mathbf{M} = \mathbf{A}^\top\mathbf{W}\mathbf{A}$, where $\mathbf{W}$ is a positive diagonal matrix. Choosing $\mathbf{W} = \mathbf{I}$ yields $C = \sqrt{m}$ by relating ℓ_2^m and ℓ_∞^m (norms defined over $\mathbb{R}^m$), which gives an iteration complexity of $\widetilde{O}(m^{1/3}\varepsilon^{-2/3})$. However, Manoj and Ovsiankin [MO25] give an algorithm that, with $\widetilde{O}(1)$ linear system solves in matrices $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for diagonal $\mathbf{D}$, finds $\mathbf{W}$ such that we get $C = O(\sqrt{\mathsf{rank}(\mathbf{A})})$. The resulting weights can be seen as a generalization of Lewis weights, which have been fundamental in getting tight geometric relationships between subspaces of ℓ_p and

ℓ_2 [LS19; JLS22] and for various matrix approximation problems [BLM89; MMWY22]. This choice of $\mathbf{W}$ then yields a tighter relationship between a subspace of $\|\cdot\|_{\mathcal{G}_\infty}$ and an ℓ_2 geometry.

Thus, plugging in this algorithm and $C = \sqrt{\mathsf{rank}(\mathbf{A})}$ yields a $\widetilde{O}(\mathsf{rank}(\mathbf{A})^{1/3}\varepsilon^{-2/3})$ iteration complexity. Finally, switching based on whether $\mathsf{rank}(\mathbf{A}) \leq m$ concludes the proof. We state the full algorithm in Appendix A.

4. Future work

It would be exciting to see whether one could use inverse maintenance techniques (such as those of Lee and Sidford [LS19]) to obtain a low amortized runtime for each linear system solution. Another exciting (but probably challenging) open problem is to design high-accuracy algorithms for minimizing (1.3) whose iteration complexities are independent of the number of groups m. For the particular case of ℓ_∞ regression, the state-of-the-art follows from [LS19], which gives a $\sqrt{\mathsf{rank}\,(\mathbf{A})}$-iteration complexity via a specialized self-concordant barrier construction.

References

[AAKMRZ22] Jacob D Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. Active sampling for min-max fairness. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 53–65. PMLR, July 2022. URL: https://proceedings.mlr.press/v162/abernethy22a.html (cited on pages 2, 4).

[ANSS22] Arash Asadpour, Rad Niazadeh, Amin Saberi, and Ali Shameli. Sequential submodular maximization and applications to ranking an assortment of products. *Operations Research*, 2022 (cited on page 2).

[BDDMR13] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013 (cited on page 2).

[BLM89] Jean Bourgain, Joram Lindenstrauss, and Vitali Milman. Approximation of zonoids by zonotopes, 1989 (cited on page 6).

[BV04] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004 (cited on pages 3, 4).

[CJJJLST20] Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. Acceleration with a ball optimization oracle. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc., 2020. ISBN: 9781713829546. arXiv: 2003.08078 [math.OC] (cited on pages 4, 5, 9, 10, 16, 18).

[CGSB22] Qinyi Chen, Negin Golrezaei, Fransisca Susan, and Edy Baskoro. Fair assortment planning. *arXiv preprint arXiv:2208.07341*, 2022 (cited on page 2).

[CR18] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018 (cited on page 2).

[DGN16] J Duchi, P Glynn, and Hongseok Namkoong. Statistics of robust optimization: a generalized empirical likelihood approach. arxiv. *Machine Learning*, 2016 (cited on page 2).

[Gir14] Davide Giraudo. Bound the variance of the product of two random varables. Mathematics Stack Exchange, November 2014. URL: https://math.stackexchange.com/q/1044864 (cited on page 12).

[GNPS24] Negin Golrezaei, Rad Niazadeh, Kumar Kshitij Patel, and Fransisca Susan. Online combinatorial optimization with group fairness constraints. *Available at SSRN 4824251*, 2024 (cited on page 2).

[JLS22] Arun Jambulapati, Yang P Liu, and Aaron Sidford. Improved iteration complexities for overconstrained p-norm regression. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 529–542, 2022. arXiv: 2111.01848 [cs.DS] (cited on pages 3, 4, 6, 16).

[KLMR18] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pages 22–27, 2018 (cited on page 2).

[LS19] Yin Tat Lee and Aaron Sidford. Solving linear programs with sqrt(rank) linear system solves, 2019. arXiv: 1910.08033 [cs.DS] (cited on pages 3, 4, 6).

[LCDS20] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020 (cited on page 2).

[MO25] Naren Sarayu Manoj and Max Ovsiankin. *The change-of-measure method, block lewis weights, and approximating matrix block norms*. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2025. arXiv: 2311.10013 [math.FA] (cited on pages 5, 9, 16, 19).

[MS13] Renato D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013. DOI: 10.1137/110833786. eprint: https://doi.org/10.1137/110833786. URL: https://doi.org/10.1137/110833786 (cited on page 4).

[MMWY22] Cameron Musco, Christopher Musco, David P Woodruff, and Taisuke Yasuda. Active linear regression for ℓ_p norms and beyond. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 744–753. IEEE, 2022. arXiv: 2111.04888 [cs.LG] (cited on page 6).

[OB20] Dmitrii Ostrovskii and Francis Bach. Finite-sample analysis of m-estimators using self-concordance, 2020. arXiv: 1810.06838 [math.ST]. URL: https://arxiv.org/abs/1810.06838 (cited on page 15).

[RVFRWYT19] Aida Rahmattalabi, Phebe Vayanos, Anthony Fulginiti, Eric Rice, Bryan Wilder, Amulya Yadav, and Milind Tambe. Exploring algorithmic fairness in robust graph covering problems. *Advances in neural information processing systems*, 32, 2019 (cited on page 2).

[SKHL19] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: on the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019 (cited on page 2).

[SGJ22] Tasuku Soma, Khashayar Gatmiry, and Stefanie Jegelka. Optimal algorithms for group distributionally robust optimization and beyond. *arXiv preprint arXiv:2212.13669*, 2022 (cited on pages 2, 3).

[SVWZ24] Zhao Song, Ali Vakilian, David Woodruff, and Samson Zhou. On socially fair regression and low-rank approximation, 2024. URL: https://openreview.net/forum?id=KJHUYWviZ6 (cited on pages 1, 2).

[ZZZYZ24] Lijun Zhang, Peng Zhao, Zhen-Hua Zhuang, Tianbao Yang, and Zhi-Hua Zhou. Stochastic approximation approaches to group distributionally robust optimization. *Advances in Neural Information Processing Systems*, 36, 2024 (cited on page 3).

Appendix A. Full Algorithm

Algorithm 1 MinMaxRegression: optimizes (1.3) to $(1+\varepsilon)$-multiplicative error

1: **Input:** Regression problems $(\mathbf{A}_{S_1}, \boldsymbol{b}_{S_1}), \ldots, (\mathbf{A}_{S_m}, \boldsymbol{b}_{S_m})$, accuracy $\varepsilon > 0$

2: Using [MO25, Algorithm 2] with input $[\mathbf{A}|\boldsymbol{b}]$, find nonnegative diagonal $\mathbf{W}$ such that for all $\boldsymbol{x} \in \mathbb{R}^d$ and $c \in \mathbb{R}$,

$$\|\mathbf{A}\boldsymbol{x} - c\boldsymbol{b}\|_{\mathcal{G}_\infty} \le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x} - c\mathbf{W}^{1/2}\boldsymbol{b}\right\|_2 \le \sqrt{2(d+1)}\,\|\mathbf{A}\boldsymbol{x} - c\boldsymbol{b}\|_{\mathcal{G}_\infty}.$$

3: Let $\boldsymbol{x}_0 = \left(\mathbf{A}^\top\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{W}\boldsymbol{b}$. ▷ $\boldsymbol{x}_0 \coloneqq \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x} - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2$.

4: Let

$$\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) \coloneqq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{\sqrt{\delta^2 + \|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\|_2^2} - \delta}{\beta}\right)\right)$$

▷ A family of smoothenings of the objective.

5: Let $\widehat{f}(\boldsymbol{x}) \coloneqq \widetilde{f}_{\varepsilon/4\log m,\varepsilon/4}(\boldsymbol{x}) + \frac{\varepsilon}{1000(d+1)}\left\|\mathbf{W}^{1/2}\mathbf{A}(\boldsymbol{x} - \boldsymbol{x}_0)\right\|_2^2$.

6: Using [CJJJLST20, Algorithm 3], implement a $\left(C \cdot \sqrt{\frac{2}{d+1}}, \frac{76}{\varepsilon}\right)$-ball optimization oracle for $\widehat{f}$, where C is a universal constant.

7: Using [CJJJLST20, Algorithm 2] and the oracle from the previous line, implement a $\frac{1}{2}$-MS oracle for $\widehat{f}$.

8: Run [CJJJLST20, Algorithm 1] for $\widetilde{O}(n^{1/3}\varepsilon^{-2/3})$ iterations using the MS oracle from the previous line and with initial point $\boldsymbol{x}_0$ and final point $\widehat{\boldsymbol{x}}$.

9: **return** $\widehat{\boldsymbol{x}}$

Appendix B. Analysis

It may be helpful to refer to the overview in Section 3 for an outline of the analysis of Algorithm 1. For $\boldsymbol{y} \in \mathbb{R}^n$, let $\|\boldsymbol{y}\|_{\mathcal{G}_\infty} \coloneqq \max_{1\le i\le m}\|\boldsymbol{y}_{S_i}\|_2$, where for $\boldsymbol{y} \in \mathbb{R}^n$ we let $\boldsymbol{y}_{S_i}$ refer to the vector in $\mathbb{R}^{n_i}$ indexed by the indices in S_i. Also, for $\boldsymbol{y} \in \mathbb{R}^m$, let $\mathsf{lse}_\beta(\boldsymbol{y})$ refer to the function

$$\mathsf{lse}_\beta(\boldsymbol{y}) \coloneqq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{y_i}{\beta}\right)\right).$$

At a high level, our algorithm will minimize the function

$$\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) \coloneqq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{\sqrt{\delta^2 + \|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\|_2^2} - \delta}{\beta}\right)\right)$$

for appropriate choices of the parameters β and δ. This choice of smoothening is natural because of the following approximation statement – see Lemma B.1.

Lemma B.1. *For all $\boldsymbol{x} \in \mathbb{R}^d$, we have*

$$\left|\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) - \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_{\mathcal{G}_\infty}\right| \leq \beta \log m + \delta.$$

Proof of Lemma B.1. These guarantees are well-known, but we prove them anyway for the sake of self-containment. We first prove that for any $\boldsymbol{v} \in \mathbb{R}^m$, we have

$$\max_{1\leq i\leq m} v_i \leq \mathsf{lse}_\beta(\boldsymbol{v}) \leq \max_{1\leq i\leq m} v_i + \beta \log m.$$

In one direction, we have

$$\mathsf{lse}_\beta(\boldsymbol{v}) \leq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{\max_{1\leq i\leq m} v_i}{\beta}\right)\right) = \beta \log m + \max_{1\leq i\leq m} v_i,$$

and in the other, we have

$$\mathsf{lse}_\beta(\boldsymbol{v}) \geq \beta \log\left(\exp\left(\frac{\max_{1\leq i\leq m} v_i}{\beta}\right)\right) = \max_{1\leq i\leq m} v_i.$$

Next, for $\boldsymbol{v} \in \mathbb{R}^m$, we will show that

$$\|\boldsymbol{v}\|_2 - \delta \leq \sqrt{\delta^2 + \|\boldsymbol{v}\|_2^2} - \delta \leq \|\boldsymbol{v}\|_2.$$

Indeed, we have

$$\sqrt{\delta^2 + \|\boldsymbol{v}\|_2^2} - \delta \leq \sqrt{\delta^2} + \sqrt{\|\boldsymbol{v}\|_2^2} - \delta = \|\boldsymbol{v}\|_2,$$

and

$$\sqrt{\delta^2 + \|\boldsymbol{v}\|_2^2} - \delta \geq \sqrt{\|\boldsymbol{v}\|_2^2} - \delta = \|\boldsymbol{v}\|_2 - \delta.$$

From this, we get

$$\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) \leq \max_{1\leq i\leq m}\left(\sqrt{\delta^2 + \|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\|_2^2} - \delta\right) + \beta \log m \leq \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_{\mathcal{G}_\infty} + \beta \log m$$

and

$$\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) \geq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{\|\mathbf{A}_{S_i}\boldsymbol{x} - \boldsymbol{b}_{S_i}\|_2 - \delta}{\beta}\right)\right) \geq \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_{\mathcal{G}_\infty} - \delta.$$

Putting these together gives

$$\left|\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) - \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_{\mathcal{G}_\infty}\right| \leq \max(\beta \log m, \delta) \leq \beta \log m + \delta,$$

completing the proof of Lemma B.1. □

Eventually, we will choose $\beta = \varepsilon/(4 \log m)$ and $\delta = \varepsilon/4$ and then minimize $\widetilde{f}_{\beta,\delta}$ to $\varepsilon/2$ additive error. In light of Lemma B.1, this will be enough to get an ε-additive approximation to the optimum for $\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_{\mathcal{G}_\infty}$.

The rest of this section is organized as follows. In Appendix B.1, we derive the smoothness results we need regarding a family of objectives that generalizes $\widetilde{f}_{\beta,\delta}$. In Appendix B.2, we specialize these results to $\widetilde{f}_{\beta,\delta}$. Finally, in Appendix B.3, we combine these results with the framework from [CJJJLST20] to complete the proof of Theorem 1.

B.1. Calculus for LOGSUMEXP

We investigate certain properties of $\mathsf{lse}_\beta(\boldsymbol{y})$ when each entry $[\boldsymbol{y}]_i$ is a function $h_i(t)$ for $t \in \mathbb{R}$ for all $i \in [m]$. Let $h(t) \in \mathbb{R}^m$ denote the vector where its ith entry is given by $h_i(t)$. We treat each h_i as a one-dimensional restriction of a function $g_i \colon \mathbb{R}^m \to \mathbb{R}$, so $h_i(t) = g_i(\boldsymbol{y} + t\boldsymbol{d})$ for center $\boldsymbol{y}$ and direction $\boldsymbol{d}$ (we omit the parameters $\boldsymbol{y}, \boldsymbol{d}$ in the notation h_i as it will be clear from context).

We begin with calculating the first two derivatives of $\mathsf{lse}_\beta(h(t))$ with respect to t in Lemma B.2.

Lemma B.2. *Let $\lambda_i(t) \coloneqq \exp\left(h_i(t)/\beta\right)$. Then, we have*

$$\left(\frac{d}{dt}\right) \mathsf{lse}_\beta(h(t)) = \frac{\sum_{i=1}^m \left(\lambda_i(t) \cdot h_i'(t)\right)}{\sum_{i=1}^m \lambda_i(t)}$$
$$\left(\frac{d}{dt}\right)^2 \mathsf{lse}_\beta(h(t)) = \frac{1}{\beta}\left(\frac{\sum_{i=1}^m \lambda_i(t) h_i'(t)^2}{\sum_{i=1}^m \lambda_i(t)} - \left(\frac{\sum_{i=1}^m \lambda_i(t) h_i'(t)}{\sum_{i=1}^m \lambda_i(t)}\right)^2\right) + \frac{\sum_{i=1}^m \lambda_i(t) h_i''(t)}{\sum_{i=1}^m \lambda_i(t)}.$$

Proof of Lemma B.2. The first derivative follows from the chain rule. Indeed, we have

$$\mathsf{lse}_\beta'(h(t)) = \beta \cdot \frac{\sum_{i=1}^m \lambda_i'(t)}{\sum_{i=1}^m \lambda_i(t)} = \beta \cdot \frac{\sum_{i=1}^m \left(\lambda_i(t) \cdot \frac{h_i'(t)}{\beta}\right)}{\sum_{i=1}^m \lambda_i(t)} = \frac{\sum_{i=1}^m \left(\lambda_i(t) \cdot h_i'(t)\right)}{\sum_{i=1}^m \lambda_i(t)} \le \max_i h_i'(t).$$

For the second derivative, we use the differentiation rule for multiplication and division and the chain rule, giving

$$\begin{aligned}
\mathsf{lse}_\beta''(h(t)) &= \frac{\left[\left(\sum_{i=1}^m \lambda_i'(t) h_i'(t) + \lambda_i(t) h_i''(t)\right)\left(\sum_{i=1}^m \lambda_i(t)\right)\right] - \frac{1}{\beta}\left(\sum_{i=1}^m \lambda_i(t) h_i'(t)\right)^2}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2} \\
&= \frac{\left[\frac{1}{\beta}\left(\sum_{i=1}^m \lambda_i(t) h_i'(t)^2 + \beta \lambda_i(t) h_i''(t)\right)\left(\sum_{i=1}^m \lambda_i(t)\right)\right] - \frac{1}{\beta}\left(\sum_{i=1}^m \lambda_i(t) h_i'(t)\right)^2}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2} \\
&= \frac{1}{\beta}\left(\frac{\sum_{i=1}^m \lambda_i(t) h_i'(t)^2}{\sum_{i=1}^m \lambda_i(t)} - \frac{\left(\sum_{i=1}^m \lambda_i(t) h_i'(t)\right)^2}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2}\right) + \frac{\sum_{i=1}^m \lambda_i(t) h_i''(t)}{\sum_{i=1}^m \lambda_i(t)}.
\end{aligned}$$

This completes the proof of Lemma B.2. □

Next, we prove a general fact regarding composing lse with a vector formed by functions that are themselves quasi self concordant. See Lemma B.3.

Lemma B.3. *Let $\|\cdot\|$ be an arbitrary norm and $h_1, \ldots, h_m$ be such that for all $1 \le i \le m$ and for all $\boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$ and $t \in \mathbb{R}$,*

$$\left(\frac{d}{dt}\right) h_i(t) \le \|\boldsymbol{d}\| \qquad \textit{(Lipschitzness)}$$
$$\left|\left(\frac{d}{dt}\right)^3 h_i(t)\right| \le \nu \|\boldsymbol{d}\| \left(\frac{d}{dt}\right)^2 h_i(t) \qquad \textit{(quasi self concordance)}.$$

Then, for all $\boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$ and all $t \in \mathbb{R}$, we have

$$\left|\left(\frac{d}{dt}\right)^3 \mathsf{lse}_\beta(h(t))\right| \le \left(\frac{16}{\beta} + \nu\right) \|\boldsymbol{d}\| \left(\frac{d}{dt}\right)^2 \mathsf{lse}_\beta(h(t)).$$

As far as we are aware, this type of composition result was not previously known and may be of independent interest.

To prove Lemma B.3, we need Lemma B.4.

Lemma B.4. *For any two random variables X, Y, we have*

$$\mathsf{Var}\left[XY\right] \le 2\left\|Y\right\|_\infty^2 \mathsf{Var}\left[X\right] + 2\left\|X\right\|_\infty^2 \mathsf{Var}\left[Y\right].$$

Proof of Lemma B.4. The proof follows that of [Gir14], but we reproduce it here for completeness. First, notice that for random variables U, V, we have

$$2\mathsf{Var}\left[U\right] + 2\mathsf{Var}\left[V\right] - \mathsf{Var}\left[U+V\right] = \mathsf{Var}\left[U\right] + \mathsf{Var}\left[V\right] - 2\mathsf{Cov}\left[U,V\right] = \mathsf{Var}\left[U-V\right] \ge 0.$$

Let $U = (X - \mathbb{E}\left[X\right])Y$ and $V = \mathbb{E}\left[X\right]Y$. Then, $U + V = XY$, and we have

$$\mathsf{Var}\left[XY\right] \le 2\mathsf{Var}\left[(X-\mathbb{E}\left[X\right])Y\right] + 2\mathsf{Var}\left[\mathbb{E}\left[X\right]Y\right] = 2\mathsf{Var}\left[(X-\mathbb{E}\left[X\right])Y\right] + 2\mathbb{E}\left[X\right]^2\mathsf{Var}\left[Y\right].$$

It remains to bound $\mathsf{Var}\left[(X - \mathbb{E}\left[X\right])Y\right]$. By Hölder's inequality, we have

$$\mathsf{Var}\left[(X-\mathbb{E}\left[X\right])Y\right] \le \mathbb{E}\left[((X-\mathbb{E}\left[X\right])Y)^2\right] \le \mathbb{E}\left[(X-\mathbb{E}\left[X\right])^2\right]\left\|Y\right\|_\infty^2 = \mathsf{Var}\left[X\right]\left\|Y\right\|_\infty^2.$$

Combining everything gives us the conclusion of Lemma B.4. □

We are now ready to prove Lemma B.3.

Proof of Lemma B.3. Let $\lambda_i(t) \coloneqq \exp\left(h_i(t)/\beta\right)$.

In this proof, we will encounter many weighted averages of vectors $\boldsymbol{z} \in \mathbb{R}^m$ of the form

$$\frac{\sum_{i=1}^m \lambda_i(t) z_i}{\sum_{i=1}^m \lambda_i(t)}.$$

Let $\mathcal{D}$ be the distribution over $[m]$ whose entries are given by $\mathcal{D}_j = \lambda_j(t)/\sum_{i=1}^m \lambda_i(t)$. In the rest of this proof, all expected values, variances, and covariances will be taken with respect to this distribution. In an abuse of notation, let $h(t)$ denote the "random" variable that is $h_i(t)$ with probability $\mathcal{D}_i$. Define $h'(t), h''(t), h'''(t)$ analogously.

To find the third derivative of $\mathsf{lse}_\beta(h(t))$, we start with its second derivative. By Lemma B.2, it is given by

$$\begin{aligned}\mathsf{lse}_\beta''(h(t)) &= \underbrace{\frac{1}{\beta}\left(\frac{\sum_{i=1}^m \lambda_i(t)h_i'(t)^2}{\sum_{i=1}^m \lambda_i(t)} - \left(\frac{\sum_{i=1}^m \lambda_i(t)h_i'(t)}{\sum_{i=1}^m \lambda_i(t)}\right)^2\right)}_{T_1} + \underbrace{\frac{\sum_{i=1}^m \lambda_i(t)h_i''(t)}{\sum_{i=1}^m \lambda_i(t)}}_{T_2} \\ &= \frac{1}{\beta}\mathsf{Var}\left[h'(t)\right] + \mathbb{E}\left[h''(t)\right].\end{aligned}$$

We now differentiate the above term by term. First, we have

$$T_2'(t) = \frac{\sum_{i=1}^m \lambda_i(t)\left(\left(\frac{h_i'(t)h_i''(t)}{\beta}\right) + h_i'''(t)\right)}{\sum_{i=1}^m \lambda_i(t)} - \frac{1}{\beta}\cdot\frac{\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)\right)\left(\sum_{i=1}^m \lambda_i(t)h_i''(t)\right)}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2}$$

$$= \frac{1}{\beta}\left(\frac{\sum_{i=1}^m \lambda_i(t)h_i'(t)h_i''(t)}{\sum_{i=1}^m \lambda_i(t)} - \frac{\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)\right)\left(\sum_{i=1}^m \lambda_i(t)h_i''(t)\right)}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2}\right) + \frac{\sum_{i=1}^m \lambda_i(t)h_i'''(t)}{\sum_{i=1}^m \lambda_i(t)}$$
$$= \frac{1}{\beta}\mathsf{Cov}\left[h'(t), h''(t)\right] + \mathbb{E}\left[h'''(t)\right].$$

Next, we have

$$\frac{d}{dt}\mathbb{E}\left[h'(t)\right]^2 = 2\mathbb{E}\left[h'(t)\right]\cdot\frac{d}{dt}\mathbb{E}\left[h'(t)\right] = 2\mathbb{E}\left[h'(t)\right]\left(\frac{1}{\beta}\mathsf{Var}\left[h'(t)\right] + \mathbb{E}\left[h''(t)\right]\right)$$

and

$$\begin{aligned}
&\frac{d}{dt}\mathbb{E}\left[h'(t)^2\right] \\
&= \frac{\left(\sum_{i=1}^m \lambda_i'(t)h_i'(t)^2 + 2h_i'(t)h_i''(t)\lambda_i(t)\right)\left(\sum_{i=1}^m \lambda_i(t)\right) - \frac{1}{\beta}\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)\right)\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)^2\right)}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2} \\
&= \frac{\left(\sum_{i=1}^m \lambda_i'(t)h_i'(t)^2 + 2h_i'(t)h_i''(t)\lambda_i(t)\right)}{\sum_{i=1}^m \lambda_i(t)} - \frac{1}{\beta}\cdot\frac{\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)\right)\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)^2\right)}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2} \\
&= \frac{\sum_{i=1}^m \lambda_i(t)\left(\frac{h_i'(t)^3}{\beta} + 2h_i'(t)h_i''(t)\right)}{\sum_{i=1}^m \lambda_i(t)} - \frac{1}{\beta}\cdot\frac{\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)\right)\left(\sum_{i=1}^m \lambda_i(t)h_i'(t)^2\right)}{\left(\sum_{i=1}^m \lambda_i(t)\right)^2} \\
&= \frac{1}{\beta}\mathsf{Cov}\left[h'(t), h'(t)^2\right] + 2\mathbb{E}\left[h'(t)h''(t)\right].
\end{aligned}$$

Combining everything gives us

$$\begin{aligned}
&\mathsf{lse}_\beta'''(h(t)) \\
&= \frac{1}{\beta}\left(\frac{1}{\beta}\mathsf{Cov}\left[h'(t), h'(t)^2\right] + 2\mathbb{E}\left[h'(t)h''(t)\right] - 2\mathbb{E}\left[h'(t)\right]\left(\frac{1}{\beta}\mathsf{Var}\left[h'(t)\right] + \mathbb{E}\left[h''(t)\right]\right)\right) \\
&\quad + \frac{1}{\beta}\mathsf{Cov}\left[h'(t), h''(t)\right] + \mathbb{E}\left[h'''(t)\right] \\
&= \frac{1}{\beta^2}\mathsf{Cov}\left[h'(t), h'(t)^2\right] - \frac{2}{\beta^2}\mathbb{E}\left[h'(t)\right]\mathsf{Var}\left[h'(t)\right] + \frac{3}{\beta}\mathsf{Cov}\left[h'(t), h''(t)\right] + \mathbb{E}\left[h'''(t)\right].
\end{aligned}$$

We first analyze the terms that only depend on $h'(t)$. To do so, we use Lemma B.4 to write

$$\left|\mathsf{Cov}\left[h'(t), h'(t)^2\right]\right| \le \sqrt{\mathsf{Var}\left[h'(t)\right]}\sqrt{\mathsf{Var}\left[h'(t)^2\right]} \le 2\left\|\boldsymbol{d}\right\|\mathsf{Var}\left[h'(t)\right].$$

Now, we have

$$\begin{aligned}
&\frac{1}{\beta^2}\left|\mathsf{Cov}\left[h'(t), h'(t)^2\right] - 2\mathbb{E}\left[h'(t)\right]\mathsf{Var}\left[h'(t)\right]\right| \\
&\le \frac{1}{\beta^2}\left|\mathsf{Cov}\left[h'(t), h'(t)^2\right]\right| + \frac{2}{\beta^2}\left|\mathbb{E}\left[h'(t)\right]\mathsf{Var}\left[h'(t)\right]\right| \\
&\le \frac{4}{\beta^2}\left\|\boldsymbol{d}\right\|\mathsf{Var}\left[h'(t)\right] \le \frac{4}{\beta}\left\|\boldsymbol{d}\right\|\mathsf{lse}_\beta''(h(t)).
\end{aligned}$$

Next, we take care of the remaining terms. We have

$$\begin{aligned}\frac{3}{\beta}\left|\mathsf{Cov}\left[h'(t),h''(t)\right]\right|+\left|\mathbb{E}\left[h'''(t)\right]\right| &\le \frac{6}{\beta}\left(\max_i h_i'(t)\right)\mathbb{E}\left[\left|h''(t)-\mathbb{E}\left[h''(t)\right]\right|\right]+\left|\mathbb{E}\left[h'''(t)\right]\right| \\ &\le \frac{12}{\beta}\left\|\boldsymbol{d}\right\|\mathsf{lse}_\beta''(h(t))+\mathbb{E}\left[\left|h'''(t)\right|\right] \\ &\le \frac{12}{\beta}\left\|\boldsymbol{d}\right\|\mathsf{lse}_\beta''(h(t))+\nu\left\|\boldsymbol{d}\right\|\mathbb{E}\left[h''(t)\right] \\ &\le \left(\frac{12}{\beta}+\nu\right)\left\|\boldsymbol{d}\right\|\mathsf{lse}_\beta''(h(t)),\end{aligned}$$

where the penultimate line follows from Lemma B.7. Combining these conclusions yields

$$\left|\mathsf{lse}_\beta'''(h(t))\right| \le \left(\frac{16}{\beta}+\nu\right)\left\|\boldsymbol{d}\right\|\mathsf{lse}_\beta''(h(t)),$$

completing the proof of Lemma B.3. □

B.2. Smoothness and quasi-self concordance of the modified objective

The main result of this subsection is Lemma B.5.

Lemma B.5. *Let* $\mathbf{W}$ *be such that for all* $\boldsymbol{z}\in\mathbb{R}^d$*, we have* $\left\|\mathbf{A}\boldsymbol{z}\right\|_{\mathcal{G}_\infty} \le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{z}\right\|_2$*. For all* $\boldsymbol{x},\boldsymbol{z}\in\mathbb{R}^d$ *and* $t\in\mathbb{R}$*, we have*

$$\left(\frac{d}{dt}\right)^2\widetilde{f}_{\beta,\delta}(\boldsymbol{x}+t\boldsymbol{z}) \le \left(\frac{1}{\delta}+\frac{1}{\beta}\right)\left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{z}\right\|_2^2 \qquad \textit{(smoothness)}$$

$$\left|\left(\frac{d}{dt}\right)^3\widetilde{f}_{\beta,\delta}(\boldsymbol{x}+t\boldsymbol{z})\right| \le \left(\frac{16}{\delta}+\frac{3}{\beta}\right)\left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{z}\right\|_2\left(\frac{d}{dt}\right)^2\widetilde{f}_{\beta,\delta}(\boldsymbol{x}+t\boldsymbol{z}) \quad \textit{(quasi self concordance)}.$$

Our goal in the rest of this section is to prove Lemma B.5.

We begin with defining $h_i(t)$ as (absorb the $\delta,\boldsymbol{y},\boldsymbol{d}$ parameters into the definition of h_i)

$$h_i(t) \coloneqq \sqrt{\delta^2+\left\|\boldsymbol{y}_{S_i}+t\boldsymbol{d}_{S_i}\right\|_2^2}.$$

Let $h(t)$ denote the vector whose ith entry is $h_i(t)$. Then, observe that

$$\mathsf{lse}_\beta(h(t)) = \beta\log\left(\sum_{i=1}^m\exp\left(\frac{h_i(t)}{\beta}\right)\right) = \beta\log\left(\sum_{i=1}^m\exp\left(\frac{\sqrt{\delta^2+\left\|\boldsymbol{y}_{S_i}+t\boldsymbol{d}_{S_i}\right\|_2^2}}{\beta}\right)\right).$$

It is easy to see that every one-dimensional restriction of $\widetilde{f}_{\beta,\delta}$ can be obtained by an affine transformation of $\mathsf{lse}_\beta(h(t))$ after appropriate choices of $\boldsymbol{y},\boldsymbol{d}\in\mathbb{R}^m$. Hence, we first analyze $\mathsf{lse}_\beta(h(t))$ for all $\boldsymbol{y},\boldsymbol{d}\in\mathbb{R}^m$.

We begin with proving the smoothness of $\mathsf{lse}_\beta(h(t))$ with respect to $\left\|\cdot\right\|_{\mathcal{G}_\infty}$.

Lemma B.6. *For all $\boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$ and all $t \in \mathbb{R}$, we have*

$$\left(\frac{d}{dt}\right)^2 \mathsf{lse}_\beta(h(t)) \leq \left(\frac{1}{\delta} + \frac{1}{\beta}\right) \|\boldsymbol{d}\|_{\mathcal{G}_\infty}^2 .$$

Proof of Lemma B.6. By direct calculation, it is easy to see that

$$\begin{aligned} h_i'(t) &= \frac{\langle \boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}, \boldsymbol{d}_{S_i}\rangle}{h_i(t)} \\ h_i''(t) &= \frac{\|\boldsymbol{d}_{S_i}\|_2^2 h_i(t) - h_i'(t)^2 h_i(t)}{h_i(t)^2} = \frac{\|\boldsymbol{d}_{S_i}\|_2^2 - h_i'(t)^2}{h_i(t)}. \end{aligned} \tag{B.1}$$

We plug this into the result of Lemma B.2 and get

$$\begin{aligned} \mathsf{lse}_\beta''(h(t)) &\leq \frac{1}{\beta} \max_i h_i'(t)^2 + \max_i h_i''(t) \\ &= \frac{1}{\beta} \max_i \left(\frac{\langle \boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}, \boldsymbol{d}_{S_i}\rangle}{\sqrt{\delta^2 + \|\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}\|_2^2}} \right)^2 + \max_i \frac{\|\boldsymbol{d}_{S_i}\|_2^2 - h_i'(t)^2}{\sqrt{\delta^2 + \|\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}\|_2^2}} \\ &\leq \frac{1}{\beta} \max_i \|\boldsymbol{d}_{S_i}\|_2^2 + \frac{1}{\delta} \max_i \|\boldsymbol{d}_{S_i}\|_2^2 = \left(\frac{1}{\beta} + \frac{1}{\delta}\right) \|\boldsymbol{d}\|_{\mathcal{G}_\infty}^2 , \end{aligned}$$

completing the proof of Lemma B.6. □

Our next task is to show that $\mathsf{lse}_\beta(h(t))$ is $O(1/\beta + 1/\delta)$-quasi self concordant in $\|\cdot\|_{\mathcal{G}_\infty}$. To do so, we will appeal to Lemma B.3. To be able to do this, we first have to prove the quasi self concordance of each component function in $\mathsf{lse}_\beta(h(t))$.

Lemma B.7. *For all $\boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$ and all $t \in \mathbb{R}$, we have*

$$\left| \left(\frac{d}{dt}\right)^3 \sqrt{\delta^2 + \|\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}\|_2^2} \right| \leq \frac{3}{\delta} \|\boldsymbol{d}_{S_i}\|_2 \left(\left(\frac{d}{dt}\right)^2 \sqrt{\delta^2 + \|\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}\|_2^2} \right).$$

Proof of Lemma B.7. Although a similar fact appears in [OB20, Section 2.1.2], it is not in the exact form we need. So, we prove the required statement here.

Recycling the computation from (B.1), recall

$$h_i''(t) = \frac{\|\boldsymbol{d}_{S_i}\|_2^2 - h_i'(t)^2}{h_i(t)},$$

which gives

$$h_i'''(t) = \frac{-2h_i'(t)h_i''(t)h_i(t) - h_i'(t)(h_i(t)h_i''(t))}{h_i(t)^2} = -\frac{3h_i'(t)h_i''(t)}{h_i(t)}.$$

Finally, again recalling (B.1), notice that

$$\left|\frac{h_i'(t)}{h_i(t)}\right| = \left|\frac{\langle \boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}, \boldsymbol{d}_{S_i}\rangle}{h_i(t)^2}\right| = \left| \left\langle \frac{\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}}{\sqrt{\delta^2 + \|\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}\|_2^2}}, \frac{\boldsymbol{d}_{S_i}}{\sqrt{\delta^2 + \|\boldsymbol{y}_{S_i} + t\boldsymbol{d}_{S_i}\|_2^2}} \right\rangle \right| \leq \frac{\|\boldsymbol{d}_{S_i}\|_2}{\delta}.$$

Combining everything completes the proof of Lemma B.7. □

We are now ready to prove the quasi self concordance of $\mathsf{lse}_\beta(h(t))$ in $\|\cdot\|_{\mathcal{G}_\infty}$.

Lemma B.8. *For all $\boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$ and $t \in \mathbb{R}$, we have*

$$\left|\left(\frac{d}{dt}\right)^3 \mathsf{lse}_\beta(h(t))\right| \leq \left(\frac{16}{\beta} + \frac{3}{\delta}\right) \|\boldsymbol{d}\|_{\mathcal{G}_\infty} \left(\frac{d}{dt}\right)^2 \mathsf{lse}_\beta(h(t)).$$

Proof of Lemma B.8. In the statement of Lemma B.3, let $\|\cdot\| = \|\cdot\|_{\mathcal{G}_\infty}$. By the definition of $\|\cdot\|_{\mathcal{G}_\infty}$ and h_i, we have for all i and t that $h_i'(t) \leq \|\boldsymbol{d}\|_{\mathcal{G}_\infty}$. Additionally, from Lemma B.7, we have that the $h_i(t)$ are $3/\delta$-quasi self concordant in the norm $\|\boldsymbol{d}\|_{\mathcal{G}_\infty}$ for all i. Lemma B.8 now follows immediately from Lemma B.3. □

Finally, we can prove Lemma B.5.

Proof of Lemma B.5. By the conclusion of Lemma B.6, we know that for all $\boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$ and $t \in \mathbb{R}$ that

$$\left(\frac{d}{dt}\right)^2 \mathsf{lse}_\beta(h(t)) \leq \left(\frac{1}{\delta} + \frac{1}{\beta}\right) \|\boldsymbol{z}\|_{\mathcal{G}_\infty}^2 .$$

Let $\boldsymbol{y} = \mathbf{A}\boldsymbol{x} - \boldsymbol{b}$ for some $\boldsymbol{x}$ and $\boldsymbol{d} = \mathbf{A}\boldsymbol{z}$ for some $\boldsymbol{z}$. Let

$$g(\boldsymbol{y}) \coloneqq \beta \log\left(\sum_{i=1}^m \exp\left(\frac{\sqrt{\delta^2 + \|\boldsymbol{y}_{S_i}\|_2^2} - \delta}{\beta}\right)\right).$$

Then,

$$\left(\frac{d}{dt}\right)^2 \widetilde{f}_{\beta,\delta}(\boldsymbol{x} + t\boldsymbol{z}) = \left(\frac{d}{dt}\right)^2 g(\mathbf{A}\boldsymbol{x} - \boldsymbol{b} + t\mathbf{A}\boldsymbol{z}) \leq \left(\frac{1}{\delta} + \frac{1}{\beta}\right) \|\mathbf{A}\boldsymbol{z}\|_{\mathcal{G}_\infty}^2 .$$

With the exact same reasoning applied to the conclusion of Lemma B.8, we also see that

$$\left|\left(\frac{d}{dt}\right)^3 \widetilde{f}_{\beta,\delta}(\boldsymbol{x} + t\boldsymbol{z})\right| \leq \left(\frac{16}{\delta} + \frac{3}{\beta}\right) \|\mathbf{A}\boldsymbol{z}\|_{\mathcal{G}_\infty} \left(\frac{d}{dt}\right)^2 \widetilde{f}_{\beta,\delta}(\boldsymbol{x} + t\boldsymbol{z}).$$

The conclusion of Lemma B.5 then follows from remembering that we have $\mathbf{W}$ such that for all $\boldsymbol{z} \in \mathbb{R}^d$, $\|\mathbf{A}\boldsymbol{z}\|_{\mathcal{G}_\infty} \leq \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{z}\right\|_2$. □

B.3. Analysis of Algorithm 1

In this subsection, we use the calculus facts from the previous two subsections to analyze Algorithm 1. The outline of this proof follows that of [JLS22, Theorem 2], which in turn builds up to using the proof used in [CJJJLST20, Corollary 12]. The main idea is to define the algorithm based on the norm given by the right choice of positive semidefinite $\mathbf{M}$.

In the rest of this section, let $\mathbf{W}$ be factor-2 block Lewis weight overestimates for $[\mathbf{A}|\boldsymbol{b}]$. As in Line 2 of Algorithm 1 and from the corresponding guarantee given in [MO25, Lemmas 5.6, 5.8], this

means that within $2\log m$ linear system solves in $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for diagonal $\mathbf{D}$, we can find $\mathbf{W}$ such that for all $\boldsymbol{x} \in \mathbb{R}^d$ and $c \in \mathbb{R}$ we have

$$\|\mathbf{A}\boldsymbol{x} - c\boldsymbol{b}\|_{\mathcal{G}_\infty} \le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x} - c\mathbf{W}^{1/2}\boldsymbol{b}\right\|_2 \le \sqrt{2(\mathsf{rank}\,(\mathbf{A})+1)}\,\|\mathbf{A}\boldsymbol{x} - c\boldsymbol{b}\|_{\mathcal{G}_\infty}.$$

Note that choosing $c = 1$ yields our original objective on either side of the above inequality. Motivated by the above, it is natural to use the norm given by $\mathbf{M} \coloneqq \mathbf{A}^\top\mathbf{W}\mathbf{A}$ to give the geometry for the ball optimization oracle and for the analysis. Additionally, without loss of generality and for the sake of the analysis, let us rescale the problem so that

$$1 = \mathsf{OPT} \coloneqq \|\mathbf{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_{\mathcal{G}_\infty}.$$

Additionally, assume without loss of generality that $\mathsf{rank}\,(\mathbf{A}) = d$.

We begin with Lemma B.9, which bounds our initial suboptimality in $\widetilde{f}$ and in $\|\cdot\|_{\mathbf{M}}$.

Lemma B.9. *Let* $\widetilde{\boldsymbol{x}}_{\beta,\delta} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \widetilde{f}_{\beta,\delta}(\boldsymbol{x})$. *Then,*

$$\begin{aligned}\|\widetilde{\boldsymbol{x}}_{\beta,\delta} - \boldsymbol{x}_0\|_{\mathbf{M}} &\le (2 + 2(\beta\log m + \delta))\sqrt{2(d+1)} \\ \widetilde{f}_{\beta,\delta}(\boldsymbol{x}_0) - \widetilde{f}_{\beta,\delta}(\widetilde{\boldsymbol{x}}_{\beta,\delta}) &\le \sqrt{2(d+1)} - 1 + 2(\beta\log m + \delta)\end{aligned}.$$

Proof of Lemma B.9. It is easy to check that

$$\boldsymbol{x}_0 \coloneqq \left(\mathbf{A}^\top\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{W}\boldsymbol{b} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x} - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2.$$

By Lemma B.1, for all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\left|\widetilde{f}_{\beta,\delta}(\boldsymbol{x}) - \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_{\mathcal{G}_\infty}\right| \le \beta\log m + \delta,$$

implying

$$\left|\|\mathbf{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_{\mathcal{G}_\infty} - \widetilde{f}_{\beta,\delta}(\widetilde{\boldsymbol{x}}_{\beta,\delta})\right| \le \beta\log m + \delta.$$

This easily implies

$$1 \le \|\mathbf{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_{\mathcal{G}_\infty} \le \|\mathbf{A}\boldsymbol{x}_0 - \boldsymbol{b}\|_{\mathcal{G}_\infty} \le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0 - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2$$

and

$$\frac{\left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0 - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2}{\sqrt{2(d+1)}} \le \frac{\left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}^\star - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2}{\sqrt{2(d+1)}} \le \|\mathbf{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_{\mathcal{G}_\infty} = 1.$$

Combining these gives

$$1 \le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0 - \mathbf{W}^{1/2}\boldsymbol{b}\right\|_2 \le \sqrt{2(d+1)}.$$

Additionally,

$$\begin{aligned}\left\|\mathbf{W}^{1/2}\mathbf{A}\widetilde{\boldsymbol{x}}_{\beta,\delta}-\mathbf{W}^{1/2}\boldsymbol{b}\right\|_2 &\le \sqrt{2(d+1)}\left\|\mathbf{A}\widetilde{\boldsymbol{x}}_{\beta,\delta}-\boldsymbol{b}\right\|_{\mathcal{G}_\infty}\\ &\le \sqrt{2(d+1)}\left(\widetilde{f}_{\beta,\delta}(\widetilde{\boldsymbol{x}}_{\beta,\delta})+\beta\log m+\delta\right)\\ &\le \sqrt{2(d+1)}\left(\left\|\mathbf{A}\boldsymbol{x}^\star-\boldsymbol{b}\right\|_{\mathcal{G}_\infty}+2(\beta\log m+\delta)\right)\\ &= \sqrt{2(d+1)}(1+2(\beta\log m+\delta)).\end{aligned}$$

Then,

$$\begin{aligned}\left\|\widetilde{\boldsymbol{x}}-\boldsymbol{x}_0\right\|_{\mathbf{M}} &= \left\|\left(\mathbf{W}^{1/2}\mathbf{A}\widetilde{\boldsymbol{x}}_{\beta,\delta}-\mathbf{W}^{1/2}\boldsymbol{b}\right)-\left(\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0-\mathbf{W}^{1/2}\boldsymbol{b}\right)\right\|_2\\ &\le \left\|\mathbf{W}^{1/2}\mathbf{A}\widetilde{\boldsymbol{x}}_{\beta,\delta}-\mathbf{W}^{1/2}\boldsymbol{b}\right\|_2+\left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0-\mathbf{W}^{1/2}\boldsymbol{b}\right\|_2\\ &\le (2+2(\beta\log m+\delta))\sqrt{2(d+1)},\end{aligned}$$

and

$$\begin{aligned}\widetilde{f}_{\beta,\delta}(\boldsymbol{x}_0)-\widetilde{f}_{\beta,\delta}(\widetilde{\boldsymbol{x}}_{\beta,\delta}) &\le \left\|\mathbf{A}\boldsymbol{x}_0-\boldsymbol{b}\right\|_{\mathcal{G}_\infty}-\left\|\mathbf{A}\boldsymbol{x}^\star-\boldsymbol{b}\right\|_{\mathcal{G}_\infty}+2(\beta\log m+\delta)\\ &\le \left\|\mathbf{W}^{1/2}\mathbf{A}\boldsymbol{x}_0-\mathbf{W}^{1/2}\boldsymbol{b}\right\|_2-\mathsf{OPT}+2(\beta\log m+\delta)\\ &\le \sqrt{2(d+1)}-1+2(\beta\log m+\delta).\end{aligned}$$

This completes the proof of Lemma B.9. □

We are now ready to prove Theorem 1.

Proof of Theorem 1. Algorithm 1 optimizes the regularization of $\widetilde{f}$ given by

$$\widehat{f}(\boldsymbol{x}) \coloneqq \widetilde{f}_{\beta,\delta}(\boldsymbol{x})+\frac{\varepsilon}{110R^2}\left\|\mathbf{W}^{1/2}\mathbf{A}(\boldsymbol{x}-\boldsymbol{x}_0)\right\|_2^2,$$

where R is such that $\left\|\boldsymbol{x}_0-\widetilde{\boldsymbol{x}}_{\beta,\delta}\right\|_{\mathbf{M}} \le R$. Let $\widehat{\boldsymbol{x}} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d}\widehat{f}(\boldsymbol{x})$. Using [CJJJLST20, Proof of Corollary 12], we know that for every iterate $\boldsymbol{x}$ of Algorithm 1,

$$\left|\widehat{f}(\boldsymbol{x})-\widetilde{f}_{\beta,\delta}(\boldsymbol{x})\right| \le \frac{\varepsilon}{4}.$$

We now choose $\beta=\varepsilon/(4\log m)$ and $\delta=\varepsilon/4$, so that $\widetilde{f}_{\beta,\delta}$ approximates f up to error $\varepsilon/2$ on every point. Using Lemma B.9, this gives $R=(2+\varepsilon)\sqrt{2(d+1)}$. It is therefore sufficient to optimize $\widehat{f}$ up to $\varepsilon/4$ additive error.

Next, using Lemma B.5 and [CJJJLST20, Lemmas 11, 43], we have that $\widehat{f}$ is $(1/\nu,e)$-Hessian stable in $\|\cdot\|_{\mathbf{M}}$ for $\nu=\Omega(1/(\varepsilon\log m))$. Finally, using [CJJJLST20, Theorems 6, 9], we get that Algorithm 1 has a Newton iteration complexity of

$$O\left(\left(\frac{(1+\varepsilon)\sqrt{d}\log m}{\varepsilon}\right)^{2/3}\log\left(\frac{\sqrt{d}+\varepsilon}{\varepsilon}\right)\left(\log\left(\frac{(\log m/\varepsilon)d(1+(1+\varepsilon)\sqrt{d}\log m/\varepsilon)}{\varepsilon}\right)\right)^3\right)$$

$$= O\left(\frac{d^{1/3}}{\varepsilon^{2/3}} \log\left(\frac{d \log m}{\varepsilon}\right)^{14/3}\right),$$

as promised.

It remains to determine the form of the Newton steps. For this, it is sufficient to understand the Hessian of $\widehat{f}$. A straightforward calculation shows that it is of the form $\mathbf{A}^\top \mathbf{B} \mathbf{A}$ where $\mathbf{B}$ is a block-diagonal matrix where each block has size $|S_i| \times |S_i|$. Thus, each Newton step solves a linear system of the form $\mathbf{A}^\top \mathbf{B} \mathbf{A} \boldsymbol{z} = \boldsymbol{v}$.

Combining this with the iteration complexity guarantee to find $\mathbf{W}$ (arising from [MO25, Lemmas 5.6, 5.8]) completes the proof of Theorem 1. □