Faster Randomized Infeasible Interior Point Methods for Tall/Wide Linear Programs

Agniva Chowdhury
Department of Statistics
Purdue University
West Lafayette, IN, USA
chowdhu5@purdue.edu

Palma London
ORIE Department
Cornell University
Ithaca, NY, USA
plondon@cornell.edu

Haim Avron
School of Mathematical Sciences
Tel Aviv University
Tel Aviv, Israel
haimav@tauex.tau.ac.il

Petros Drineas
Department of Computer Science
Purdue University
West Lafayette, IN, USA
pdrineas@purdue.edu

Abstract

Linear programming (LP) is used in many machine learning applications, such as ℓ_1-regularized SVMs, basis pursuit, nonnegative matrix factorization, etc. Interior Point Methods (IPMs) are one of the most popular methods to solve LPs both in theory and in practice. Their underlying complexity is dominated by the cost of solving a system of linear equations at each iteration. In this paper, we consider *infeasible* IPMs for the special case where the number of variables is much larger than the number of constraints (i.e., wide), or vice-versa (i.e., tall) by taking the dual. Using tools from Randomized Linear Algebra, we present a preconditioning technique that, when combined with the Conjugate Gradient iterative solver, provably guarantees that infeasible IPM algorithms (suitably modified to account for the error incurred by the approximate solver), converge to a feasible, approximately optimal solution, without increasing their iteration complexity. Our empirical evaluations verify our theoretical results on both real and synthetic data.

1 Introduction

Linear programming (LP) is one of the most useful tools available to theoreticians and practitioners throughout science and engineering. In Machine Learning, LP appears in numerous settings, including ℓ_1-regularized SVMs [57], basis pursuit (BP) [54], sparse inverse covariance matrix estimation (SICE) [55], the nonnegative matrix factorization (NMF) [45], MAP inference [37], etc. Not surprisingly, designing and analyzing LP algorithms is a topic of paramount importance in computer science and applied mathematics.

One of the most successful paradigms for solving LPs is the family of Interior Point Methods (IPMs), pioneered by Karmarkar in the mid 1980s [25]. Path-following IPMs and, in particular, long-step path following IPMs, are among the most practical approaches for solving linear programs. Consider the standard form of the primal LP problem:

$$\min \mathbf{c}^\mathsf{T}\mathbf{x}\,, \text{ subject to } \mathbf{A}\mathbf{x} = \mathbf{b}\,, \mathbf{x} \geq \mathbf{0}\,, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$ are the inputs, and $\mathbf{x} \in \mathbb{R}^n$ is the vector of the primal variables. The associated dual problem is

$$\max \mathbf{b}^\mathsf{T}\mathbf{y}\,, \text{ subject to } \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\,, \mathbf{s} \geq \mathbf{0}\,, \tag{2}$$

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

where $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{s} \in \mathbb{R}^n$ are the vectors of the dual and slack variables respectively. Triplets $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ that uphold both (1) and (2) are called *primal-dual solutions*. Path-following IPMs typically converge towards a primal-dual solution by operating as follows: given the current iterate $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$, they compute the Newton search direction $(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ and update the current iterate by following a step towards the search direction. To compute the search direction, one standard approach [41] involves solving the *normal equations*[1]:

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\Delta\mathbf{y} = \mathbf{p}. \tag{3}$$

Here, $\mathbf{D} = \mathbf{X}^{1/2}\mathbf{S}^{-1/2}$ is a diagonal matrix, $\mathbf{X}, \mathbf{S} \in \mathbb{R}^{n\times n}$ are diagonal matrices whose i-th diagonal entries are equal to $\mathbf{x}_i$ and $\mathbf{s}_i$, respectively, and $\mathbf{p} \in \mathbb{R}^m$ is a vector whose exact definition is given in eqn. (16)[2]. Given $\Delta\mathbf{y}$, computing $\Delta\mathbf{s}$ and $\Delta\mathbf{x}$ only involves matrix-vector products.

The core computational bottleneck in IPMs is the need to solve the linear system of eqn. (3) at each iteration. This leads to two key challenges: first, for high-dimensional matrices $\mathbf{A}$, solving the linear system is computationally prohibitive. Most implementations of IPMs use a *direct solver*; see Chapter 6 of [41]. However, if $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$ is large and dense, direct solvers are computationally impractical. If $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$ is sparse, specialized direct solvers have been developed, but these do not apply to many LP problems arising in machine learning applications due to irregular sparsity patterns. Second, an alternative to direct solvers is the use of iterative solvers, but the situation is further complicated since $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$ is typically ill-conditioned. Indeed, as IPM algorithms approach the optimal primal-dual solution, the diagonal matrix $\mathbf{D}$ is ill-conditioned, which also results in the matrix $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$ being ill-conditioned. Additionally, using approximate solutions for the linear system of eqn. (3) causes certain invariants, which are crucial for guaranteeing the convergence of IPMs, to be violated; see Section 1.1 for details.

In this paper, we address the aforementioned challenges, for the special case where $m \ll n$, i.e., the number of constraints is much smaller than the number of variables; see Appendix A for a generalization. This is a common setting in ML applications of LP solvers, since ℓ_1-SVMs and basis pursuit problems often exhibit such structure when the number of available features (n) is larger than the number of objects (m). This setting has been of interest in recent work on LPs [17, 4, 31]. For simplicity of exposition, we also assume that the constraint matrix $\mathbf{A}$ has full rank, equal to m. First, we propose and analyze a preconditioned Conjugate Gradient (CG) iterative solver for the normal equations of eqn. (3), using matrix sketching constructions from the Randomized Linear Algebra (RLA) literature. We develop a preconditioner for $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$ using matrix sketching which allows us to prove strong convergence guarantees for the *residual* of CG solvers. Second, building upon the work of [39], we propose and analyze a provably accurate long-step *infeasible* IPM algorithm. The proposed IPM solves the normal equations using iterative solvers. In this paper, for brevity and clarity, we primarily focus our description and analysis on the CG iterative solver. We note that a non-trivial concern is that the use of iterative solvers and matrix sketching tools implies that the normal equations at each iteration will be solved only approximately. In our proposed IPM, we develop a novel way to *correct* for the error induced by the approximate solution in order to guarantee convergence. Importantly, this correction step is relatively computationally light, unlike a similar step proposed in [39]. Third, we empirically show that our algorithm performs well in practice. We consider solving LPs that arise from ℓ_1-regularized SVMs and test them on a variety of synthetic and real datasets. Several extensions of our work are discussed in Appendix A.

1.1 Our contributions

Our point of departure in this work is the introduction of preconditioned, iterative solvers for solving eqn. (3). Preconditioning is used to address the ill-conditioning of the matrix $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$. Iterative solvers allow the computation of approximate solutions using only matrix-vector products while avoiding matrix inversion, Cholesky or LU factorizations, etc. A preconditioned formulation of eqn. (3) is:

$$\mathbf{Q}^{-1}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\Delta\mathbf{y} = \mathbf{Q}^{-1}\mathbf{p}, \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is the preconditioning matrix; $\mathbf{Q}$ should be easily invertible (see [3, 22] for background). An alternative yet equivalent formulation of eqn. (4), which is more amenable to

[1] Another widely used approach is to solve the augmented system [41] which is less relevant for this paper.

[2] The superscript k in eqn. (16) simply indicates iteration count and is omitted here for notational simplicity.

theoretical analysis, is

$$\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\mathbf{z} = \mathbf{Q}^{-1/2}\mathbf{p}, \tag{5}$$

where $\mathbf{z} \in \mathbb{R}^m$ is a vector such that $\Delta\mathbf{y} = \mathbf{Q}^{-1/2}\mathbf{z}$. Note that the matrix in the left-hand side of the above equation is always symmetric, which is not necessarily the case for eqn. (4). We do emphasize that one can use eqn. (4) in the actual implementation of the preconditioned solver; eqn. (5) is much more useful in theoretical analyses.

Recall that we focus on the special case where $\mathbf{A} \in \mathbb{R}^{m\times n}$ has $m \ll n$, i.e., it is a short-and-fat matrix. Our first contribution starts with the design and analysis of a preconditioner for the Conjugate Gradient solver that satisfies, with high probability,

$$\frac{2}{2+\zeta} \leq \sigma^2_{\min}(\mathbf{Q}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}) \leq \sigma^2_{\max}(\mathbf{Q}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}) \leq \frac{2}{2-\zeta}, \tag{6}$$

for some error parameter $\zeta \in [0, 1]$. In the above, $\sigma_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$ correspond to the smallest and largest singular value of the matrix in parentheses. The above condition says that the preconditioner effectively reduces the condition number of $\mathbf{A}\mathbf{D}$ to a constant. We note that the particular form of the lower and upper bounds in eqn. (6) was chosen to simplify our derivations. RLA matrix-sketching techniques allow us to construct preconditioners for all short-and-fat matrices that satisfy the above inequality *and* can be inverted efficiently. Such constructions go back to the work of [2]; see Section 2 for details on the construction of $\mathbf{Q}$ and its inverse. Importantly, given such a preconditioner, we then prove that the resulting CG iterative solver satisfies

$$\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t - \mathbf{Q}^{-1/2}\mathbf{p}\|_2 \leq \zeta^t\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2. \tag{7}$$

Here $\tilde{\mathbf{z}}^t$ is the approximate solution returned by the CG iterative solver after t iterations. In words, the above inequality states that the *residual* achieved after t iterations of the CG iterative solver drops exponentially fast. To the best of our knowledge, this result is not known in the CG literature: indeed, it is actually well-known that the residual of CG may oscillate [21], even in cases where the energy norm of the solution error decreases monotonically. However, we prove that if the preconditioner is sufficiently good, i.e., it satisfies the constraint of eqn. (6), then the residual decreases as well.

Our second contribution is the analysis of a novel variant of a long-step *infeasible* IPM algorithm proposed by [39]. Recall that such algorithms can, in general, start with an initial point that is not necessarily feasible, but does need to satisfy some, more relaxed, constraints. Following the lines of [56, 39], let $\mathcal{S}$ be the set of feasible and optimal solutions of the form $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{s}^*)$ for the primal and dual problems of eqns. (1) and (2) and assume that $\mathcal{S}$ is not empty. Then, long-step infeasible IPMs can start with any initial point $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ that satisfies $(\mathbf{x}^0, \mathbf{s}^0) > 0$ *and* $(\mathbf{x}^0, \mathbf{s}^0) \geq (\mathbf{x}^*, \mathbf{s}^*)$, for some feasible and optimal solution $(\mathbf{x}^*, \mathbf{s}^*) \in \mathcal{S}$. In words, the starting primal and slack variables must be strictly positive *and* larger (element-wise) when compared to some feasible, optimal primal-dual solution. See Chapter 6 of [52] for a discussion regarding why such choices of starting points are relevant to computational practice and can be identified more efficiently than feasible points.

The flexibility of infeasible IPMs comes at a cost: long-step *feasible* IPMs converge in $\mathcal{O}(n \log 1/\epsilon)$ iterations, while long-step *infeasible* IPMs need $\mathcal{O}(n^2 \log 1/\epsilon)$ iterations to converge [56, 39] (Here ϵ is the accuracy of the approximate LP solution returned by the IPM; see Algorithm 2 for the exact definition.). Let

$$\mathbf{A}\mathbf{x}^0 - \mathbf{b} = \mathbf{r}^0_p, \tag{8}$$

$$\mathbf{A}^\mathsf{T}\mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c} = \mathbf{r}^0_d, \tag{9}$$

where $\mathbf{r}^0_p \in \mathbb{R}^n$ and $\mathbf{r}^0_d \in \mathbb{R}^m$ are the *primal* and *dual* residuals, respectively, and characterize how far the initial point is from being feasible. As long-step infeasible IPM algorithms iterate and update the primal and dual solutions, the residuals are updated as well. Let $\mathbf{r}^k = (\mathbf{r}^k_p, \mathbf{r}^k_d) \in \mathbb{R}^{n+m}$ be the primal and dual residual at the k-th iteration: it is well-known that the convergence analysis of infeasible long-step IPMs critically depends on $\mathbf{r}^k$ lying on the line segment between 0 and $\mathbf{r}^0$. Unfortunately, using approximate solvers (such as the CG solver proposed above) for the normal equations violates this invariant. [39] proposed a simple solution to fix this problem by adding a perturbation vector $\mathbf{v}$ to the current primal-dual solution that guarantees that the invariant is satisfied. Again, we use RLA matrix sketching principles to propose an efficient construction for $\mathbf{v}$ that provably satisfies the invariant. Next, we combine the above two primitives to prove that Algorithm 2 in Section 3 satisfies the following theorem.

Theorem 1 *Let $0 \leq \epsilon \leq 1$ be an accuracy parameter. Consider the long-step infeasible IPM Algorithm 2 (Section 3) that solves eqn. (5) using the CG solver of Algorithm 1 (Section 2). Assume that the CG iterative solver runs with accuracy parameter $\zeta = 1/2$ and iteration count $t = \mathcal{O}(\log n)$. Then, with probability at least 0.9, the long-step infeasible IPM converges after $\mathcal{O}(n^2 \log 1/\epsilon)$ iterations.*

We note that the 0.9 success probability above is for simplicity of exposition and can be easily amplified using standard techniques. Also, at each iteration of our infeasible long-step IPM algorithm, the running time is $\mathcal{O}((\mathsf{nnz}(\mathbf{A}) + m^3) \log n)$, ignoring constant terms. See Section 3 for a detailed discussion of the overall running time.

Our empirical evaluation demonstrates that our algorithm requires an order of magnitude much fewer inner CG iterations than a standard IPM using CG, while producing a comparably accurate solution (see Section 4).

1.2 Prior Work

There is a large body of literature on solving LPs using IPMs. We only review literature that is immediately relevant to our work. Recall that we solve the normal equations inexactly at each iteration, and develop a way to *correct* for the error incurred. We also focus on IPMs that can use a sufficiently positive, infeasible initial point (see Section 1.1). We discuss below two papers that present related ideas.

[39] proposed the use of an approximate iterative solver for eqn. (3), followed by a correction step to "fix" the approximate solution (see our discussion in Section 1.1). We propose efficient, RLA-based approaches to precondition and solve eqn. (3), as well as a novel approach to correct for the approximation error in order to guarantee the convergence of the IPM algorithm. Specifically, [39] propose to solve eqn. (3) using the so-called *maximum weight basis* preconditioner [46]. However, computing such a preconditioner needs access to a maximal linearly independent set of columns of $\mathbf{AD}$ in each iteration, which is costly, taking $\mathcal{O}(m^2 n)$ time in the worst-case. More importantly, while [38] was able to provide a bound on the condition number of the preconditioned matrix that depends only on properties of $\mathbf{A}$, and is independent of $\mathbf{D}$, this bound might, in general, be very large. In contrast, our bound is a constant and it does not depend on properties of $\mathbf{A}$ or its dimensions. In addition, [39] assumed a bound on the two-norm of the residual of the preconditioned system, but it is unclear how their preconditioner guarantees such a bound. Similar concerns exist for the construction of the correction vector $\mathbf{v}$ proposed by [39], which our work alleviates.

The line of research in the Theoretical Computer Science literature that is closest to our work is [15], who presented an IPM that uses an approximate solver in each iteration. However, their accuracy guarantee is in terms of the final objective value which is different from ours. More importantly, [15] focuses on *short-step*, feasible IPMs, whereas ours is *long-step* and does not require a feasible starting point. Finally, the approximate solver proposed by [15] works only for the special case of input matrices that correspond to graph Laplacians, following the lines of [47, 48].

We also note that in the Theoretical Computer Science literature, [26, 27, 28, 29, 30, 7, 12] proposed and analyzed theoretically ground-breaking algorithms for LPs based on novel tools such as the so-called *inverse maintenance* for accelerating the linear system solvers in IPMs. However, all these endeavors are primarily focused on the theoretically fast but practically inefficient short-step feasible IPMs and, to the best of our knowledge, no implementations of these approaches are available for comparisons to standard long-step IPMs. We highlight that our work is focused on infeasible *long-step* IPMs, known to work efficiently in practice.

Another relevant line of research is the work of [14], which proposed solving eqn. (3) using preconditioned Krylov subspace methods, including variants of *generalized minimum residual* (GMRES) or CG methods. Indeed, [14] conducted extensive numerical experiments on LP problems taken from standard benchmark libraries, but did not provide any theoretical guarantees.

From a matrix-sketching perspective, our work was also partially motivated by [8], which presented an iterative, sketching-based algorithm to solve under-constrained ridge regression problems, but did not address how to make use of such approaches in an IPM-based framework, as we do here. In another work, [1] proposed a similar sketching-based preconditioning technique. However, their efforts broadly revolved around speeding up and scaling *kernel ridge regression*. [43, 53] proposed the so-called *Newton sketch* to construct an approximate Hessian matrix for more general convex objective functions of which LP is a special case. Nevertheless, these randomized second-order

methods are significantly faster than the conventional approach only when the data matrix is over-constrained, *i.e.* $m \gg n$. It is unclear whether the approach of [43, 53] is faster than IPMs when the optimization problem to be solved is linear. [49] proposed a probabilistic algorithm to solve LP approximately in a random projection-based reduced feature-space. A possible drawback of this paper is that the approximate solution is infeasible with respect to the original region. Finally, we refer the interested reader to the surveys [51, 19, 33, 18, 24, 34] for more background on Randomized Linear Algebra.

1.3 Notation and Background

$\mathbf{A}, \mathbf{B}, \ldots$ denote matrices and $\mathbf{a}, \mathbf{b}, \ldots$ denote vectors. For vector $\mathbf{a}$, $\|\mathbf{a}\|_2$ denotes its Euclidean norm; for a matrix $\mathbf{A}$, $\|\mathbf{A}\|_2$ denotes its spectral norm and $\|\mathbf{A}\|_F$ denotes its Frobenius norm. We use $\mathbf{0}$ to denote a null vector or null matrix, dependent upon context, and $\mathbf{1}$ to denote the all-ones vector. For any matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ with $m \leq n$ of rank m its thin Singular Value Decomposition (SVD) is the product $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$, with $\mathbf{U} \in \mathbb{R}^{m \times m}$ (the matrix of the left singular vectors), $\mathbf{V} \in \mathbb{R}^{n \times m}$(the matrix of the top-$m$ right singular vectors), and $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$ a diagonal matrix whose entries are equal to the singular values of $\mathbf{X}$. We use $\sigma_i(\cdot)$ to denote the i-th singular value of the matrix in parentheses.

We now briefly discuss a result on matrix sketching [13, 11] that is particularly useful in our theoretical analyses. In our parlance, [13] proved that, for any matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$, there exists a sketching matrix $\mathbf{W} \in \mathbb{R}^{n \times w}$ such that

$$\left\|\mathbf{Z}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{Z}^\mathsf{T} - \mathbf{Z}\mathbf{Z}^\mathsf{T}\right\|_2 \leq \frac{\zeta}{4}\left(\|\mathbf{Z}\|_2^2 + \frac{\|\mathbf{Z}\|_F^2}{r}\right) \tag{10}$$

holds with probability at least $1 - \delta$ for any $r \geq 1$. Here $\zeta \in [0, 1]$ is a (constant) accuracy parameter. Ignoring constant terms, $w = \mathcal{O}(r \log(r/\delta))$; $\mathbf{W}$ has $s = \mathcal{O}(\log(r/\delta))$ non-zero entries per row with s uniformly random entries are chosen without replacement and set to $\pm\frac{1}{s}$ independently; the product $\mathbf{Z}\mathbf{W}$ can be computed in time $\mathcal{O}(\log(r/\delta) \cdot \mathsf{nnz}(\mathbf{Z}))$.

2 Conjugate Gradient Solver

In this section, we discuss the computation of the preconditioner $\mathbf{Q}$ (and its inverse), followed by a discussion on how such a preconditioner can be used to satisfy eqns. (6) and (7).

Algorithm 1 Solving eqn. (5) via CG

Input: $\mathbf{A}\mathbf{D} \in \mathbb{R}^{m \times n}$, $\mathbf{p} \in \mathbb{R}^m$, sketching matrix $\mathbf{W} \in \mathbb{R}^{n \times w}$, iteration count t;

1: Compute $\mathbf{A}\mathbf{D}\mathbf{W}$ and its SVD: let $\mathbf{U}_\mathbf{Q}$ be the matrix of its left singular vectors and let $\mathbf{\Sigma}_\mathbf{Q}^{1/2}$ be the matrix of its singular values;

2: Compute $\mathbf{Q}^{-1/2} = \mathbf{U}_\mathbf{Q}\mathbf{\Sigma}_\mathbf{Q}^{-1/2}\mathbf{U}_\mathbf{Q}^\mathsf{T}$;

3: Initialize $\tilde{\mathbf{z}}^0 \leftarrow \mathbf{0}_m$ and run standard CG on the preconditioned system of eqn. (5) for t iterations;

Output: $\tilde{\mathbf{z}}^t$;

Algorithm 1 takes as input the sketching matrix $\mathbf{W} \in \mathbb{R}^{n \times w}$, which we construct as discussed in Section 1.3. Our preconditioner $\mathbf{Q}$ is equal to

$$\mathbf{Q} = \mathbf{A}\mathbf{D}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{D}\mathbf{A}^\mathsf{T}. \tag{11}$$

Notice that we only need to compute $\mathbf{Q}^{-1/2}$ in order to use it to solve eqn. (5). Towards that end, we first compute the sketched matrix $\mathbf{A}\mathbf{D}\mathbf{W} \in \mathbb{R}^{m \times w}$. Then, we compute the SVD of the matrix $\mathbf{A}\mathbf{D}\mathbf{W}$: let $\mathbf{U}_\mathbf{Q}$ be the matrix of its left singular vectors and let $\mathbf{\Sigma}_\mathbf{Q}^{1/2}$ be the matrix of its singular values. Notice that the left singular vectors of $\mathbf{Q}^{-1/2}$ are equal to $\mathbf{U}_\mathbf{Q}$ and its singular values are equal to $\mathbf{\Sigma}_\mathbf{Q}^{-1/2}$. Therefore, $\mathbf{Q}^{-1/2} = \mathbf{U}_\mathbf{Q}\mathbf{\Sigma}_\mathbf{Q}^{-1/2}\mathbf{U}_\mathbf{Q}^\mathsf{T}$.

Let $\mathbf{A}\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$ be the thin SVD representation of $\mathbf{A}\mathbf{D}$. We apply the results of [13] (see Section 1.3) to the matrix $\mathbf{Z} = \mathbf{V}^\mathsf{T} \in \mathbb{R}^{m \times n}$ with $r = m$ to get that, with probability at least $1 - \delta$,

$$\left\|\mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V} - \mathbf{I}_m\right\|_2 \leq \zeta/2 \tag{12}$$

The running time needed to compute the sketch $\mathbf{ADW}$ is equal to (ignoring constant factors) $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log(m/\delta))$. Note that $\mathsf{nnz}(\mathbf{AD}) = \mathsf{nnz}(\mathbf{A})$. The cost of computing the SVD of $\mathbf{ADW}$ (and therefore $\mathbf{Q}^{-1/2}$) is $\mathcal{O}(m^3 \log(m/\delta))$. Overall, computing $\mathbf{Q}^{-1/2}$ can be done in time

$$\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log(m/\delta) + m^3 \log(m/\delta)). \tag{13}$$

Given these results, we now discuss how to satisfy eqns. (6) and (7) using the sketching matrix $\mathbf{W}$. We start with the following bound, which is relatively straight-forward given prior RLA work (see Appendix C.1 for a proof).

Lemma 2 *If the sketching matrix* $\mathbf{W}$ *satisfies eqn.* (12)*, then, for all* $i = 1 \dots m$*,*

$$(1 + \zeta/2)^{-1} \leq \sigma_i^2(\mathbf{Q}^{-1/2}\mathbf{AD}) \leq (1 - \zeta/2)^{-1}.$$

This lemma directly implies eqn. (6). We now proceed to show that the above construction for $\mathbf{Q}^{-1/2}$, when combined with the conjugate gradient solver to solve eqn. (5), indeed satisfies eqn. (7)[3]. We do note that in prior work most of the convergence guarantees for CG focus on the error of the approximate solution. However, in our work, we are interested in the convergence of the *residuals* and it is known that even if the energy norm of the error of the approximate solution decreases monotonically, the norms of the CG residuals may oscillate. Interestingly, we can combine a result on the residuals of CG from [6] with Lemma 2 to prove that in our setting the norms of the CG residuals also decrease monotonically (see Appendix C.2 for details).

We remark that one can consider using MINRES [42] instead of CG. Our results hinges on bounding the two-norm of the residual. MINRES finds, at each iteration, the optimal vector with respect the two-norm of the residual inside the same Krylov subspace of CG for the corresponding iteration. Thus, the bound we prove for CG applies to MINRES as well.

3 The Infeasible IPM algorithm

In order to avoid spurious solutions, primal-dual path-following IPMs bias the search direction towards the *central path* and restrict the iterates to a neighborhood of the central path. This search is controlled by the *centering parameter* $\sigma \in [0, 1]$. At each iteration, given the current solution $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$, a standard infeasible IPM obtains the search direction $(\Delta\mathbf{x}^k, \Delta\mathbf{y}^k, \Delta\mathbf{s}^k)$ by solving the following system of linear equations:

$$\mathbf{AD}^2\mathbf{A}^\mathsf{T}\Delta\mathbf{y}^k = \mathbf{p}^k\,, \tag{14a}$$

$$\Delta\mathbf{s}^k = -\mathbf{r}_d^k - \mathbf{A}^\mathsf{T}\Delta\mathbf{y}^k\,, \tag{14b}$$

$$\Delta\mathbf{x}^k = -\mathbf{x}^k + \sigma\mu_k\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\Delta\mathbf{s}^k. \tag{14c}$$

Here $\mathbf{D}$ and $\mathbf{S}$ are computed given the current iterate ($\mathbf{x}^k$ and $\mathbf{s}^k$). After solving the above system, the infeasible IPM Algorithm 2 proceeds by computing a step-size $\bar{\alpha}$ to return:

$$(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) + \bar{\alpha}(\Delta\mathbf{x}^k, \Delta\mathbf{y}^k, \Delta\mathbf{s}^k). \tag{15}$$

Recall that $\mathbf{r}^k = (\mathbf{r}_p^k, \mathbf{r}_d^k)$ is a vector with $\mathbf{r}_p^k = \mathbf{Ax}^k - \mathbf{b}$ and $\mathbf{r}_d^k = \mathbf{A}^\mathsf{T}\mathbf{y}^k + \mathbf{s}^k - \mathbf{c}$ (the primal and dual residuals). We also use the *duality measure* $\mu_k = {\mathbf{x}^k}^\mathsf{T}\mathbf{s}^k/n$ and the vector

$$\mathbf{p}^k = -\mathbf{r}_p^k - \sigma\mu_k\mathbf{AS}^{-1}\mathbf{1}_n + \mathbf{Ax}^k - \mathbf{AD}^2\mathbf{r}_d^k. \tag{16}$$

Given $\Delta\mathbf{y}^k$ from eqn. (14a), $\Delta\mathbf{s}^k$ and $\Delta\mathbf{x}^k$ are easy to compute from eqns. (14b) and (14c), as they only involve matrix-vector products. However, since we will use Algorithm 1 to solve eqn. (14a) approximately using the sketching-based preconditioned CG solver, the primal and dual residuals *do not* lie on the line segment between $\mathbf{0}$ and $\mathbf{r}^0$. This invalidates known proofs of convergence for infeasible IPMs.

For notational simplicity, we now drop the dependency of vectors and scalars on the iteration counter k. Let $\hat{\Delta\mathbf{y}} = \mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t$ be the approximate solution to eqn. (14a). In order to account for the loss of accuracy due to the approximate solver, we compute $\hat{\Delta\mathbf{x}}$ as follows:

$$\hat{\Delta\mathbf{x}} = -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\hat{\Delta\mathbf{s}} - \mathbf{S}^{-1}\mathbf{v}. \tag{17}$$

[3] See Chapter 9 of [32] for a detailed overview of CG.

Here $\mathbf{v} \in \mathbb{R}^n$ is a perturbation vector that needs to exactly satisfy the following invariant at each iteration of the infeasible IPM:

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p}\,. \tag{18}$$

We note that the computation of $\hat{\Delta}\mathbf{s}$ is still done using eqn. (14b), which does not change. [39] argued that if $\mathbf{v}$ satisfies eqn. (18), the primal and dual residuals lie in the correct line segment.

Construction of $\mathbf{v}$. There are many choices for $\mathbf{v}$ satisfying eqn. (18). A general choice is $\mathbf{v} = (\mathbf{A}\mathbf{S}^{-1})^\dagger(\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p})$, which involves the computation of the pseudoinverse $(\mathbf{A}\mathbf{S}^{-1})^\dagger$, which is expensive, taking time $\mathcal{O}(m^2 n)$. Instead, we propose to construct $\mathbf{v}$ using the sketching matrix $\mathbf{W}$ of Section 1.3. More precisely, we construct the perturbation vector

$$\mathbf{v} = (\mathbf{X}\mathbf{S})^{1/2}\mathbf{W}(\mathbf{A}\mathbf{D}\mathbf{W})^\dagger(\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p}). \tag{19}$$

The following lemma proves that the proposed $\mathbf{v}$ satisfies eqn. (18); see Appendix C.3 for the proof.

Lemma 3 *Let $\mathbf{W} \in \mathbb{R}^{n\times w}$ be the sketching matrix of Section 1.3 and $\mathbf{v}$ be the perturbation vector of eqn. (19). Then, with probability at least $1-\delta$, $\operatorname{rank}(\mathbf{A}\mathbf{D}\mathbf{W}) = m$ and $\mathbf{v}$ satisfies eqn. (18).*

We emphasize here that we will use the same exact sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$ to form the preconditioner used in the CG algorithm of Section 2 *as well as* the vector $\mathbf{v}$ in eqn.(19). This allows us to form the sketching matrix only once, thus saving time in practice. Next, we present a bound for the two-norm of the perturbation vector $\mathbf{v}$ of eqn. (19); see Appendix C.4 for the proof.

Lemma 4 *With probability at least $1-\delta$, our perturbation vector $\mathbf{v}$ in Lemma 3 satisfies*

$$\|\mathbf{v}\|_2 \le \sqrt{3n\mu}\,\|\tilde{\mathbf{f}}^{(t)}\|_2, \tag{20}$$

with $\tilde{\mathbf{f}}^{(t)} = \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t - \mathbf{Q}^{-1/2}\mathbf{p}$.

Intuitively, the bound in eqn. (20) implies that $\|\mathbf{v}\|_2$ depends on how close the approximate solution $\hat{\Delta}\mathbf{y}$ is to the exact solution. Lemma 4 is particularly useful in proving the convergence of Algorithm 2, which needs $\|\mathbf{v}\|_2$ to be a small quantity. More precisely, combining a result from [39] with our preconditioner $\mathbf{Q}^{-1/2}$, we can prove that $\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2 \le \mathcal{O}(n)\sqrt{\mu}$. This bound allows us to prove that if we run Algorithm 1 for $\mathcal{O}(\log n)$ iterations, then $\|\tilde{\mathbf{f}}^{(t)}\|_2 \le \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu}$ and $\|\mathbf{v}\|_2 \le \frac{\gamma\sigma}{4}\mu$. The last two inequalities are critical in the convergence analysis of Algorithm 2; see Appendix F.1 and Appendix F.2 for details.

We are now ready to present the infeasible IPM algorithm. We will need the following definition for the neighborhood $\mathcal{N}(\gamma) = \{(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) : (\mathbf{x}^k, \mathbf{s}^k) > \mathbf{0}, x_i^k s_i^k \ge (1-\gamma)\mu \text{ and } \|\mathbf{r}^k\|_2/\|\mathbf{r}^0\|_2 \le \mu_k/\mu_0\}$. Here $\gamma \in (0,1)$ and we note that the duality measure μ_k steadily reduces at each iteration.

Algorithm 2 Infeasible IPM

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$, $\gamma \in (0,1)$, tolerance $\epsilon > 0$, $\sigma \in (0, 4/5)$;
Initialize: $k \leftarrow 0$; initial point $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$;

1: **while** $\mu_k > \epsilon$ **do**
2: Compute sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$ (Section 1.3) with $\zeta = 1/2$ and $\delta = O(n^{-2})$;
3: Compute $\mathbf{r}_p^k = \mathbf{A}\mathbf{x}^k - \mathbf{b}$; $\mathbf{r}_d^k = \mathbf{A}^\mathsf{T}\mathbf{y}^k + \mathbf{s}^k - \mathbf{c}$; and $\mathbf{p}^k$ from eqn. (16);
4: Solve the linear system of eqn. (5) for $\mathbf{z}$ using Algorithm 1 with $\mathbf{W}$ from step (2) and $t = \mathcal{O}(\log n)$. Compute $\hat{\Delta}\mathbf{y} = \mathbf{Q}^{-1/2}\mathbf{z}$;
5: Compute $\mathbf{v}$ using eqn. (19) with $\mathbf{W}$ from step (2); $\hat{\Delta}\mathbf{s}$ using eqn. (14b); $\hat{\Delta}\mathbf{x}$ using eqn. (17);
6: Compute $\tilde{\alpha} = \operatorname{argmax}\{\alpha \in [0,1] : (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) + \alpha(\hat{\Delta}\mathbf{x}^k, \hat{\Delta}\mathbf{y}^k, \hat{\Delta}\mathbf{s}^k) \in \mathcal{N}(\gamma)\}$.
7: Compute $\bar{\alpha} = \operatorname{argmin}\{\alpha \in [0,\tilde{\alpha}] : (\mathbf{x}^k + \alpha\hat{\Delta}\mathbf{x}^k)^\mathsf{T}(\mathbf{s}^k + \alpha\hat{\Delta}\mathbf{s}^k)\}$.
8: Compute $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) + \bar{\alpha}(\hat{\Delta}\mathbf{x}^k, \hat{\Delta}\mathbf{y}^k, \hat{\Delta}\mathbf{s}^k)$; set $k \leftarrow k+1$;
9: **end while**

Running time of Algorithm 2. We start by discussing the running time to compute $\mathbf{v}$. As discussed in Section 2, $(\mathbf{A}\mathbf{D}\mathbf{W})^\dagger$ can be computed in $\mathcal{O}(\mathsf{nnz}(\mathbf{A})\cdot\log(m/\delta) + m^3\log(m/\delta))$ time. Now, as

$\mathbf{W}$ has $\mathcal{O}(\log(m/\delta))$ non-zero entries per row, pre-multiplying by $\mathbf{W}$ takes $\mathcal{O}(\mathsf{nnz}(\mathbf{A})\log(m/\delta))$ time (assuming $\mathsf{nnz}(\mathbf{A}) \geq n$). Since $\mathbf{X}$ and $\mathbf{S}$ are diagonal matrices, computing $\mathbf{v}$ takes $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log(m/\delta) + m^3 \log(m/\delta))$ time, which is asymptotically the same as computing $\mathbf{Q}^{-1/2}$ (see eqn. (13)).

We now discuss the overall running time of Algorithm 2. At each iteration, with failure probability δ, the preconditioner $\mathbf{Q}^{-1/2}$ and the vector $\mathbf{v}$ can be computed in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log(m/\delta) + m^3 \log(m/\delta))$ time. In addition, for $t = \mathcal{O}(\log n)$ iterations of Algorithm 1, all the matrix-vector products in the CG solver can be computed in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log n)$ time. Therefore, the computational time for steps (2)-(5) is given by $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot (\log n + \log(m/\delta)) + m^3 \log(m/\delta))$. Finally, taking a union bound over all iterations with $\delta = \mathcal{O}(n^{-2})$ (ignoring constant factors), Algorithm 2 converges with probability at least 0.9. The running time at each iteration is given by $\mathcal{O}((\mathsf{nnz}(\mathbf{A}) + m^3) \log n)$.

4 Experiments

We demonstrate the empirical performance of our algorithm on a variety of synthetic and real-world datasets from the UCI ML Repository [20], such as ARCENE, DEXTER [23], DrivFace [16], and a gene expression cancer RNA-Sequencing dataset that is part of the PANCAN dataset [50]. See Appendix G, Table 1 for a description of the datasets. We observed that the results for both synthetic (Appendix G.2) and real-world data were qualitatively similar; we highlight results on representative real datasets. The experiments were implemented in Python and run on a server with Intel E5-2623V3@3.0GHz 8 cores and 64GB RAM. As an application, we consider ℓ_1-regularized SVMs: all of the datasets are concerned with binary classification with $m \ll n$, where n is the number of features. In Appendix G.1, we describe the ℓ_1-SVM problem and how it can be formulated as an LP. Here, m is the number of training points, n is the feature dimension, and the size of the constraint matrix in the LP becomes $m \times (2n + 1)$.

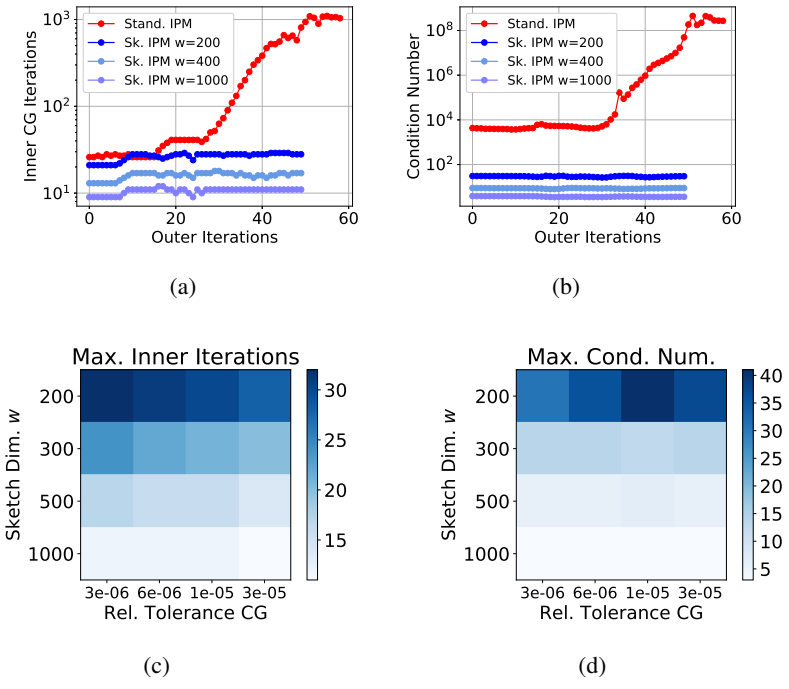


Figure 1: *ARCENE data set:* Our Algorithm 2 (Sk. IPM) requires an order of magnitude fewer (a) inner iterations than the Standard IPM with CG, at each outer iteration, due to the improved (b) conditioning of $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}$ compared to $\mathbf{A}\mathbf{D}^2\mathbf{A}^T$. For various ($w$, *tolCG*) settings, (c) the maximum number of inner iterations used by our algorithm and (d) the maximum condition number of $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}$, across outer iterations. The standard IPM, across all settings, needed on the order of 1,000 iterations and $\kappa(\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T})$ was on the order of 10^8.

Experimental Results. We compare our Algorithm 2 with a standard IPM (see Chapter 10, [44]) using CG and a standard IPM using a direct solver. We also use CVXPY as a benchmark to compare the accuracy of the solutions; we define the *relative error* $\|\hat{\mathbf{x}}-\mathbf{x}^\star\|_2/\|\mathbf{x}^\star\|_2$, where $\hat{\mathbf{x}}$ is our solution and $\mathbf{x}^\star$ is the solution generated by CVXPY. We also consider the number of *outer iterations*, namely the number of iterations of the IPM algorithm, as well as the number of *inner iterations*, namely the number of iterations of the CG solver. We denote the relative stopping tolerance for CG by *tolCG* and we denote the outer iteration residual by τ. If not specified: $\tau = 10^{-9}$, *tolCG* $= 10^{-5}$, and $\sigma = 0.5$. We evaluated a Gaussian sketching matrix and the initial triplet $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ for all IPM algorithms was set to be all ones.

Figure 1(a) shows that our Algorithm 2 uses an order of magnitude fewer *inner* iterations than the un-preconditioned standard solver. This is due to the improved conditioning of the respective matrices in the normal equations, as demonstrated in Figure 1(b). Across various real and synthetic data sets, the results were qualitatively similar to those shown in Figure 1. Results for several real data sets are summarized in Appendix G, Table 1. The number of *outer* iterations is unaffected by our internal approximation methods and is generally the same for our Algorithm 2, the standard IPM with CG, and the standard IPM with a direct linear solver (denoted IPM w/Dir), as seen in Appendix G, Table 1. Figure 1 also demonstrates the relative insensitivity to the choice of w (the sketching dimension, i.e., the number of columns of the sketching matrix $\mathbf{W}$ of Section 1.3). For smaller values of w, our algorithm requires more inner iterations. However, across various choices of w, the number of inner iterations is always an order of magnitude smaller than the number required by the standard solver.

Figures 1(c)-1(d) show the performance of our algorithm for a range of (w, *tolCG*) pairs. Figure 1(c) demonstrates that the number of the inner iterations is robust to the choice of *tolCG* and w. The number of inner iterations varies between 15 and 35 for the ARCENE data set, while the standard IPM took on the order of $1,000$ iterations across all parameter settings. Across all settings, the relative error was fixed at 0.04%. In general, our sketched IPM is able to produce an extremely high accuracy solution across parameter settings. Thus we do not report additional numerical results for the relative error, which was consistently 10^{-3} or less. Figure 1(d) demonstrates a tradeoff of our approach: as both *tolCG* and w are increased, the condition number $\kappa(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2})$ decreases, corresponding to better conditioned systems. As a result, fewer inner iterations are required. Additional experiments can be found in Appendix G.4.

5 Conclusions

We proposed and analyzed an infeasible IPM algorithm using a preconditioned conjugate gradient solver for the normal equations and a novel perturbation vector to correct for the error due to the approximate solver. Thus, we speed up each iteration of the IPM algorithm, without increasing the overall number of iterations. We demonstrate empirically that our IPM requires an order of magnitude fewer inner iterations within each linear solve than standard IPMs. Several extensions of our work are discussed in Appendix A.

Broader Impact

Our work is focused on speeding up algorithms for tall/wide LPs. As such, it could have significant broader impacts by allowing users to solve increasingly larger LPs in the numerous settings discussed in our introduction. While applications of our work to real data could result into ethical considerations, this is an indirect (and unpredictable) side-effect of our work. Our experimental work uses publicly available datasets to evaluate the performance of our algorithms; no ethical considerations are raised.

Acknowledgements, We thank the anonymous reviewers for their helpful comments. AC and PD were partially supported by NSF FRG 1760353 and NSF CCF-BSF 1814041. HA was partially supported by BSF grant 2017698. PL was supported by an Amazon Graduate Fellowship in Artificial Intelligence.

References

[1] Haim Avron, Kenneth L Clarkson, and David P Woodruff. Faster kernel ridge regression using sketching and preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1116–1138, 2017.

[2] Haim Avron, Petar Maymounkov, and Sivan Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010.

[3] Owe Axelsson and Vincent A. Barker. *Finite element solution of boundary value problems: Theory and computation*, volume 35. Society for Industrial and Applied Mathematics, 1984.

[4] Daniel Bienstock and Garud Iyengar. Approximating fractional packings and coverings in $\mathcal{O}(1/\epsilon)$ iterations. *SIAM Journal on Computing*, 35(4):825–854, 2006.

[5] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

[6] R. Bouyouli, Gérard Meurant, Laurent Smoch, and Hassane Sadok. New results on the convergence of the conjugate gradient method. *Numerical Linear Algebra with Applications*, 16(3):223–236, 2009.

[7] Jan van den Brand, Yin Tat Lee, Aaron Sidford, and Zhao Song. Solving tall dense linear programs in nearly linear time. *arXiv preprint arXiv:2002.02304*, 2020.

[8] Agniva Chowdhury, Jiasen Yang, and Petros Drineas. An iterative, sketching-based framework for ridge regression. In *Proceedings of the 35th International Conference on Machine Learning*, pages 988–997, 2018.

[9] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM symposium on Theory of Computing*, pages 81–90, 2013.

[10] Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):54, 2017.

[11] Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 278–287, 2016.

[12] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*, pages 938–942, 2019.

[13] Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal approximate matrix product in terms of stable rank. In *43rd International Colloquium on Automata, Languages, and Programming*, pages 11:1–11:14, 2016.

[14] Yiran Cui, Keiichi Morikuni, Takashi Tsuchiya, and Ken Hayami. Implementation of interior-point methods for LP based on Krylov subspace iterative solvers with inner-iteration preconditioning. *Computational Optimization and Applications*, 74(1):143–176, 2019.

[15] Samuel I. Daitch and Daniel A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 451–460, 2008.

[16] Katerine Diaz-Chito, Aura Hernández-Sabaté, and Antonio M. López. A reduced feature set for driver head pose estimation. *Applied Soft Computing*, 45:98–107, 2016.

[17] David L. Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 9446–9451, 2005.

[18] Petros Drineas and Michael W. Mahoney. RandNLA: Randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[19] Petros Drineas and Michael W. Mahoney. *Lectures on randomized numerical linear algebra*, volume 25 of *The Mathematics of Data, IAS/Park City Mathematics Series*. American Mathematical Society, 2018.

[20] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[21] David Chin-Lung Fong and Michael Saunders. CG vs. MINRES: An empirical comparison. *Sultan Qaboos University Journal for Science [SQUJS]*, 17(1):44–62, 2012.

[22] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.

[23] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in Neural Information Processing Systems*, pages 545–552, 2005.

[24] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[25] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.

[26] Yin Tat Lee and Aaron Sidford. Path finding I: Solving linear programs with $\tilde{\mathcal{O}}(\sqrt{rank})$ linear system solves. *arXiv preprint arXiv:1312.6677*, 2013.

[27] Yin Tat Lee and Aaron Sidford. Path finding II: An $\tilde{\mathcal{O}}(m\sqrt{n})$ algorithm for the minimum cost flow problem. *arXiv preprint arXiv:1312.6713*, 2013.

[28] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{\mathcal{O}}(\sqrt{rank})$ iterations and faster algorithms for maximum flow. In *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science*, pages 424–433, 2014.

[29] Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science*, pages 230–249, 2015.

[30] Yin Tat Lee and Aaron Sidford. Solving linear programs with $\tilde{\mathcal{O}}(\sqrt{rank})$ linear system solves. *arXiv preprint arXiv:1910.08033*, 2019.

[31] Palma London, Shai Vardi, Adam Wierman, and Hanling Yi. A parallelizable acceleration framework for packing linear programs. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 3706 – 3713, 2018.

[32] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer Publishing Company, Incorporated, 3rd edition, 2008.

[33] Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[34] Per-Gunnar Martinsson and Joel Tropp. Randomized numerical linear algebra: Foundations & algorithms. *arXiv preprint arXiv:2002.01387*, 2020.

[35] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 91–100, 2013.

[36] Xiangrui Meng, Michael A. Saunders, and Michael W. Mahoney. LSRN: A parallel iterative solver for strongly over- or underdetermined systems. *SIAM Journal on Scientific Computing*, 36(2):95–118, 2014.

[37] Ofer Meshi and Amir Globerson. An alternating direction method for dual MAP LP relaxation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 470–483. Springer, 2011.

[38] Renato D. C. Monteiro, Jerome W. O'Neal, and Takashi Tsuchiya. Uniform boundedness of a preconditioned normal matrix used in interior-point methods. *SIAM Journal on Optimization*, 15(1):96–100, 2004.

[39] Renato D. C. Monteiro and Jerome W. O'Neal. Convergence analysis of a long-step primal-dual infeasible interior-point LP algorithm based on iterative linear solvers. *Georgia Institute of Technology*, 2003.

[40] Jelani Nelson and Huy L. Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 54th IEEE Symposium on Foundations of Computer Science*, pages 117–126, 2013.

[41] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[42] Christopher C. Paige and Michael A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[43] Mert Pilanci and Martin J. Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

[44] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical recipes 3rd edition: The art of scientific computing. In *The Oxford Handbook of Innovation*, chapter 10. Cambridge University Press, 2007.

[45] Ben Recht, Christopher Re, Joel Tropp, and Victor Bittorf. Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222, 2012.

[46] Mauricio G. C. Resende and Geraldo Veiga. An implementation of the dual affine scaling algorithm for minimum-cost flow on bipartite uncapacitated networks. *SIAM Journal on Optimization*, 3(3):516–537, 1993.

[47] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, volume 4, pages 81–90, 2004.

[48] Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[49] Ky Vu, Pierre-Louis Poirion, and Leo Liberti. Random projections for linear programming. *Mathematics of Operations Research*, 43(4):1051–1071, 2018.

[50] John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna R. Mills Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M. Stuart, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013.

[51] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2), 2014.

[52] Stephen J. Wright. *Primal-dual interior-point methods*, volume 54. Society for Industrial and Applied Mathematics, 1997.

[53] Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

[54] Junfeng Yang and Yin Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011.

[55] Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11(Aug):2261–2286, 2010.

[56] Yin Zhang. On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM Journal on Optimization*, 4(1):208–227, 1994.

[57] Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J. Hastie. 1-norm support vector machines. In *Advances in Neural Information Processing Systems*, pages 49–56, 2004.

Appendix to Faster Randomized Infeasible Interior Point Methods for Tall/Wide Linear Programs

Appendix A Extensions

We briefly discuss extensions of our work. First, there is nothing special about using a CG solver for solving eqn. (5). We analyze two more solvers that could replace the proposed CG solver without any loss in accuracy or any increase in the number of iterations for the long-step infeasible IPM Algorithm 2 of Section 3. In Appendix D, we analyze the performance of the preconditioned Richardson Iteration and in Appendix E, we analyze the performance of the preconditioned Steepest Descent. In both cases, if the respective preconditioned solver (with the preconditioner of Section 2) runs for $t = \mathcal{O}(\log n)$ steps, Theorem 1 still holds, with small differences in the constant terms. While preconditioned Richardson iteration and preconditioned Steepest Descent are interesting from a theoretical perspective, they are not particularly practical.

Second, recall that our approach focused on full rank input matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \ll n$. Our overall approach still works if $\mathbf{A}$ in any $m \times n$ matrix that is low-rank, e.g., $\text{rank}(\mathbf{A}) = k \ll \min\{m, n\}$. In that case, using the thin SVD of $\mathbf{A}$, we can rewrite the linear constraints as follows $\mathbf{U_A \Sigma_A V_A^\mathsf{T} x} = \mathbf{b}$, where $\mathbf{U_A} \in \mathbb{R}^{m\times k}$ and $\mathbf{V_A} \in \mathbb{R}^{n\times k}$ are the matrices of left and right singular vecors of $\mathbf{A}$ respectively; $\mathbf{\Sigma_A} \in \mathbb{R}^{k\times k}$ is the diagonal matrix with the k non-zero singular values of $\mathbf{A}$ as its diagonal elements. The LP of eqn. (1) can be restated as

$$\min \mathbf{c}^\mathsf{T}\mathbf{x}\,, \text{ subject to } \mathbf{V_A^\mathsf{T} x} = \widetilde{\mathbf{b}}\,, \mathbf{x} \geq \mathbf{0}\,, \tag{21}$$

where $\widetilde{\mathbf{b}} = \mathbf{\Sigma_A^{-1} U_A^\mathsf{T} b}$. Note that, $\text{rank}(\mathbf{V_A}) = k \ll n$ and therefore eqn. (21) can be solved using our framework. The matrices $\mathbf{U_A}$, $\mathbf{V_A}$, and $\mathbf{\Sigma_A}$ can be approximately recovered using the fast SVD algorithms of [24, 5, 10]. However, the accuracy of the final solution will depend on the accuracy of the approximate SVD and we defer this analysis to future work.

Third, even though we chose to use the Count-Min sketch and its analysis from [13] (Section 1.3), there are many other alternative sketching matrix constructions that would lead to similar results. A particularly simple one is the Gaussian sketching matrix $\mathbf{W}_G \in \mathbb{R}^{n\times w}$, where every entry is a $\mathcal{N}(0, 1)$ random variable. Setting $w = \mathcal{O}\left({}^{m+\log(1/\delta)}/_{\zeta^2}\right)$ would result in the same accuracy guarantees as the sketching matrix of Section 1.3. However, the (theoretical) running time needed to compute $\mathbf{ADW}$ increases to $\mathcal{O}(m \cdot \mathsf{nnz}(\mathbf{A}))$. In practice, at least for relatively small matrices, using Gaussian sketching matrices is a reasonable alternative; see the discussion in [36] which argued that the Gaussian matrix sketching-based solvers are considerably better than direct solvers. We also opted to use Gaussian matrices in our empirical evaluation, since we primarily interested in measuring the accuracy of the final solution as a function of the number of iterations of the solver and the IPM algorithm. Other known constructions of sketching matrices that are also applicable in our setting include (any) sub-gaussian sketching matrix; the Subsampled Randomized Hadamard transform (SRHT); and any of the Sparse Subspace Embeddings of [9, 40, 35, 11].

We conclude by noting that our work can also be extended to analyze feasible IPMs, namely Algorithm 2 can start with a strictly feasible point. In this case, the analysis is somewhat simpler and the iteration complexity of the IPM algorithm reduces to $\mathcal{O}(n \log(1/\epsilon))$, which is the best known for feasible long-step path following IPM algorithms. We chose to present the more technically challenging infeasible IPM in this paper and delegate the feasible case to future work.

Appendix B Additional Notations

As before, we take $\mathbf{AD} = \mathbf{U\Sigma V}^\mathsf{T}$ to be the thin SVD representation of $\mathbf{AD}$. Additionally, for any two symmetric positive semidefinite (positive definite) matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ with same order, $\mathbf{A}_1 \preccurlyeq \mathbf{A}_2$ ($\mathbf{A}_1 \prec \mathbf{A}_2$) denotes that $\mathbf{A}_2 - \mathbf{A}_1$ is positive semidefinite (positive definite). For any two vectors $\mathbf{a} = (a_1, \ldots, a_\ell)^\mathsf{T}$ and $\mathbf{b} = (b_1, \ldots, b_\ell)^\mathsf{T}$ let $\mathbf{a} \circ \mathbf{b} = (a_1 b_1, \ldots, a_\ell b_\ell)^\mathsf{T}$. For any vector $\mathbf{a} \in \mathbb{R}^n$ its ℓ_∞ norm is defined as $\|\mathbf{a}\|_\infty = \max_i |a_i|$.

Appendix C Proofs

C.1 Proof of Lemma 2

Proof Consider the condition of eqn. (12):

$$\|\mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V} - \mathbf{I}_m\|_2 \leq \frac{\zeta}{2} \;\Leftrightarrow\; -\frac{\zeta}{2}\mathbf{I}_m \preccurlyeq \mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V} - \mathbf{I}_m \preccurlyeq \frac{\zeta}{2}\mathbf{I}_m \tag{22}$$

$$\Leftrightarrow\; -\frac{\zeta}{2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \preccurlyeq \mathbf{A}\mathbf{D}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{D}\mathbf{A}^\mathsf{T} - \mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \preccurlyeq \frac{\zeta}{2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \tag{23}$$

$$\Leftrightarrow\; \left(1-\frac{\zeta}{2}\right)\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \preccurlyeq \underbrace{\mathbf{A}\mathbf{D}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{D}\mathbf{A}^\mathsf{T}}_{\mathbf{Q}} \preccurlyeq \left(1+\frac{\zeta}{2}\right)\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}, \tag{24}$$

where we obtain eqn. (23) by pre- and post-multiplying the previous inequality by $\mathbf{U}\mathbf{\Sigma}$ and $\mathbf{\Sigma}\mathbf{U}^\mathsf{T}$ respectively and using the facts that $\mathbf{A}\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$ and $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\mathsf{T}$. Also, from eqn. (22), note that all the eigenvalues of $\mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V}$ lie between $(1-\frac{\zeta}{2})$ and $(1+\frac{\zeta}{2})$ *i.e.*, $\mathrm{rank}(\mathbf{V}^\mathsf{T}\mathbf{W}) = m$. Therefore, $\mathrm{rank}(\mathbf{A}\mathbf{D}\mathbf{W}) = \mathrm{rank}(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}\mathbf{W}) = m$, as $\mathbf{U}\mathbf{\Sigma}$ is non-singular and we know rank of a matrix remains unaltered by pre (or post)-multiplying by a non-singular matrix. So, we have $\mathrm{rank}(\mathbf{Q}) = m$; in words $\mathbf{Q}$ has full rank. Therefore, all the diagonal entries of $\mathbf{\Sigma}_\mathbf{Q}$ are positive and $\mathbf{Q}^{-1/2}\mathbf{Q}\mathbf{Q}^{-1/2} = (\mathbf{U}_\mathbf{Q}\mathbf{\Sigma}_\mathbf{Q}^{-1/2}\mathbf{U}_\mathbf{Q}^\mathsf{T})\,\mathbf{U}_\mathbf{Q}\mathbf{\Sigma}_\mathbf{Q}\mathbf{U}_\mathbf{Q}^\mathsf{T}\,(\mathbf{U}_\mathbf{Q}\mathbf{\Sigma}_\mathbf{Q}^{-1/2}\mathbf{U}_\mathbf{Q}^\mathsf{T}) = \mathbf{I}_m$.

Using above arguments, pre- and post- multiplying eqn. (24) by $\mathbf{Q}^{-1/2}$, we obtain

$$\left(1-\frac{\zeta}{2}\right)\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} \preccurlyeq \mathbf{I}_m \preccurlyeq \left(1+\frac{\zeta}{2}\right)\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}$$

$$\Leftrightarrow \left(1+\frac{\zeta}{2}\right)^{-1}\mathbf{I}_m \preccurlyeq \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} \preccurlyeq \left(1-\frac{\zeta}{2}\right)^{-1}\mathbf{I}_m. \tag{25}$$

Eqn. (25) implies and is implied by the fact that all the eigenvalues of $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}$ are bounded between $\left(1+\frac{\zeta}{2}\right)^{-1}$ and $\left(1-\frac{\zeta}{2}\right)^{-1}$. Therefore, we have

$$\left(1+\frac{\zeta}{2}\right)^{-1} \leq \sigma_i^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) \leq \left(1-\frac{\zeta}{2}\right)^{-1}, \text{ for } i = 1,\ldots,m.$$

■

C.2 Satisfying eqn. (7) using CG Solver

Let $\tilde{\mathbf{f}}^{(j)}$ be the residual at the j-th iteration of the CG algorithm, i.e., $\tilde{\mathbf{f}}^{(j)} = \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^j - \mathbf{Q}^{-1/2}\mathbf{p}$. Recall from Algorithm 1 that $\tilde{\mathbf{z}}^0 = \mathbf{0}$ and thus $\tilde{\mathbf{f}}^{(0)} = -\mathbf{Q}^{-1/2}\mathbf{p}$. In our parlance, Theorem 8 of [6] proved the following bound.

Lemma 5 (Theorem 8 of [6]) *Let $\tilde{\mathbf{f}}^{(j-1)}$ and $\tilde{\mathbf{f}}^{(j)}$ be the residuals obtained by the CG solver at steps $j-1$ and j. Then,*

$$\|\tilde{\mathbf{f}}^{(j)}\|_2 \leq \frac{\kappa^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) - 1}{2}\|\tilde{\mathbf{f}}^{(j-1)}\|_2,$$

where $\kappa(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D})$ is the condition number of $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}$.

From Lemma 2, we get

$$\kappa^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) = \frac{\sigma_{\max}^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D})}{\sigma_{\min}^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D})} \leq \frac{1+\zeta/2}{1-\zeta/2}. \tag{26}$$

Combining eqn. (26) with Lemma 5,

$$\|\tilde{\mathbf{f}}^{(j)}\|_2 \leq \frac{\frac{1+\zeta/2}{1-\zeta/2} - 1}{2}\|\tilde{\mathbf{f}}^{(j-1)}\|_2 = \frac{\zeta}{2-\zeta}\|\tilde{\mathbf{f}}^{(j-1)}\|_2 \leq \zeta\|\tilde{\mathbf{f}}^{(j-1)}\|_2, \tag{27}$$

where the last inequality follows from $\zeta \leq 1$. Applying eqn. (27) recursively, we get

$$\|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \zeta \|\tilde{\mathbf{f}}^{(t-1)}\|_2 \leq \cdots \leq \zeta^t \|\tilde{\mathbf{f}}^{(0)}\|_2 = \zeta^t \|\mathbf{Q}^{-1/2}\mathbf{p}\|_2 \,,$$

which proves the condition of eqn. (7).

C.3 Proof of Lemma 3

Proof Let $\mathbf{AD} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T}$ be the thin SVD representation of $\mathbf{AD}$. We use the exact same $\mathbf{W}$ as discussed in Section 2. Therefore, eqn. (12) holds with probability $1-\delta$ and it directly follows from the proof of Lemma 2 that $\mathrm{rank}(\mathbf{ADW}) = m$.

Now, as $\mathbf{ADW}$ has full *row-rank*, right-inverse exists and $\mathbf{ADW}\,(\mathbf{ADW})^\dagger = \mathbf{I}_m$. Therefore, taking $\mathbf{v} = (\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^\dagger(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p})$, we finally have

$$\begin{aligned}
\mathbf{AS}^{-1}\,\mathbf{v} &= \mathbf{AS}^{-1}(\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^\dagger(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p}) \\
&= \mathbf{ADW}(\mathbf{ADW})^\dagger(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p}) \\
&= \mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p}\,,
\end{aligned}$$

where the second equality follows from the fact that $\mathbf{D} = \mathbf{X}^{1/2}\mathbf{S}^{-1/2}$. This concludes the proof. ■

C.4 Proof of Lemma 4

Proof We already have, $\mathbf{Q} = \mathbf{ADW}(\mathbf{ADW})^\mathsf{T} = \mathbf{U}_\mathbf{Q}\boldsymbol{\Sigma}_\mathbf{Q}\mathbf{U}_\mathbf{Q}^\mathsf{T}$. From this, we know that $\mathbf{U}_\mathbf{Q}$ and $\boldsymbol{\Sigma}_\mathbf{Q}^{1/2}$ are respectively the matrices of left singular vectors and singular values of $\mathbf{ADW}$. Now, let $\widehat{\mathbf{V}}$ be the right singular vector of $\mathbf{ADW}$. Therefore, $\mathbf{ADW} = \mathbf{U}_\mathbf{Q}\boldsymbol{\Sigma}_\mathbf{Q}^{1/2}\widehat{\mathbf{V}}^\mathsf{T}$ is the thin SVD representation of $\mathbf{ADW}$. Also, from Lemma 2, we know $\mathbf{Q}$ has full rank. Therefore, $\mathbf{Q}^{1/2}\mathbf{Q}^{-1/2} = \mathbf{I}_m$.

Next, we bound $\|\mathbf{v}\|_2$ in the following way

$$\begin{aligned}
\|\mathbf{v}\|_2 &= \|(\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^\dagger(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p})\|_2 \\
&= \|(\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^\dagger\mathbf{Q}^{1/2}\mathbf{Q}^{-1/2}(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p})\|_2 \\
&\leq \|(\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^\dagger\mathbf{Q}^{1/2}\|_2\,\|\tilde{\mathbf{f}}^{(t)}\|_2 \,, \qquad (28)
\end{aligned}$$

where we have used the fact that $\mathbf{Q}^{-1/2}(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} - \mathbf{p}) = \tilde{\mathbf{f}}^{(t)}$ and the last inequality follows from the sub-multiplicativity property of spectral-norm.

Again, using SVD of $\mathbf{ADW}$ and $\mathbf{Q}$, we have $(\mathbf{ADW})^\dagger\mathbf{Q}^{1/2} = \widehat{\mathbf{V}}\boldsymbol{\Sigma}_\mathbf{Q}^{-1/2}\mathbf{U}_\mathbf{Q}^\mathsf{T}\,\mathbf{U}_\mathbf{Q}\boldsymbol{\Sigma}_\mathbf{Q}^{1/2}\mathbf{U}_\mathbf{Q}^\mathsf{T} = \widehat{\mathbf{V}}\mathbf{U}_\mathbf{Q}^\mathsf{T}$. Now, note that $\mathbf{U}_\mathbf{Q} \in \mathbb{R}^{m\times m}$ is an orthogonal matrix and $\widehat{\mathbf{V}} \in \mathbb{R}^{w\times m}$ has orthogonal columns *i.e.* $\|\widehat{\mathbf{V}}\|_2 = 1$. Therefore, combining these with eqn. (28) yields,

$$\begin{aligned}
\|\mathbf{v}\|_2 &\leq \|(\mathbf{XS})^{1/2}\mathbf{W}\widehat{\mathbf{V}}\mathbf{U}_\mathbf{Q}^\mathsf{T}\|_2\|\tilde{\mathbf{f}}^{(t)}\|_2 = \|(\mathbf{XS})^{1/2}\mathbf{W}\widehat{\mathbf{V}}\|_2\|\tilde{\mathbf{f}}^{(t)}\|_2 \\
&\leq \|(\mathbf{XS})^{1/2}\mathbf{W}\|_2\|\widehat{\mathbf{V}}\|_2\|\tilde{\mathbf{f}}^{(t)}\|_2 = \|(\mathbf{XS})^{1/2}\mathbf{W}\|_2\|\tilde{\mathbf{f}}^{(t)}\|_2 \,, \qquad (29)
\end{aligned}$$

where the first equality in eqn. (29) follows from the unitary invariance property of the spectral norm; the second inequality follows from the sub-multiplicativity of the spectral norm and the last equality is due to $\|\widehat{\mathbf{V}}\|_2 = 1$. Now, as we use the exact same $\mathbf{W}$ discussed in Section 2 to construct $\mathbf{v}$ and note that eqn. (10) holds for any matrix $\mathbf{Z}$ (irrespective of its dimensions). Therefore, taking $\mathbf{Z} = (\mathbf{XS})^{1/2}$ with that $\mathbf{W}$, eqn. (10) in Section 1.3 boils down to

$$\left\|(\mathbf{XS})^{1/2}\mathbf{W}\mathbf{W}^\mathsf{T}(\mathbf{XS})^{1/2} - (\mathbf{XS})\right\|_2 \leq \frac{\zeta}{4}\left(\|(\mathbf{XS})^{1/2}\|_2^2 + \frac{\|(\mathbf{XS})^{1/2}\|_F^2}{m}\right) \qquad (30)$$

holds with probability at least $1-\delta$.

Now, applying Weyl's inequality on the left hand side of the eqn. (30), we further have

$$\left|\left\|(\mathbf{XS})^{1/2}\mathbf{W}\right\|_2^2 - \left\|(\mathbf{XS})^{1/2}\right\|_2^2\right| \leq \frac{\zeta}{4}\left(\|(\mathbf{XS})^{1/2}\|_2^2 + \frac{\|(\mathbf{XS})^{1/2}\|_F^2}{m}\right) \qquad (31)$$

Now, using the facts that $\frac{\zeta}{4} \leq 1$, $\|(\mathbf{XS})^{1/2}\|_2 \leq \|(\mathbf{XS})^{1/2}\|_F$, and $\frac{\|(\mathbf{XS})^{1/2}\|_F^2}{m} \leq \|(\mathbf{XS})^{1/2}\|_F^2$, from eqn. (31),

$$\left\|(\mathbf{XS})^{1/2}\mathbf{W}\right\|_2^2 \leq 3\|(\mathbf{XS})^{1/2}\|_F^2 = 3n\mu\,, \tag{32}$$

where the last equality follows from $\|(\mathbf{XS})^{1/2}\|_F^2 = \mathbf{x}^\mathsf{T}\mathbf{s} = n\mu$.

Finally, combining eqns. (29) and (32), we conclude

$$\|\mathbf{v}\|_2 \leq \sqrt{3n\mu}\|\tilde{\mathbf{f}}^{(t)}\|_2.$$

■

Appendix D Richardson Iteration

Here, we show that all our analyses still hold, even if we replace Step 3 of Algorithm 1 (CG solver) with Richardson iteration. Basically, all we need to show is the condition in eqn. (7) holds. Note that the condition in eqn. (6) already holds from Lemma 2, as we use the exact same sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$ discussed in Section 2.

Algorithm 3 Richardson Iteration Solver

Input: $\mathbf{AD} \in \mathbb{R}^{m\times n}$, $\mathbf{p} \in \mathbb{R}^m$; number of iterations $t > 0$; sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$;
Initialize: $\tilde{\mathbf{z}}^0 \leftarrow \mathbf{0}_m$;
for $j = 1$ **to** t **do**
  $\tilde{\mathbf{z}}^j \leftarrow \tilde{\mathbf{z}}^{j-1} + \mathbf{Q}^{-1/2}(\mathbf{p} - \mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j-1})$;
end for
Output: return $\tilde{\mathbf{z}}^t$;

Our first result expresses the residual vector $\tilde{\mathbf{f}}^{(j)}$ in terms of $\tilde{\mathbf{f}}^{(j-1)}$ for $j = 1, 2, \ldots, t$.

Lemma 6 *Let $\tilde{\mathbf{f}}^{(j)}$, $j = 1, 2, \ldots, t$ be the residual vectors at each iteration.Then,*

$$\tilde{\mathbf{f}}^{(j)} = \left(\mathbf{I}_m - \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right)\tilde{\mathbf{f}}^{(j-1)}\,. \tag{33}$$

Recall that $\mathbf{Q} = \mathbf{ADWW}^\mathsf{T}\mathbf{DA}^\mathsf{T}$ and $\tilde{\mathbf{f}}^{(j)} = \mathbf{Q}^{-1/2}(\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^j - \mathbf{p})$.

Proof Using Algorithm 3, we express $\tilde{\mathbf{f}}^{(j)}$ as

$$\begin{aligned}
\tilde{\mathbf{f}}^{(j)} &= \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^j - \mathbf{Q}^{-1/2}\mathbf{p} \\
&= \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\left(\tilde{\mathbf{z}}^{j-1} + \mathbf{Q}^{-1/2}(\mathbf{p} - \mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j-1})\right) - \mathbf{Q}^{-1/2}\mathbf{p} \\
&= \left(\mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j-1} - \mathbf{Q}^{-1/2}\mathbf{p}\right) \\
&\qquad - \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\left(\mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j-1} - \mathbf{Q}^{-1/2}\mathbf{p}\right) \\
&= \left(\mathbf{I}_m - \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right)\left(\mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j-1} - \mathbf{Q}^{-1/2}\mathbf{p}\right) \\
&= \left(\mathbf{I}_m - \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right)\tilde{\mathbf{f}}^{(j-1)}\,,
\end{aligned}$$

which concludes the proof. ■

In the next result, we show that the spectral norm of $(\mathbf{I}_m - \mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2})$ is upper bounded by ζ.

Lemma 7 *Let the condition in eqn. (6) holds for the sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$, then*

$$\|\mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} - \mathbf{I}_m\|_2 \leq \zeta\,.$$

Proof As the condition in eqn. (6) holds, we can go backwards in the proof of Lemma 2 and see that eqn. (25) holds. So, we subtract $\mathbf{I}_m$ from each side of eqn. (25) to get

$$\left(\frac{2}{2+\zeta}-1\right)\mathbf{I}_m \preccurlyeq \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}-\mathbf{I}_m \preccurlyeq \left(\frac{2}{2-\zeta}-1\right)\mathbf{I}_m$$

$$\Leftrightarrow\ -\frac{\zeta}{2+\zeta}\mathbf{I}_m \preccurlyeq \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}-\mathbf{I}_m \preccurlyeq \frac{\zeta}{2-\zeta}\mathbf{I}_m$$

$$\Rightarrow\ -\frac{\zeta}{2-\zeta}\mathbf{I}_m \preccurlyeq \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}-\mathbf{I}_m \preccurlyeq \frac{\zeta}{2-\zeta}\mathbf{I}_m \tag{34}$$

$$\Leftrightarrow \|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}-\mathbf{I}_m\|_2 \leq \frac{\zeta}{2-\zeta} \leq \zeta\,, \tag{35}$$

where eqn. (34) holds as $\frac{\zeta}{2+\zeta} \leq \frac{\zeta}{2-\zeta}$ and the last inequality of eqn. (35) follows from $\zeta < 1$. ■

Satisfying eqn. (6). Note that the condition in eqn. (6) already holds from Lemma 2, as we use the exact same sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$ discussed in Section 2.

Satisfying eqn. (7). Using Lemma 7 and applying Lemma 6 recursively, we get

$$\|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \zeta\|\tilde{\mathbf{f}}^{(t-1)}\|_2 \leq \cdots \leq \zeta^t\|\tilde{\mathbf{f}}^{(0)}\|_2 = \zeta^t\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2\,.$$

Appendix E Steepest Descent

We will now replace Step 3 of Algorithm 1 (our proposed CG solver) by preconditioned steepest descent. We will again prove that our analysis of the proposed infeasible long-step IPM remains essentially the same.

First, we construct the sketching matrix $\mathbf{W}$ as discussed in Section 1.3, with a slightly more stringent accuracy guarantee. More specifically, we necessitate that

$$\left\|\mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V}-\mathbf{I}_m\right\|_2 \leq \frac{\zeta(1-\zeta)}{2} \tag{36}$$

holds with probability at least $1-\delta$ for a constant $\zeta \in [0,1]$. Notice that the sketching dimension $w = \mathcal{O}(m\log(m/\delta))$ and the running time needed to compute $\mathbf{Q}^{-1/2}$ (which is $\mathcal{O}(\mathsf{nnz}(\mathbf{A})\cdot\log(m/\delta)+m^3\log(m/\delta))$) remain, asymptotically, the same. In the case of steepest descent, it turns out that at each iteration the search direction is the negative of the gradient, which is equal to the residual $\tilde{\mathbf{f}}^{(j)}$. Moreover, the step size α_j is determined by an exact *line search* that minimizes the underlying quadratic function:

$$\alpha_j = \frac{\tilde{\mathbf{f}}^{(j)\mathsf{T}}\tilde{\mathbf{f}}^{(j)}}{\tilde{\mathbf{f}}^{(j)\mathsf{T}}\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)}}.$$

For this choice of α_j, it is easy to verify that the current gradient is orthogonal to the previous one.

Algorithm 4 Steepest Descent Solver

Input: $\mathbf{A}\mathbf{D} \in \mathbb{R}^{m\times n}$, $\mathbf{p} \in \mathbb{R}^m$; number of iterations $t > 0$; sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$;
Initialize: $\tilde{\mathbf{z}}^0 \leftarrow \mathbf{0}_m$;
for $j = 0$ **to** $t-1$ **do**

$\alpha_j = \frac{\tilde{\mathbf{f}}^{(j)\mathsf{T}}\tilde{\mathbf{f}}^{(j)}}{\tilde{\mathbf{f}}^{(j)\mathsf{T}}\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)}}$;

$\tilde{\mathbf{z}}^{j+1} \leftarrow \tilde{\mathbf{z}}^j - \alpha_j\tilde{\mathbf{f}}^{(j)}$;
end for
Output: return = $\tilde{\mathbf{z}}^t$;

Similar to Lemma 6, our next result reveals a recursive relation between the search directions which, later on, will be instrumental in bounding $\tilde{\mathbf{f}}^{(t)}$.

Lemma 8 *Let $\tilde{\mathbf{f}}^{(j)}$, $j = 1, 2, \ldots, t$ be the residual vectors at each iteration and α_j is given by Algorithm 4. Then,*

$$\tilde{\mathbf{f}}^{(j+1)} = \left(\mathbf{I}_m - \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right)\tilde{\mathbf{f}}^{(j)}\,, \tag{37}$$

Recall that $\mathbf{Q} = \mathbf{A}\mathbf{D}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{D}\mathbf{A}^\mathsf{T}$ and $\tilde{\mathbf{f}}^{(j)} = \mathbf{Q}^{-1/2}(\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^j - \mathbf{p})$.

Proof In Algorithm 4, we pre-multiply $\tilde{\mathbf{z}}^{j+1}$ with $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}$ and then subtract $\mathbf{Q}^{-1/2}\mathbf{p}$ to get

$$\begin{aligned}
\tilde{\mathbf{f}}^{(j+1)} &= \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j+1} - \mathbf{Q}^{-1/2}\mathbf{p} \\
&= \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^{j} - \mathbf{Q}^{-1/2}\mathbf{p} - \alpha_j\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)} \\
&= \tilde{\mathbf{f}}^{(j)} - \alpha_j\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)} = \left(\mathbf{I}_m - \alpha_j\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right)\tilde{\mathbf{f}}^{(j)}\,,
\end{aligned}$$

which concludes the proof. ■

Next, using this new condition in eqn. (36), we will bound $\left\|\mathbf{I}_m - \alpha_j\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right\|_2$ through a couple of results.

Lemma 9 *If eqn. (36) is satisfied, then $|\alpha_j - 1| \leq \frac{\zeta(1-\zeta)}{2}$.*

Proof First, we rewrite eqn. (36) as follows,

$$-\frac{\zeta(1-\zeta)}{2}\mathbf{I}_m \preccurlyeq \mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V} - \mathbf{I}_m \preccurlyeq \frac{\zeta(1-\zeta)}{2}\mathbf{I}_m$$

Next, we pre and post-multiply the the above expression by $\mathbf{U}\mathbf{\Sigma}$ and $\mathbf{\Sigma}\mathbf{U}^\mathsf{T}$ to get

$$-\frac{\zeta(1-\zeta)}{2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \preccurlyeq \underbrace{\mathbf{A}\mathbf{D}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{D}\mathbf{A}^\mathsf{T}}_{\mathbf{Q}} - \mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \preccurlyeq \frac{\zeta(1-\zeta)}{2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T} \tag{38}$$

Now, pre and post-multiplying eqn. (38) again by $\mathbf{Q}^{-1/2}$, we have

$$\begin{aligned}
&\left(1 - \frac{\zeta(1-\zeta)}{2}\right)\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} \preccurlyeq \mathbf{I}_m \preccurlyeq \left(1 + \frac{\zeta(1-\zeta)}{2}\right)\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} \\
\Rightarrow &\left(1 - \frac{\zeta(1-\zeta)}{2}\right)\tilde{\mathbf{f}}^{(j)^\mathsf{T}}\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)} \leq \tilde{\mathbf{f}}^{(j)^\mathsf{T}}\tilde{\mathbf{f}}^{(j)} \leq \left(1 + \frac{\zeta(1-\zeta)}{2}\right)\tilde{\mathbf{f}}^{(j)^\mathsf{T}}\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)} \\
\Rightarrow &\left(1 - \frac{\zeta(1-\zeta)}{2}\right) \leq \frac{\tilde{\mathbf{f}}^{(j)^\mathsf{T}}\tilde{\mathbf{f}}^{(j)}}{\tilde{\mathbf{f}}^{(j)^\mathsf{T}}\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{f}}^{(j)}} \leq \left(1 + \frac{\zeta(1-\zeta)}{2}\right) \\
\Leftrightarrow &\ |\alpha_j - 1| \leq \frac{\zeta(1-\zeta)}{2}\,,\ \text{for } j = 1, 2, \ldots, t\,.
\end{aligned} \tag{39}$$

■

Our next result shows that under eqn. (36), $\left\|\mathbf{I}_m - \alpha_j\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right\|_2$ is upper bounded by a small quantity for for $j = 1, 2, \ldots, t$.

Lemma 10 *If eqn. (36) is satisfied, then $\left\|\mathbf{I}_m - \alpha_j\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right\|_2 \leq \zeta$, for $j = 1, 2, \ldots, t$.*

Proof We note that eqn. (36) directly implies

$$\left\|\mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V} - \mathbf{I}_m\right\|_2 \leq \frac{\zeta}{2} \tag{40}$$

Now, as eqn. (40) holds, from eqn. (25) in the proof of Lemma 2, we have

$$\left(1 + \frac{\zeta}{2}\right)^{-1}\mathbf{I}_m \preccurlyeq \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} \preccurlyeq \left(1 - \frac{\zeta}{2}\right)^{-1}\mathbf{I}_m$$

$$\Leftrightarrow \left(\frac{2\alpha_j}{2+\zeta}-1\right)\mathbf{I}_m \preccurlyeq \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} - \mathbf{I}_m \preccurlyeq \left(\frac{2\alpha_j}{2-\zeta}-1\right)\mathbf{I}_m$$
$$\Leftrightarrow \frac{2(\alpha_j-1)-\zeta}{2+\zeta}\mathbf{I}_m \preccurlyeq \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} - \mathbf{I}_m \preccurlyeq \frac{2(\alpha_j-1)+\zeta}{2-\zeta}\mathbf{I}_m\,, \tag{41}$$

where the above expression follows from multiplying eqn. (25) by α_j and then subtracting $\mathbf{I}_m$.

Now, from Lemma 9, we have, $-\zeta(1-\zeta) \leq 2(\alpha_j - 1) \leq \zeta(1-\zeta)$ for $j = 1, 2, \ldots, t$. Using this in eqn. (41), we further have

$$-\frac{\zeta(1-\zeta)+\zeta}{2+\zeta}\mathbf{I}_m \preccurlyeq \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} - \mathbf{I}_m \preccurlyeq \frac{\zeta(1-\zeta)+\zeta}{2-\zeta}\mathbf{I}_m$$
$$\Leftrightarrow -\frac{\zeta(2-\zeta)}{2+\zeta}\mathbf{I}_m \preccurlyeq \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} - \mathbf{I}_m \preccurlyeq \zeta\,\mathbf{I}_m$$
$$\Rightarrow -\zeta\,\mathbf{I}_m \preccurlyeq \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2} - \mathbf{I}_m \preccurlyeq \zeta\,\mathbf{I}_m \tag{42}$$
$$\Rightarrow \left\|\mathbf{I}_m - \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\right\|_2 \leq \zeta\,,$$

where eqn. (42) is due to the fact that $\frac{2-\zeta}{2+\zeta} \leq 1$. ■

Satisfying eqn. (6). As eqn. (40) holds, eqn. (6) directly follows from Lemma 2.

Satisfying eqn. (7). Using Lemma 10 and applying Lemma 8 recursively, we get

$$\|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \zeta\|\tilde{\mathbf{f}}^{(t-1)}\|_2 \leq \cdots \leq \zeta^t\|\tilde{\mathbf{f}}^{(0)}\|_2 = \zeta^t\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2\,.$$

Appendix F Convergence Analysis of Algorithm 2

F.1 Number of Iterations for the CG Solver

In this section, most of the proofs follow [39] except for the fact that we used our sketching based preconditioner $\mathbf{Q}^{-1/2}$. Recall that $\mathcal{S}$ is the set of optimal and feasible solutions for the proposed LP.

Lemma 11 *Let $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ be the initial point with $(\mathbf{x}^0, \mathbf{s}^0) > \mathbf{0}$ and $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{s}^*) \in \mathcal{S}$ such that $(\mathbf{x}^*, \mathbf{s}^*) \leq (\mathbf{x}^0, \mathbf{s}^0)$ with $\mathbf{s}^0 \geq |\mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c}|$. Then, for any point $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$ such that $\mathbf{r} = \eta\,\mathbf{r}^0$ and $0 \leq \eta \leq \min\left\{1, \frac{\mathbf{s}^\mathsf{T}\mathbf{x}}{\mathbf{s}^{0\mathsf{T}}\mathbf{x}^0}\right\}$, then we have*

$$(i)\ \ \eta\,(\mathbf{x}^\mathsf{T}\mathbf{s}^0 + \mathbf{s}^\mathsf{T}\mathbf{x}^0) \leq 3n\mu\,, \tag{43a}$$
$$(ii)\ \ \eta\,\|\mathbf{S}(\mathbf{x}^* - \mathbf{x}^0)\|_2 \leq \eta\,\|\mathbf{S}\mathbf{x}^0\|_2 \leq \eta\mathbf{s}^\mathsf{T}\mathbf{x}^0 \leq 3n\mu\,, \tag{43b}$$
$$(iii)\ \ \eta\,\|\mathbf{X}(\mathbf{s}^0 + \mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c})\|_2 \leq 2\eta\,\|\mathbf{X}\mathbf{s}^0\|_2 \leq 2\eta\,\mathbf{x}^\mathsf{T}\mathbf{s}^0 \leq 6n\mu\,. \tag{43c}$$

Proof We prove eqns. (43a)–(43c) below.

Proof of eqn. (43a). For completeness, we provide a proof of eqn. (43a) which is already discussed in [39]. Since $(\mathbf{x}^*, \mathbf{s}^*, \mathbf{y}^*) \in \mathcal{S}$, the following equalities hold:

$$\mathbf{A}\mathbf{x}^* = \mathbf{b} \tag{44a}$$
$$\mathbf{A}^\mathsf{T}\mathbf{y}^* + \mathbf{s}^* = \mathbf{c} \tag{44b}$$

Furthermore, $\mathbf{r} = \eta\mathbf{r}^0$ implies

$$\mathbf{A}\mathbf{x} - \mathbf{b} = \eta(\mathbf{A}\mathbf{x}^0 - b) \tag{45a}$$
$$\mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} - \mathbf{c} = \eta(\mathbf{A}^\mathsf{T}\mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c}) \tag{45b}$$

Combining eqn. (44a) with eqn. (45a) and eqn. (44b) with eqn. (45b), we get

$$\mathbf{A}\left(\mathbf{x} - \eta\mathbf{x}^0 - (1-\eta)\mathbf{x}^*\right) = \mathbf{0} \tag{46a}$$

$$\mathbf{A}^\mathsf{T}(\mathbf{y} - \eta\mathbf{y}^0 - (1-\eta)\mathbf{y}^*) + (\mathbf{s} - \eta\mathbf{s}^0 - (1-\eta)\mathbf{s}^*) = \mathbf{0} \tag{46b}$$

Multiplying the eqn. (46b) by $\left(\mathbf{x} - \eta\mathbf{x}^0 - (1-\eta)\mathbf{x}^*\right)^\mathsf{T}$ on the left and using eqn. (46a), we get

$$\left(\mathbf{x} - \eta\mathbf{x}^0 - (1-\eta)\mathbf{x}^*\right)^\mathsf{T}\left(\mathbf{s} - \eta\mathbf{s}^0 - (1-\eta)\mathbf{s}^*\right) = 0\,,$$

expanding which we get

$$\begin{aligned}\eta\left(\mathbf{x}^{0^\mathsf{T}}\mathbf{s} + \mathbf{x}^\mathsf{T}\mathbf{s}^0\right) &= \eta^2\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 + (1-\eta)^2(\mathbf{x}^*)^\mathsf{T}\mathbf{s}^* + \mathbf{x}^\mathsf{T}\mathbf{s} \\ &\quad + \eta(1-\eta)\left(\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^* + (\mathbf{x}^*)^\mathsf{T}\mathbf{s}^0\right) - (1-\eta)\left((\mathbf{x}^*)^\mathsf{T}\mathbf{s} + \mathbf{x}^\mathsf{T}\mathbf{s}^*\right)\end{aligned} \tag{47}$$

Next, we use the given conditions and rewrite eqn. (47) as

$$\begin{aligned}\eta\left(\mathbf{x}^{0^\mathsf{T}}\mathbf{s} + \mathbf{s}^{0^\mathsf{T}}\mathbf{x}\right) &\le \eta^2\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 + \mathbf{x}^\mathsf{T}\mathbf{s} + \eta(1-\eta)\left(\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^* + \mathbf{s}^{0^\mathsf{T}}\mathbf{x}^*\right) \\ &\le \eta^2\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 + \mathbf{x}^\mathsf{T}\mathbf{s} + 2\eta(1-\eta)\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 \\ &\le 2\eta\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 + \mathbf{x}^\mathsf{T}\mathbf{s} \le 3\mathbf{x}^\mathsf{T}\mathbf{s} = 3n\mu\,,\end{aligned} \tag{48}$$

where the first inequality in eqn. (48) follows from from a couple of facts. First, $(1-\eta)((\mathbf{x}^*)^\mathsf{T}\mathbf{s} + \mathbf{x}^\mathsf{T}\mathbf{s}^*) \ge 0$ as $(\mathbf{x}^*, \mathbf{s}^*) \ge \mathbf{0}$ and $(\mathbf{x}^0, \mathbf{s}^0) \ge \mathbf{0}$; second, as $(\mathbf{x}^*, \mathbf{s}^*, \mathbf{y}^*) \in \mathcal{S}$ (which implies $\mathbf{x}^* \circ \mathbf{s}^* = \mathbf{0}$), we have $(\mathbf{x}^*)^\mathsf{T}\mathbf{s}^* = 0$. Second inequality in eqn. (48) holds as $\mathbf{x}^* \le \mathbf{x}^0$, $\mathbf{s}^* \le \mathbf{s}^0$, $(\mathbf{x}^*, \mathbf{s}^*) \ge \mathbf{0}$ and $(\mathbf{x}^0, \mathbf{s}^0) \ge \mathbf{0}$; combining which we have $(\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^* + \mathbf{s}^{0^\mathsf{T}}\mathbf{x}^*) \le 2\,\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0$. Third inequality in eqn. (48) is true as we have $\eta^2\mathbf{x}^{0^\mathsf{T}} + 2\eta(1-\eta)\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 = 2\eta\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 - \eta^2\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0 \le 2\eta\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0$. Final inequality holds as $\eta \le \frac{\mathbf{x}^\mathsf{T}\mathbf{s}}{\mathbf{x}^{0^\mathsf{T}}\mathbf{s}^0}$.

Proof of eqn. (43b). The last inequality directly follows from eqn. (43a); second last inequality is also easy to prove as

$$\|\mathbf{S}\mathbf{x}^0\|_2 = \sqrt{\sum_{i=1}^{s}(s_i x_i^0)^2} \le \sqrt{\left(\sum_{i=1}^{s} s_i x_i^0\right)^2} = \mathbf{s}^\mathsf{T}\mathbf{x}^0\,. \tag{49}$$

To prove the first inequality in eqn. (43b), we use the fact $\mathbf{x}^0 \ge \mathbf{x}^*$ as follows

$$\begin{aligned}\|\mathbf{S}\mathbf{x}^0\|_2^2 - \|\mathbf{S}(\mathbf{x}^* - \mathbf{x}^0)\|_2^2 &= \sum_{i=1}^{n}(s_i x_i^0)^2 - \sum_{i=1}^{n} s_i^2\left((x_i^*)^2 + (x_i^0)^2 - 2x_i^* x_i^0\right) \\ &= \sum_{i=1}^{n} s_i^2\left(2x_i^* x_i^0 - (x_i^*)^2\right) \ge 0\,.\end{aligned}$$

Proof of eqn. (43c). This can be proven using a similar approach as in eqn. (43b). Last inequality directly follows from eqn. (43a); second last inequality is also easy to prove as

$$\|\mathbf{X}\mathbf{s}^0\|_2 = \sqrt{\sum_{i=1}^{n}(x_i s_i^0)^2} \le \sqrt{\left(\sum_{i=1}^{n} x_i s_i^0\right)^2} = \mathbf{x}^\mathsf{T}\mathbf{s}^0\,. \tag{50}$$

For the first inequality, we proceed as follows

$$\begin{aligned}\|\mathbf{X}(\mathbf{s}^0 + \mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c})\|_2^2 &= \|\mathbf{X}\mathbf{s}^0\|_2^2 + \|\mathbf{X}(\mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c})\|_2^2 + 2\mathbf{s}^{0^\mathsf{T}}\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c}) \\ &= \|\mathbf{X}\mathbf{s}^0\|_2^2 + \sum_{i=1}^{n} x_i^2(\mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c})_i^2 + 2\sum_{i=1}^{n} x_i^2 s_i^0(\mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c})_i \\ &\le \|\mathbf{X}\mathbf{s}^0\|_2^2 + \sum_{i=1}^{n}(x_i s_i^0)^2 + 2\sum_{i=1}^{n}(x_i s_i^0)^2\end{aligned}$$

$$= \|\mathbf{X}\mathbf{s}^0\|_2^2 + \|\mathbf{X}\mathbf{s}^0\|_2^2 + 2\|\mathbf{X}\mathbf{s}^0\|_2^2 = 4\|\mathbf{X}\mathbf{s}^0\|_2^2\,, \tag{51}$$

where the inequality in eqn. (51) follows from $x_i \geq 0$, $s_i^0 \geq 0$ and $\left|(\mathbf{A}^\mathsf{T}\mathbf{y}^0 - \mathbf{c})_i\right| \leq s_i^0$ for all $i = 1, 2, \ldots n$. This concludes the proof of Lemma 11. ■

Our next result bounds $\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2$ which will be instrumental in proving the final convergence bound.

Lemma 12 *Let $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ be the initial point with $(\mathbf{x}^0, \mathbf{s}^0) > \mathbf{0}$ such that $\mathbf{x}^0 \geq \mathbf{x}^*$ and $\mathbf{s}^0 \geq \max\{\mathbf{s}^*, |\mathbf{c} - \mathbf{A}^\mathsf{T}\mathbf{y}^0|\}$ for some $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{s}^*) \in \mathcal{S}$. Furthermore, let $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$ with $\mathbf{r} = \eta\,\mathbf{r}^0$ for some $0 \leq \eta \leq 1$. If the sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$ satisfies the condition in eqn. (6), then*

$$\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2 \leq \sqrt{2}\left(\frac{9n}{\sqrt{1-\gamma}} + \sigma\sqrt{\frac{n}{1-\gamma}} + \sqrt{n}\right)\sqrt{\mu}\,.$$

Recall that, $\mathbf{r} = (\mathbf{r}_p, \mathbf{r}_d) = (\mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} - \mathbf{c})$ and $\mathbf{r}^0 = (\mathbf{r}_p^0, \mathbf{r}_d^0) = (\mathbf{A}\mathbf{x}^0 - \mathbf{b}, \mathbf{A}^\mathsf{T}\mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c})$.

Proof Note that after correcting the approximation error of the CG solver using $\mathbf{v}$, the primal and dual residuals $\mathbf{r} = (\mathbf{r}_p, \mathbf{r}_d)$ corresponding to an iterate $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$ always lie on the line segment between zero and $\mathbf{r}^{(0)}$. In other words, $\mathbf{r} = \eta\mathbf{r}^{(0)}$ always holds for some $\eta \in [0, 1]$. This was formally proven in Lemma 3.3 of [39]. To bound $\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2$, first we express $\mathbf{p}$ as in eqn. (3) and rewrite

$$\mathbf{Q}^{-1/2}\mathbf{p} = \mathbf{Q}^{-1/2}\left(-\mathbf{r}_p - \sigma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}_n + \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{D}^2\mathbf{r}_d\right) \tag{52}$$

Then, applying triangle inequality on $\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2$ in eqn. (52), we get

$$\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2 \leq \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4\,, \tag{53}$$

where

$$\begin{aligned}
\Delta_1 &= \|\mathbf{Q}^{-1/2}\mathbf{r}_p\|_2\,,\\
\Delta_2 &= \sigma\mu\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{1}_n\|_2\,,\\
\Delta_3 &= \|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}\mathbf{D}^{-1}\mathbf{x}\|_2\,,\\
\Delta_4 &= \|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{r}_d\|_2\,.
\end{aligned}$$

To bound Δ_1, Δ_2, Δ_3 and Δ_4 separately, we will heavily use the condition in eqn. (6). In particular, from eqn. (6), note that we have $\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}\|_2 \leq \sqrt{2}$ as $\zeta \leq 1$.

Bounding Δ_1. Putting $\mathbf{r}_p = \eta\,\mathbf{r}_p^0$, $\mathbf{r}_p^0 = \mathbf{A}\mathbf{x}^0 - \mathbf{b}$ and $\mathbf{b} = \mathbf{A}\mathbf{x}^*$, we rewrite Δ_1 as

$$\begin{aligned}
\Delta_1 &= \eta\,\|\mathbf{Q}^{-1/2}\mathbf{A}(\mathbf{x}^0 - \mathbf{x}^*)\|_2\\
&= \eta\,\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}\mathbf{D}^{-1}(\mathbf{x}^0 - \mathbf{x}^*)\|_2\\
&\leq \eta\,\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}\|_2\|\mathbf{D}^{-1}(\mathbf{x}^0 - \mathbf{x}^*)\|_2\\
&\leq \sqrt{2}\eta\,\|\mathbf{D}^{-1}(\mathbf{x}^0 - \mathbf{x}^*)\|_2\\
&= \sqrt{2}\eta\,\|(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{S}(\mathbf{x}^0 - \mathbf{x}^*)\|_2\\
&\leq \sqrt{2}\eta\,\|(\mathbf{X}\mathbf{S})^{-1/2}\|_2\,\|\mathbf{S}(\mathbf{x}^0 - \mathbf{x}^*)\|_2\,,
\end{aligned} \tag{54}$$

where the above steps follow from submultiplicativity and eqn. (6). From eqn. (6), note that we have $\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}\|_2 \leq \sqrt{2}$ as $\zeta \leq 1$. Now, applying eqn. (43b) and $\|(\mathbf{X}\mathbf{S})^{-1/2}\|_2 = \max_{1\leq i\leq n}\frac{1}{\sqrt{x_i s_i}}$, we further have

$$\begin{aligned}
\Delta_1 &\leq \sqrt{2}\max_{1\leq i\leq n}\frac{1}{\sqrt{x_i s_i}}\cdot 3n\mu\\
&\leq 3\sqrt{2}\,n\sqrt{\frac{\mu}{1-\gamma}}\,,
\end{aligned} \tag{55}$$

where the last inequality follows from $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$.

Bounding Δ_2. Applying submultiplicativity, we have

$$\begin{aligned}\Delta_2 &= \sigma\mu\,\|\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}\,(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{1}_n\|_2\\ &\le \sigma\mu\,\|\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}\|_2\|(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{1}_n\|_2\\ &\le \sqrt{2}\,\sigma\mu\,\|(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{1}_n\|_2\\ &= \sqrt{2}\,\sigma\mu\sqrt{\sum_{i=1}^{n}\frac{1}{x_i s_i}} \le \sqrt{2}\,\sigma\mu\sqrt{\sum_{i=1}^{n}\frac{1}{(1-\gamma)\mu}}\\ &= \sqrt{2}\,\sigma\sqrt{\frac{n\,\mu}{(1-\gamma)}}\,, \end{aligned} \tag{56}$$

where the second last inequality follows from eqn. (6) and last inequality holds as $(\mathbf{x},\mathbf{y},\mathbf{s}) \in \mathcal{N}(\gamma)$.

Bounding Δ_3. Putting $\mathbf{D} = \mathbf{S}^{-1/2}\mathbf{X}^{1/2}$; $\mathbf{x} = \mathbf{X}\,\mathbf{1}_n$ and

$$\begin{aligned}\Delta_3 &= \|\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}\,(\mathbf{S}^{1/2}\mathbf{X}^{-1/2})\,\mathbf{X}\,\mathbf{1}_n\|_2\\ &= \|\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}\,(\mathbf{S}\mathbf{X})^{1/2}\,\mathbf{1}_n\|_2\\ &\le \|\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}\|_2\|(\mathbf{S}\mathbf{X})^{1/2}\,\mathbf{1}_n\|_2\\ &\le \sqrt{2}\sqrt{\sum_{i=1}^{n} x_i s_i} = \sqrt{2n\,\mu}\,, \end{aligned} \tag{57}$$

where the inequalities follows respectively from submultiplicativity and eqn. (6).

Bounding Δ_4. Putting $\mathbf{r}_d = \eta\,\mathbf{r}_d^0$, we have

$$\begin{aligned}\Delta_4 &= \eta\|\mathbf{Q}^{-1/2}\,\mathbf{A}\,\mathbf{D}^2\mathbf{r}_d^0\|_2\\ &\le \eta\|\mathbf{Q}^{-1/2}\,\mathbf{A}\mathbf{D}\|_2\|(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{X}\mathbf{r}_d^0\|_2\\ &\le \sqrt{2}\eta\,\|(\mathbf{X}\mathbf{S})^{-1/2}\mathbf{X}(\mathbf{A}^\mathsf{T}\mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c})\|_2\\ &\le \sqrt{2}\eta\,\|(\mathbf{X}\mathbf{S})^{-1/2}\|_2\,\|\mathbf{X}(\mathbf{A}^\mathsf{T}\mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c})\|_2\,, \end{aligned}$$

where the above inequalities follow from submultiplicativity and eqn. (6). Now, applying eqn. (43c) and $\|(\mathbf{X}\mathbf{S})^{-1/2}\|_2 \le \frac{1}{\sqrt{(1-\gamma)\mu}}$, we further have

$$\Delta_4 \le\; 6\sqrt{2}n\sqrt{\frac{\mu}{1-\gamma}} \tag{58}$$

Final bound. Combining eqns. (53), (55) ,(56), (57) and (58)

$$\|\mathbf{Q}^{-1/2}\mathbf{p}\|_2 \le\; \sqrt{2}\left(\frac{9n}{\sqrt{1-\gamma}} + \sigma\sqrt{\frac{n}{1-\gamma}} + \sqrt{n}\right)\sqrt{\mu}\,. \tag{59}$$

This concludes the proof of Lemma 12. ■

Lemma 13 *Let the sketching matrix $\mathbf{W}$ satisfy the conditions in eqns. (6) and (7). Then, after $t \ge \frac{\log(4\sqrt{6n}\,\psi/\gamma\sigma)}{\log(1/\zeta)}$ iterations of the CG solver in Algorithm 1, we have the following:*

$$\|\tilde{\mathbf{f}}^{(t)}\|_2 \le \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu} \quad \textit{and} \quad \|\mathbf{v}\|_2 \le \frac{\gamma\sigma}{4}\mu\,,$$

where $\psi = \left(\frac{9n}{\sqrt{1-\gamma}} + \sigma\sqrt{\frac{n}{1-\gamma}} + \sqrt{n}\right)$ and $\tilde{\mathbf{f}}^{(t)} = \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t - \mathbf{Q}^{-1/2}\mathbf{p}$ is the residual of the solver.

Proof Combining Lemma 12 and the condition in eqn. (7), we have

$$\|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \zeta^t \psi \sqrt{2\mu}. \tag{60}$$

Now, $\|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu}$ holds if $\sqrt{2}\psi\,\zeta^t\sqrt{\mu} \leq \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu}$, which holds if $\left(\frac{1}{\zeta}\right)^t \geq \frac{4\sqrt{2\,n}\,\psi}{\gamma\sigma}$. The last inequality holds for our choice of t. Next, combining Lemma 4 and eqn. (60) we get

$$\|\mathbf{v}\|_2 \leq \sqrt{3n\mu}\,\|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \sqrt{6n}\,\zeta^t\psi\mu$$

Therefore, $\|\mathbf{v}\|_2 \leq \frac{\gamma\sigma\mu}{4}$ holds if $\sqrt{6n}\psi\,\zeta^t\psi\mu \leq \frac{\gamma\sigma\mu}{4}$, which holds for our choice of t. Now, fixing γ, σ, and ζ, after $t = \mathcal{O}(\log n)$ iterations of Algorithm 1 the conclusions of the lemma hold. ■

F.2 Determining Step-size, Bounding the Number of Iterations, and Proof of Theorem 1

Let $(\hat{\Delta\mathbf{x}}, \hat{\Delta\mathbf{y}}, \hat{\Delta\mathbf{s}})$ respectively satisfies eqns. (17), (18) and (14b). We rewrite the systen in the following alternative form

$$\mathbf{A}\hat{\Delta\mathbf{x}} = -\,\mathbf{r}_p\,, \tag{61a}$$

$$\mathbf{A}^{\mathsf{T}}\hat{\Delta\mathbf{y}} + \hat{\Delta\mathbf{s}} = -\,\mathbf{r}_d\,, \tag{61b}$$

$$\mathbf{X}\hat{\Delta\mathbf{s}} + \mathbf{S}\hat{\Delta\mathbf{x}} = -\,\mathbf{X}\mathbf{S}\,\mathbf{1}_n + \sigma\mu\,\mathbf{1}_n - \mathbf{v}\,. \tag{61c}$$

Indeed, first we now show how to satisfy eqns. (17), (18) and (14b) from eqn. (61). Pre-multiplying both sides of eqn. (61c) by $\mathbf{A}\mathbf{S}^{-1}$ and noting that $\mathbf{D}^2 = \mathbf{X}\mathbf{S}^{-1}$, we get

$$\begin{aligned} &\mathbf{A}\mathbf{D}^2\hat{\Delta\mathbf{s}} + \mathbf{A}\hat{\Delta\mathbf{x}} = -\mathbf{A}\mathbf{X}\mathbf{1}_n + \sigma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{A}\mathbf{S}^{-1}\mathbf{v} \\ \Rightarrow\; &\mathbf{A}\mathbf{D}^2\hat{\Delta\mathbf{s}} = \mathbf{r}_p - \mathbf{A}\mathbf{x} + \sigma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{A}\mathbf{S}^{-1}\mathbf{v}. \end{aligned} \tag{62}$$

Eqn. (62) holds as $\mathbf{A}\mathbf{X}\mathbf{1}_n = \mathbf{A}\mathbf{x}$ and, from eqn. (61a), $\mathbf{A}\hat{\Delta\mathbf{x}} = -\mathbf{r}_p$. Next, pre-multiplying eqn. (61b) by $\mathbf{A}\mathbf{D}^2$, we get

$$\begin{aligned} &\mathbf{A}\mathbf{D}^2\mathbf{A}^{\mathsf{T}}\hat{\Delta\mathbf{y}} + \mathbf{A}\mathbf{D}^2\hat{\Delta\mathbf{s}} = -\mathbf{A}\mathbf{D}^2\mathbf{r}_d \\ \Rightarrow\; &\mathbf{A}\mathbf{D}^2\mathbf{A}^{\mathsf{T}}\hat{\Delta\mathbf{y}} = -\mathbf{r}_p + \mathbf{A}\mathbf{x} - \sigma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{A}\mathbf{D}^2\mathbf{r}_d + \mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{p} + \mathbf{A}\mathbf{S}^{-1}\mathbf{v}. \end{aligned} \tag{63}$$

The first equality in eqn. (63) follows from eqn. (62) and the definition of $\mathbf{p}$ in eqn. (16). This establishes eqn. (18). Eqn. (14b) directly follows from eqn. (61b). Finally, we get eqn. (17) by pre-multiplying eqn. (61c) by $\mathbf{S}^{-1}$.

Next, we define each new point traversed by the algorithm as $(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha))$, where

$$(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) := (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\hat{\Delta\mathbf{x}}, \hat{\Delta\mathbf{y}}, \hat{\Delta\mathbf{s}}) \tag{64}$$

$$\mu(\alpha) := \mathbf{x}(\alpha)^T\mathbf{s}(\alpha)/n \tag{65}$$

$$\mathbf{r}(\alpha) := \mathbf{r}\left(\mathbf{x}(\alpha), \mathbf{s}(\alpha), \mathbf{y}(\alpha)\right). \tag{66}$$

The goal in this section is to bound the number of iterations required by Algorithm 2. Towards that end, we bound the magnitude of the step size α. First, we provide an upper bound on α, which allows us to show that each new point $(\mathbf{x}(\alpha), \mathbf{s}(\alpha), \mathbf{y}(\alpha))$ traversed by the algorithm stays within the neighborhood $\mathcal{N}(\gamma)$. Second, we provide a lower bound on α, which allows us to bound the number of iterations required. We use multiple lemmas from [39], which we reproduce here, without their proofs.

First, we provide an upper bound on α, ensuring that each new point $(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha))$ traversed by the algorithm stays within the neighborhood $\mathcal{N}(\gamma)$.

Lemma 14 (Lemma 3.5 of [39]) *Assume $(\hat{\Delta\mathbf{x}}, \hat{\Delta\mathbf{y}}, \hat{\Delta\mathbf{s}})$ satisfies eqns. (61) for some $\sigma > 0$, $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$ (for $\gamma \in (0, 1)$), and $\|\mathbf{v}\|_2 \leq \frac{\gamma\sigma\mu}{4}$. Then, $(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) \in \mathcal{N}(\gamma)$ for every scalar α such that*

$$0 \leq \alpha \leq \min\left\{1, \frac{\gamma\sigma\mu}{4\|\hat{\Delta\mathbf{x}} \circ \hat{\Delta\mathbf{s}}\|_\infty}\right\}. \tag{67}$$

We now provide a lower bound on the values of $\bar{\alpha}$ and the corresponding $\mu(\bar{\alpha})$; see Algorithm 2.

Lemma 15 (Lemma 3.6 of [39]) *In each iteration of Algorithm 2, if $\|\mathbf{v}\|_2 \leq \frac{\gamma\sigma\mu}{4}$, then the step size $\bar{\alpha}$ satisfies*

$$\bar{\alpha} \geq \min\left\{1, \frac{\min\{\gamma\sigma, (1-\frac{5}{4}\sigma)\}\mu}{4\|\hat{\Delta x} \circ \hat{\Delta s}\|_\infty}\right\} \tag{68}$$

and

$$\mu(\bar{\alpha}) = \left[1 - \frac{\bar{\alpha}}{2}(1-\frac{5}{4}\sigma)\right]\mu. \tag{69}$$

At this point, we have provided a lower bound (eqn. (68)) for the allowed values of the step size $\bar{\alpha}$. Next, we show that this lower bound is bounded away from zero. From eqn. (68) this is equivalent to showing that $\|\hat{\Delta\mathbf{x}} \circ \hat{\Delta\mathbf{s}}\|_\infty$ is bounded.

Lemma 16 (Lemma 3.7 of [39] (slightly modified)) *Let $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ be the initial point with $(\mathbf{x}^0, \mathbf{s}^0) > 0$ and $(\mathbf{x}^0, \mathbf{s}^0) \geq (\mathbf{x}^*, \mathbf{s}^*)$ for some $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{s}^*) \in \mathcal{S}$. Let $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$ be such that $\mathbf{r} = \eta\mathbf{r}^0$ for some $\eta \in [0,1]$ and $\|\mathbf{v}\|_2 \leq \frac{\gamma\sigma\mu}{4}$. Then, the search direction $(\hat{\Delta\mathbf{x}}, \hat{\Delta\mathbf{y}}, \hat{\Delta\mathbf{s}})$ produced by Algorithm 2 at each iteration satisfies*

$$\max\{\|\mathbf{D}^{-1}\hat{\Delta\mathbf{x}}\|_2, \|\mathbf{D}\hat{\Delta\mathbf{s}}\|_2\} \leq \left(1 + \frac{\sigma^2}{1-\gamma} - 2\sigma\right)^{1/2}\sqrt{n\mu} + \frac{6n}{\sqrt{(1-\gamma)}}\sqrt{\mu} + \frac{\gamma\sigma}{4\sqrt{1-\gamma}}\sqrt{\mu}. \tag{70}$$

We should note here that the above lemma is slightly different than Lemma 3.7 of [39]. Indeed, Lemma 3.7 of [39] actually proves the following bound:

$$\max\{\|\mathbf{D}^{-1}\hat{\Delta\mathbf{x}}\|_2, \|\mathbf{D}\hat{\Delta\mathbf{s}}\|_2\} \leq \left(1 + \frac{\sigma^2}{1-\gamma} - 2\sigma\right)^{1/2}\sqrt{n\mu} + \frac{6n}{\sqrt{(1-\gamma)}}\sqrt{\mu} + \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu}\,. \tag{71}$$

Notice that there is slight difference in the last term in the right-hand side, which does not asymptotically change the bound. The underlying reason for this difference is the fact that [39] constructed the vector $\mathbf{v}$ differently. In our case, we need to bound $\|(\mathbf{XS})^{-1/2}\mathbf{v}\|_2$, which we do as follows:

$$\|(\mathbf{XS})^{-1/2}\mathbf{v}\|_2 \leq \|(\mathbf{XS})^{-1/2}\|_2\,\|\mathbf{v}\|_2 \leq \frac{1}{\min_i\sqrt{x_i s_i}}\,\frac{\gamma\sigma\mu}{4}\,, \tag{72}$$

where in the above expression we use the fact that $\|(\mathbf{XS})^{-1/2}\|_2 = \frac{1}{\min_i\sqrt{x_i s_i}}$. Now as $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$, we further have $x_i s_i \geq (1-\gamma)\mu$ for all $i = 1 \ldots n$. Combining this with eqn. (72), we get

$$\|(\mathbf{XS})^{-1/2}\mathbf{v}\|_2 \leq \frac{\gamma\sigma\mu}{4\sqrt{(1-\gamma)\mu}} = \frac{\gamma\sigma}{4\sqrt{1-\gamma}}\sqrt{\mu}. \tag{73}$$

On the other hand, [39] had a different construction of $\mathbf{v}$ for which $\|(\mathbf{XS})^{-1/2}\mathbf{v}\|_2 = \|\tilde{\mathbf{f}}^{(t)}\|_2$ holds. Therefore they had the following bound:

$$\|(\mathbf{XS})^{-1/2}\mathbf{v}\|_2 = \|\tilde{\mathbf{f}}^{(t)}\|_2 \leq \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu}.$$

Also, note that after correcting the approximation error of the CG solver using $\mathbf{v}$, the primal and dual residuals $\mathbf{r} = (\mathbf{r}_p, \mathbf{r}_d)$ corresponding to an iterate $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$ always lie on the line segment between zero and $\mathbf{r}^{(0)}$. In other words, $\mathbf{r} = \eta\mathbf{r}^{(0)}$ always holds for some $\eta \in [0,1]$. This was formally proven in Lemma 3.3 of [39].

The next lemma bounds the number of iterations that Algorithm 2 needs when started with an infeasible point that is sufficiently positive.

Lemma 17 (Theorem 2.6 of [39]) *Assume that the constants γ and σ are such that $\max\{\gamma^{-1}, (1-\gamma)^{-1}, \sigma^{-1}, (1-\frac{5}{4}\sigma)^{-1}\} = \mathcal{O}(1)$. Let the initial point $(\mathbf{x}^0, \mathbf{s}^0, \mathbf{y}^0)$ satisfy $(\mathbf{x}^0, \mathbf{s}^0) \geq (\mathbf{x}^*, \mathbf{s}^*)$ for some $(\mathbf{x}^*, \mathbf{s}^*, \mathbf{y}^*) \in \mathcal{S}$ and $\|\mathbf{v}\|_2 \leq \frac{\gamma\sigma\mu}{4}$. Algorithm 2 generates an iterate $(\mathbf{x}^k, \mathbf{s}^k, \mathbf{y}^k)$ satisfying $\mu_k \leq \epsilon\mu_0$ and $\|\mathbf{r}^k\|_2 \leq \epsilon\|\mathbf{r}^0\|_2$ after $\mathcal{O}(n^2 \log 1/\epsilon)$ iterations.*

Finally, Theorem 1 follows from Lemmas 13 and 17.

Appendix G Additional Notes on Experiments

Problem	Size	Sketch IPM w/ Precond. CG				Stand. IPM w/ Unprec. CG			IPM w/ Dir.
	$(m \times N)$	w	In. It.	Out. It.	κ_{Sk}	In. It.	Out. It.	κ_{Stan}	Out. It.
ARCENE	$(100 \times 10K)$	200	**30**	50	38.09	**1.1K**	59	4.4×10^8	50
DEXTER	$(300 \times 20K)$	500	**39**	39	75.42	**4.6K**	39	7.6×10^9	39
DrivFace	$(606 \times 6.4K)$	$1K$	**50**	42	68.87	**139K**	43	17×10^{12}	42
Gene RNA	$(801 \times 20K)$	$2K$	**27**	44	20.03	**101K**	208	4.7×10^{12}	44

Table 1: Comparison of (our) sketched IPM with CG, standard IPM with CG, and Standard IPM with a direct solver, for the ℓ_1-SVM problem on UCI Machine Learning Repository [20] data sets. Across all, $\tau = 10^{-9}$ and a relative error of 10^{-3} or less was achieved. We define $\kappa_{\text{Sk}} = \kappa(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2})$ and $\kappa_{\text{Stan}} = \kappa(\mathbf{A}\mathbf{D}^2\mathbf{A}^T)$.

G.1 Support Vector Machines (SVMs)

The classical ℓ_1-SVM problem is as follows. We consider the task of fitting an SVM to data pairs $S = \{(x_i, y_i)\}_{i=1}^m$, where $x_i \in \mathbb{R}^N$ and $y_i \in \{+1, -1\}$ is a label for each data pair. Here, m is the number of training points, and N is the feature dimension. The SVM problem with an ℓ_1 regularizer has the following form.

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \|w\|_1 \\ \text{subject to} \quad & y_i(w^T x_i + b') \geq 1, \quad \forall i \in [m]. \end{aligned} \tag{74}$$

This problem can be written as an LP by introducing the variables w^+ and w^-, where $w = w^+ - w^-$. The objective becomes $\sum_j^n w_j^+ + w_j^-$, and we constrain $w_i^+ \geq 0$ and $w_i^- \geq 0$. Note that the size of the constraint matrix in the LP becomes $(m \times (2N + 1))$, where m is the number of training points, and N is the feature dimension.

G.2 Random Data

We generate random synthetic instances of linear programs as follows. To generate $A \in \mathbb{R}^{m \times n}$, we set $a_{ij} \sim_{i.i.d.} U(0, 1)$ with probability p and $a_{ij} = 0$ otherwise. We then add $\min\{m, n\}$ i.i.d. draws from $U(0, 1)$ to the main diagonal, to ensure each row of A has at least one nonzero entry. We set $b = Ax + 0.1z$, where x and z are random vectors drawn from $N(0, 1)$. Finally, we set $c \sim N(0, 1)$.

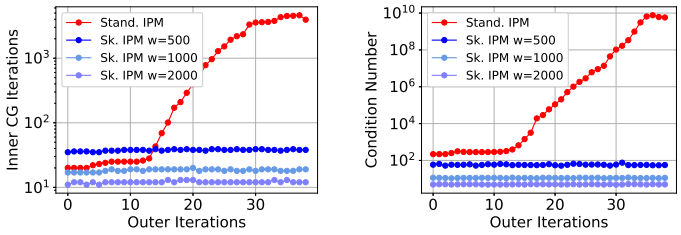
G.3 Real Data Descriptions

The following is how we made use a gene expression cancer RNA-Sequencing data set, taken from the UCI Machine Learning repository. It is part of the RNA-Seq (HiSeq) PANCAN data set [50], and is a random extraction of gene expressions from patients who have different types of tumors: BRCA, KIRC, COAD, LUAD and PRAD. We considered the binary classification task of identifying BRCA versus other types.

The following is how we made use of the DrivFace data set taken from the UCI Machine Learning repository. In the DrivFace data set, each sample corresponds to an image of a human subject, taken while driving in real scenarios. Each image is labeled as corresponding to one of 3 possible gaze directions (left, straight, or right). We considered the binary classification task of identifying two different gaze directions: (straight, or to either side left or right).

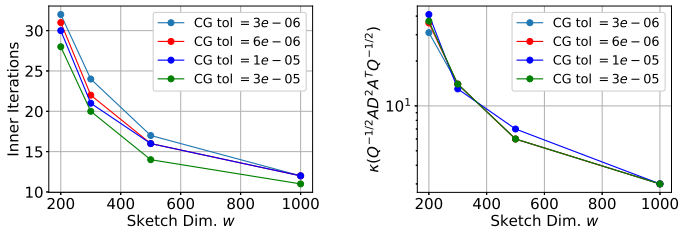
G.4 Additional Experiments

Here we include additional experiments. Figure 2 illustrates the convergence and conditioning behavior for the DEXTER data set. We see a similar behavior as found for the ARCENE data set in Figure 1. Figure 3 displays more results for the ARCENE data set.



(a) (b)

Figure 2: *DEXTER data set*: Our algorithm (Sk. IPM) requires an order of magnitude fewer inner iterations than the Standard IPM with CG, at each outer iteration, as demonstrated in (a). This is possible due to the improved conditioning of $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^{\mathsf{T}}\mathbf{Q}^{-1/2}$ compared to $\mathbf{A}\mathbf{D}^2\mathbf{A}^T$, demonstrated in (b). For all, *tolCG* $= 10^{-5}$, $\tau = 10^{-9}$.



(a) (b)

Figure 3: *ARCENE data set*: As w increases, (a) the number of inner iterations decreases, and is relatively robust to *tolCG*, and, (b) the condition number decreases as well.