A Fast Scale-Invariant Algorithm for Non-negative Least Squares with Non-negative Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Nonnegative (linear) least square problems are a fundamental class of problems that is well-studied in statistical learning and for which solvers have been implemented in many of the standard programming languages used within the machine learning community. The existing off-the-shelf solvers view the non-negativity constraint in these problems as an obstacle and, compared to unconstrained least squares, perform additional effort to address it. However, in many of the typical applications, the data itself is nonnegative as well, and we show that the nonnegativity in this case makes the problem easier. In particular, while the oracle complexity of unconstrained least squares problems necessarily scales with one of the data matrix constants (typically the spectral norm) and these problems are solved to additive error, we show that nonnegative least squares problems with nonnegative data are solvable to multiplicative error and with complexity that is independent of any matrix constants. The algorithm we introduce is accelerated and based on a primal-dual perspective. We further show how to provably obtain linear convergence using adaptive restart coupled with our method and demonstrate its effectiveness on large-scale data via numerical experiments.

1 Introduction

Nonnegative least squares (NNLS) problems, defined by $\min_{\mathbf{x}\geq\mathbf{0}} \ \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$, where $\mathbf{A}\in\mathbb{R}^{m\times n}$ and $\mathbf{b}\in\mathbb{R}^m$, are fundamental problems and have been studied for decades in optimization and statistical learning [25, 7, 22], with various off-the-shelf solvers available in standard packages of Python (as `optimize.nnls` in the SciPy package), Julia (as `nnls.jl`), and MATLAB (as `lsqnonneg`). Within machine learning, NNLS problems arise whenever having negative labels is not meaningful, for example, when representing prices, age, pixel intensities, chemical concentrations, or frequency counts. NNLS is also widely used as a subroutine in nonnegative matrix factorization [10, 17, 24] to extract sparse features in applications like clustering, collaborative filtering, and community detection.

From a statistical perspective, NNLS problems can be shown to possess a regularization property that enforces sparsity similar to LASSO [45], while being comparatively simpler, without the need to tune a regularization parameter or perform cross-validation [43, 8, 16, 23, 47, 42].

From an algorithmic standpoint, the nonnegativity constraint in NNLS problems is typically viewed as an obstacle: most NNLS algorithms perform additional work to handle it, and the problem is considered harder than unconstrained least squares. However, in many important applications of NNLS, such as text mining [6], functional MRI [3, 19], EEG data analysis [32], pulse oximetry [21, 48], statistical procedures in observational astronomy [18], and those traditionally addressed using nonnegative matrix factorization [10], *the data is also nonnegative*. We argue in this paper that when the data for NNLS is nonnegative, it is in fact possible to obtain *stronger* guarantees than for traditional least squares.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Contributions. We study NNLS problems with (element-wise) nonnegative data matrix $\mathbf{A}$, to which we refer as the NNLS+ problems, through the lens of the (equivalent) quadratic problems:

$$\min_{\mathbf{x}\geq\mathbf{0}} \left\{ \bar{f}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2}\|\mathbf{A}\mathbf{x}\|_2^2 - \mathbf{c}^\top \mathbf{x} \right\}, \tag{P}$$

where $\mathbf{c} = \mathbf{A}^\top \mathbf{b}$ may be assumed element-wise positive. This assumption is without loss of generality since if $c_j < 0$ for some j, then $\nabla_j \bar{f}(\mathbf{x}) \geq 0$, implying that the jth coordinate of the optimal solution is zero[1]. Hence, we could fix $x_j = 0$ and optimize over only the remaining coordinates.

We further assume that the matrix $\mathbf{A}$ is non-degenerate: none of its rows or columns has all elements equal to zero. This assumption is without loss of generality because (1) if such a row exists, we could remove it without affecting the objective, and (2) if the jth column had all elements equal to zero, the optimal value of (P) would be $-\infty$, obtained for $\mathbf{x}$ with $x_j \to \infty$. Having established our assumptions and setup, we now proceed to state our contributions, which are three-fold.

(1) A scale-invariant, ε-multiplicative algorithm. We design an algorithm based on coordinate descent that, in total cost $O(\frac{\text{nnz}(\mathbf{A})}{\sqrt{\varepsilon}})$, constructs an ε-*multiplicative* approximate solution to (P). Our algorithm capitalizes on structural properties of (P) that arise as a result of the nonnegativity of $\mathbf{A}$.

Theorem 1.1 (Informal; see Theorem 3.5)**.** *Given a matrix $\mathbf{A} \in \mathbb{R}_+^{m\times n}$ and $\varepsilon > 0$, define $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x}\|_2^2 - \mathbf{c}^\top\mathbf{x}$ and $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x}\geq\mathbf{0}} f(\mathbf{x})$. Then, there exists an algorithm that in $K = O(n\log n + \frac{n}{\sqrt{\varepsilon}})$ iterations and $O\big(nnz(\mathbf{A})\big(\log n + \frac{1}{\sqrt{\varepsilon}}\big)\big)$ arithmetic operations returns $\widetilde{\mathbf{x}}_K \in \mathbb{R}_+^n$ such that $\mathbb{E}\left[\langle \nabla f(\widetilde{\mathbf{x}}_K), \widetilde{\mathbf{x}}_K - \mathbf{x}^\star\rangle\right] \leq \varepsilon |f(\mathbf{x}^\star)|$.*

The application of our structural observations on (P) to Theorem 4.6 of [13] enables the recovery of our guarantee on the optimality gap; however, we provide a guarantee on the primal-dual gap, and this is *stronger* than the one on the optimality gap stated in Theorem 1.1. What is significant about Theorem 1.1 is the *invariance* of the computational complexity to the scale of $\mathbf{A}$—it does not depend on any matrix constants. This cost stands in stark contrast to that of traditional least squares, where the dependence of (oracle) complexity on matrix constants (specifically, the spectral norm of $\mathbf{A}$ in the Euclidean case) is *unavoidable* [34, 35], and multiplicative approximation is *not possible in general*.[2] Conceptually, our algorithm is a new acceleration technique inspired by VRPDA^2 [44].

(2) Linear convergence with restart. By incorporating adaptive restart in (P), we improve the guarantee of Theorem 1.1 to one with linear convergence (with $\log(1/\varepsilon)$ complexity). Thus, we establish the first theoretical guarantee for NNLS+ that simultaneously satisfies the properties of being *scale-invariant, accelerated, and linearly-convergent*.

Theorem 1.2 (Informal; see Theorem 4.1)**.** *Consider the setup of Theorem 1.1. Then, there is an algorithm that in expected $O(\text{nnz}(\mathbf{A})(\log n + \frac{\sqrt{n}}{\mu})\log(\frac{1}{\varepsilon}))$ arithmetic operations returns $\widetilde{\mathbf{x}}_K \in \mathbb{R}_+^n$ with $f(\widetilde{\mathbf{x}}_K) - f(\mathbf{x}^\star) \leq \varepsilon|f(\mathbf{x}^\star)|$, where μ is the constant in a local error bound for* (P).

Proving this bound requires bounding the expected number of iterations between restarts in conjunction with careful technical work in identifying an appropriate local error bound for NNLS+.

(3) Numerical experiments. We consolidate our theoretical contributions with a demonstration of the empirical advantage of our restarted method over state-of-the-art solvers via numerical experiments on datasets from LibSVM with sizes up to 19996×1355191. Figure 1 shows that, when combined with the restart strategy, our algorithm significantly outperforms the compared algorithms.

Related work. NNLS has seen a large body of work on the empirical front. The first method that was widely adopted in practice (including in the `lsqnonneg` implementation of MATLAB) is due to the seminal work of [25] (originally published in 1974). This method, based on active sets, solves NNLS via a sequence of (unconstrained) least squares problems and has been followed up by [7, 46, 33, 11] with improved empirical performance. While these variants are generally effective on small to mid-scale problem instances, they are not suitable for extreme-scale problems ubiquitous in machine learning. For example, in the experiments reported in [33], Fast NNLS [7] took

[1] To see this, note that by first-order optimality condition $\nabla \bar{f}(\mathbf{x}^\star)^\top(\mathbf{x} - \mathbf{x}^\star) \geq 0$ for all $\mathbf{x} \geq \mathbf{0}$. Choosing $\mathbf{x}$ with $\mathbf{x}_i = \mathbf{x}_i^\star$ for all $i \neq j$ and $\mathbf{x}_j = 0$ in the first-order optimality condition gives $\mathbf{x}_j^\star = 0$.

[2] To see why, consider a case in which the optimal objective value equals zero. Then any problem with a multiplicative guarantee of the form in Theorem 1.1 would necessarily return an optimal solution.

6.63 days to solve a problem of size 45000×45000, while the TNT-NN algorithm [33] took 2.45 hours. However, the latter requires computing the Cholesky decomposition of $\mathbf{A}^\top \mathbf{A}$ at initialization, which can be prohibitively expensive both in computation and in memory. Another prominent work on the empirical front is that of [22], which performs projected gradient descent with modified Barzilai-Borwein steps [4] and step sizes a carefully designed sequence of diminishing scalars.

Another, separate, line of work concerns optimization algorithms with multiplicative error guarantees. Of interest to us are first-order algorithms that run in time that is near-linear in the input size and are thus applicable to large-scale setting (for multiplicative-error algorithms applicable to broad classes of problems but that run in time that is superlinear in the input size, see [40, Chapter 7]). Most of the existing literature in this domain concerns positive (packing and covering) linear programs (e.g., [28, 49, 1, 29]). Results also exist for positive semidefinite programs [2], (nonlinear) fair packing and covering problems [31, 14], and fair packing problems under Schatten norms for matrices [20]. With the exception of [14] (discussed below), none of these results are directly applicable to (P).

To the best of our knowledge, theoretical guarantees explicitly for (P) have been scarce. For instance, [22] mentioned in the preceding paragraph provides only asymptotic guarantees. Orthogonally, the result on 1-fair covering by [14] solves the dual of NNLS+, which also gives a multiplicative guarantee for NNLS+, but with overall complexity $\tilde{O}(\frac{\mathrm{nnz}(\mathbf{A})}{\varepsilon})$, where $\tilde{O}(\cdot)$ hides poly-log factors.

Since our algorithm is based on the coordinate descent algorithm, we highlight some results of other coordinate descent algorithms when specialized to the closely related problem of *unconstrained linear regression*. The pioneering work of [36] proposed a coordinate descent method called `RCDM`, which in our setting has an iteration cost $O\big(\frac{\sum_{j=1}^n \|\mathbf{A}_{:j}\|_2^2 \|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\varepsilon}\big)$, where $\|\mathbf{A}_{:j}\|_2$ is the Euclidean norm of the jth column of $\mathbf{A}$. This was improved by [26], in an algorithm termed `ACDM`, by combining Nesterov's estimation technique [38] and coordinate sampling, giving an iteration complexity of $O\big(\frac{\sqrt{n \sum_{j=1}^n \|\mathbf{A}_{:j}\|_2^2} \|\mathbf{x}_0 - \mathbf{x}^\star\|}{\sqrt{\varepsilon}}\big)$ for solving (P). The latest results in this line of work by [2, 41, 39] perform non-uniform sampling atop a framework of [36] and achieve iteration complexity of $O\big(\frac{\sqrt{\sum_{j=1}^n \|\mathbf{A}_{:j}\|_2^2} \|\mathbf{x}_0 - \mathbf{x}^\star\|}{\sqrt{\varepsilon}}\big)$, with [12] dropping the dependence on $\max_{1 \le j \le n} \|\mathbf{A}_{:j}\|_2$. Additionally, the work of [27] develops an accelerated randomized proximal coordinate gradient (`APCG`) method to minimize composite convex functions.

As remarked earlier, [12], coupled with insights on NNLS+ problems provided in this work, can recover our guarantee for the optimality gap from Theorem 3.5. However, our work is the first to bring to the fore the properties of NNLS+ *required* to get such a guarantee, and our choice of primal-dual perspective allows for a stronger guarantee in terms of an upper bound on the primal-dual gap. Further, our algorithm is a novel type of acceleration, with our primal-dual perspective transparently illustrating our use of the aforementioned properties. We believe that these technical contributions, along with our techniques to obtain vastly improved theoretical guarantees with the restart strategy applied to this problem, are valuable to the broader optimization and machine learning communities.

2 Notation and preliminaries

Throughout the paper, we use bold lowercase letters to denote vectors and bold uppercase letters for matrices. For vectors and matrices, the operator $' \ge'$ is applied element-wise, and $\mathbb{R}_+$ is the non-negative part of the real line. We use $\langle \mathbf{a}, \mathbf{b} \rangle$ to denote the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ and ∇ for gradient. Given a matrix $\mathbf{A}$, we use $\mathbf{A}_{:j}$ for its jth column vector, and for a vector $\mathbf{x}$, x_j denotes its jth coordinate. We use $\mathrm{nnz}(\mathbf{A})$ for the number of non-zero entries of $\mathbf{A}$. We use $\mathbf{x}_k$ for the vector in the kth iteration and, to disambiguate indexing, use $[\mathbf{x}_k]_j$ to mean the jth coordinate of $\mathbf{x}_k$. The ith standard basis vector is denoted by $\mathbf{e}_i$. For an n-dimensional vector $\mathbf{x}$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$, we define $\mathbf{\Lambda} = \mathrm{diag}([\|\mathbf{A}_{:1}\|_2^2, \ldots, \|\mathbf{A}_{:n}\|_2^2])$ and $\|\mathbf{x}\|_{\mathbf{\Lambda}}^2 = \sum_{i=1}^n x_i^2 \|\mathbf{A}_{:i}\|_2^2$. Finally, $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$.

A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for any $\mathbf{x}, \widehat{\mathbf{x}} \in \mathbb{R}^n$, we have $f(\widehat{\mathbf{x}}) \ge f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \widehat{\mathbf{x}} - \mathbf{x} \rangle$. A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be μ-strongly convex w.r.t. the ℓ_2-norm if for any $\mathbf{x}, \widehat{\mathbf{x}} \in \mathbb{R}^n$, we have that $f(\widehat{\mathbf{x}}) \ge f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \widehat{\mathbf{x}} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2$. We have analogous definitions for concave and strongly concave functions, which flip the inequalities noted.

Given a convex program $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$, with $f:\mathbb{R}^n\to\mathbb{R}$ is differentiable and convex and $\mathcal{X}\subseteq\mathbb{R}^n$ closed and convex, the first-order optimality condition of a solution $\mathbf{x}^\star\in\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ is

$$(\forall \mathbf{x}\in\mathcal{X}): \quad \langle \nabla f(\mathbf{x}^\star), \mathbf{x}-\mathbf{x}^\star\rangle \geq 0. \tag{1}$$

Problem setup. As discussed in the introduction, our goal is to solve (P), with $\mathbf{A}\in\mathbb{R}_+^{m\times n}$. For notational convenience, we work with the problem in the following scaled form:

$$\min_{\mathbf{x}\in\mathbb{R}_+^n}\Big\{ f(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2}\|\mathbf{A}\mathbf{x}\|_2^2 - \mathbf{1}^\top\mathbf{x}\Big\}, \tag{2}$$

This assumption is w.l.o.g. since any (P) can be brought to this form by a simple change of variable $\hat{x}_j = c_j x_j$ (see also e.g., [1, 14] for similar scaling ideas). The scaling need not be explicit in the algorithm since the change of variable $\hat{x}_j = c_j x_j$ is easily reversible.

Properties of the objective. To kick off our analysis, we highlight some properties inherent to the objective defined in (2). These are central to obtain a scale-invariant algorithm for (P).

Proposition 2.1. *Given $f:\mathbb{R}_+^n\to\mathbb{R}$ as defined in (2) and $\mathbf{x}^\star\in\operatorname{argmin}_{\mathbf{x}\in\mathbb{R}_+^n} f(\mathbf{x})$, the following statements all hold.*

a) $\nabla f(\mathbf{x}^\star)\geq \mathbf{0}$.
b) $f(\mathbf{x}^\star) = -\frac{1}{2}\|\mathbf{A}\mathbf{x}^\star\|_2^2 = -\frac{1}{2}\mathbf{1}^\top\mathbf{x}^\star$.
c) for all $j\in[n]$, we have $x_j^\star \in \Big[0, \frac{1}{\|\mathbf{A}_{:j}\|_2^2}\Big]$.
d) $-\frac{1}{2}\sum_{j\in[n]}\frac{1}{\|\mathbf{A}_{:j}\|_2^2} \leq f(\mathbf{x}^\star) \leq -\frac{1}{2\min_{j\in[n]}\|\mathbf{A}_{:j}\|_2^2}$.

The validity of division by $\|\mathbf{A}_{:j}\|_2$ in the preceding proposition is by the non-degeneracy of $\mathbf{A}$ discussed in the introduction. We prove this proposition in Section 6.1.

An important consequence of Proposition 2.1 (c) is that (P) can be restricted to the hyperrectangle $\mathcal{X} = \{\mathbf{x}\in\mathbb{R}^n : 0\leq x_j \leq \frac{1}{\|\mathbf{A}_{:j}\|_2^2}\}$ without affecting its optimal solution, but effectively reducing the search space. Thus, going forward, we replace the constraint $\mathbf{x}\geq\mathbf{0}$ in (P) by $\mathbf{x}\in\mathcal{X}$.

Primal-dual gap perspective. As alluded earlier, our algorithm is analyzed through a primal-dual perspective. For this reason, it is useful to consider the Lagrangian

$$\mathcal{L}(\mathbf{x},\mathbf{y}) = \langle \mathbf{A}\mathbf{x}, \mathbf{y}\rangle - \frac{1}{2}\|\mathbf{y}\|_2^2 - \mathbf{1}^\top\mathbf{x} \tag{3}$$

from which we can derive our rescaled problem (2) as the primal problem $\min_{\mathbf{x}\in\mathcal{X}}\mathcal{P}(\mathbf{x})$, where

$$\mathcal{P}(\mathbf{x}) = \max_{\mathbf{y}\in\mathbb{R}^m}\mathcal{L}(\mathbf{x},\mathbf{y}) = -\mathbf{1}^\top\mathbf{x} + \max_{\mathbf{y}\geq\mathbf{0}}\Big[-\frac{1}{2}\|\mathbf{y}\|_2^2 + \langle\mathbf{A}\mathbf{x},\mathbf{y}\rangle\Big] = -\mathbf{1}^\top\mathbf{x} + \frac{1}{2}\|\mathbf{A}\mathbf{x}\|^2.$$

Similar to [44], we use Eq. (3) to define the following relaxation of the primal-dual gap, for arbitrary but fixed $\mathbf{u}\in\mathcal{X}$, and $\mathbf{v}\in\mathbb{R}^m$:

$$\text{Gap}_{\mathcal{L}}^{(\mathbf{u},\mathbf{v})}(\mathbf{x},\mathbf{y}) \stackrel{\text{def}}{=} \mathcal{L}(\mathbf{x},\mathbf{v}) - \mathcal{L}(\mathbf{u},\mathbf{y}). \tag{4}$$

The significance of this relaxed gap function is that for a candidate solution $\widetilde{\mathbf{x}}$ and an arbitrary $\widetilde{\mathbf{y}}\in\mathbb{R}^m$, a bound on $\text{Gap}_{\mathcal{L}}^{(\mathbf{u},\mathbf{v})}(\widetilde{\mathbf{x}},\widetilde{\mathbf{y}})$ translates to one on the primal error, as follows. First select $\mathbf{u}=\mathbf{x}^\star$, $\mathbf{v}=\mathbf{A}\widetilde{\mathbf{x}}$. Then, by observing that $\mathcal{L}(\widetilde{\mathbf{x}},\mathbf{A}\widetilde{\mathbf{x}}) = f(\widetilde{\mathbf{x}})$ and $\mathcal{L}(\mathbf{x}^\star,\mathbf{A}\mathbf{x}^\star) = f(\mathbf{x}^\star)$, we have

$$f(\widetilde{\mathbf{x}}) - f(\mathbf{x}^\star) = \mathcal{L}(\widetilde{\mathbf{x}},\mathbf{A}\widetilde{\mathbf{x}}) - \mathcal{L}(\mathbf{x}^\star,\mathbf{A}\mathbf{x}^\star).$$

For a fixed $\mathbf{x}$, $\mathcal{L}(\mathbf{x},\cdot)$ is 1-strongly concave and minimized at $\mathbf{A}\mathbf{x}$. Thus, $\mathcal{L}(\mathbf{x}^\star,\mathbf{A}\widetilde{\mathbf{x}}) \leq \mathcal{L}(\mathbf{x}^\star,\mathbf{A}\mathbf{x}^\star) - \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}-\mathbf{x}^\star)\|^2$. Hence, we have the following primal bound:

$$f(\widetilde{\mathbf{x}}) - f(\mathbf{x}^\star) + \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}-\mathbf{x}^\star)\|^2 \leq \mathcal{L}(\widetilde{\mathbf{x}},\mathbf{A}\widetilde{\mathbf{x}}) - \mathcal{L}(\mathbf{x}^\star,\widetilde{\mathbf{y}}) = \text{Gap}_{\mathcal{L}}^{(\mathbf{x}^\star,\mathbf{A}\widetilde{\mathbf{x}})}(\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}). \tag{5}$$

In light of this connection, our algorithm for bounding the primal error is one that generates iterates that can be used to construct bounds on $\text{Gap}_{\mathcal{L}}^{(\mathbf{u},\mathbf{v})}(\widetilde{\mathbf{x}},\widetilde{\mathbf{y}})$, as we detail next.

3 Our algorithm and convergence analysis

Our algorithm, Scale Invariant NNLS+ (SI-NNLS+), is an iterative algorithm using the estimate sequences $\phi_k(\mathbf{x})$ and $\psi_k(\mathbf{y})$ for $k \geq 1$ (see Section 3.1) giving the primal and dual updates

$$\mathbf{x}_k = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \phi_k(\mathbf{x}) \text{ and } \mathbf{y}_k = \operatorname*{argmax}_{\mathbf{y}\in\mathbb{R}^m} \psi_k(\mathbf{y}). \tag{6}$$

We use our algorithm's iterates from Eq. (6) to construct G_k, an upper estimate of $\mathrm{Gap}_{\mathcal{L}}^{(\mathbf{u},\mathbf{v})}(\widetilde{\mathbf{x}}_k, \widetilde{\mathbf{y}}_k)$, where $\widetilde{\mathbf{x}}_k, \widetilde{\mathbf{y}}_k$ are convex combinations of the iterates and $\mathbf{u} \in \mathcal{X}, \mathbf{v} \in \mathbb{R}_+^m$, where $\mathbf{u}$ is fixed and $\mathbf{v}$ is arbitrary. Our motivation for constructing $\mathrm{G}_k(\mathbf{u},\mathbf{v}) \geq \mathrm{Gap}_{\mathcal{L}}^{(\mathbf{u},\mathbf{v})}(\widetilde{\mathbf{x}}_k, \widetilde{\mathbf{y}}_k)$ is to obtain a bound on the primal error, using Inequality (5). The main goal in the analysis is to show that $\mathrm{G}_k \leq \frac{Q}{A_k}$, where A_k is a sequence of positive numbers and Q is bounded. To obtain the claimed multiplicative approximation, we use Inequality (5) and argue that for $\mathbf{u} = \mathbf{x}^\star, \mathbf{v} = \mathbf{A}\widetilde{\mathbf{x}}_k$, we have $Q \leq O\big(|f(\mathbf{x}^*)|\big)$.

Our algorithm may be interpreted as a variant of the VRPDA^2 algorithm [44], with our analysis technique inspired by the approximate duality gap technique [13]; however, unlike VRPDA^2, which uses estimate sequences, our analysis directly bounds the primal-dual gap. Another difference from VRPDA^2 is in our choice of primal regularizer (described shortly) and our lack of a dual regularizer. A variant of SI-NNLS+ suitable for the analysis is provided in Algorithm 1, with proofs in Section 6.2 and Section 6.3. We give an (equivalent) implementation version ("*Lazy* SI-NNLS+"[3]) in Algorithm 2 and its analysis in Section 7.

Algorithm 1 Scale Invariant Non-negative Least Squares with Non-negative Data (SI-NNLS+)

Input: Matrix $\mathbf{A} \in \mathbb{R}_+^{m\times n}$ with $n \geq 4$, accuracy ε, initial point $\mathbf{x}_0$
Initialize: $\widetilde{\mathbf{x}}_0 = \mathbf{x}_0$, $\overline{\mathbf{y}}_0 = \mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$, $K = \frac{5}{2}n\log n + \frac{6n}{\sqrt{\varepsilon}}$, $a_1 = \frac{1}{\sqrt{2}n^{1.5}}$, $a_2 = \frac{a_1}{n-1}$, $A_0 = 0$, $A_1 = a_1$, $\phi_0(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{\Lambda}}^2$.
for $k = 1, 2, \ldots, K$ **do**
  Sample j_k uniformly at random from $\{1, 2, \ldots, n\}$
  $\mathbf{x}_k \leftarrow \arg\min_{\mathbf{x}\in\mathcal{X}} \phi_k(\mathbf{x})$, for $\phi_k(\mathbf{x})$ defined by Eq. (12) and Eq. (15)
  $\mathbf{y}_k \leftarrow \arg\max_{\mathbf{y}\in\mathbb{R}^m} \psi_k(\mathbf{y})$, for $\psi_k(\mathbf{y})$ defined by Eq. (9)
  $\widetilde{\mathbf{x}}_k = \frac{1}{A_k}\left[A_{k-1}\widetilde{\mathbf{x}}_{k-1} + a_k\big(n\mathbf{x}_k - (n-1)\mathbf{x}_{k-1}\big)\right]$.
  $\overline{\mathbf{y}}_k \leftarrow \mathbf{y}_k + \frac{a_k}{a_{k+1}}(\mathbf{y}_k - \mathbf{y}_{k-1})$
  $A_{k+1} \leftarrow A_k + a_{k+1},\ a_{k+2} = \min\{\frac{na_{k+1}}{n-1}, \frac{\sqrt{A_{k+1}}}{2n}\}$
end for
Return $\widetilde{\mathbf{x}}_K$

3.1 Gap estimate construction

The gap estimate G_k is constructed as the difference $\mathrm{G}_k(\mathbf{u},\mathbf{v}) = \mathrm{U}_k(\mathbf{v}) - \mathrm{L}_k(\mathbf{u})$,where $\mathrm{U}_k(\mathbf{v}) \geq \mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{v})$ and $\mathrm{L}_k(\mathbf{u}) \leq \mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k)$ are, respectively, upper and lower bounds we construct on the Lagrangian. It then follows by Eq. (4) that $\mathrm{G}_k(\mathbf{u},\mathbf{v})$ is a valid upper estimate of $\mathrm{Gap}_{\mathcal{L}}^{(\mathbf{u},\mathbf{v})}(\widetilde{\mathbf{x}}_k, \widetilde{\mathbf{y}}_k)$.

We first introduce a technical component our constructions L_k and U_k crucially hinge on: we define two positive sequences of numbers $\{a_i\}_{i\geq 1}$ and $\{a_i^k\}_{1\leq i\leq k}$, with one of their properties being that both sum up to $A_k > 0$ for $k \geq 1$. Specifically, we define $A_0 = 0$ and $\{a_i\}_{i\geq 1}$ as $a_i = A_i - A_{i-1}$. The sequence $\{a_i^k\}$ changes with k and for $k = 1$ is defined by $a_1^1 = a_1$, while for $k \geq 2$:

$$a_i^k = \begin{cases} a_1 - (n-1)a_2, & \text{if } i = 1, \\ na_i - (n-1)a_{i+1}, & \text{if } 2 \leq i \leq k-1, \\ na_k, & \text{if } i = k. \end{cases} \tag{7}$$

Summing over $i \in [k]$ verifies that $A_k = \sum_{i=1}^k a_i^k$. For the sequence $\{a_i^k\}_{1\leq i\leq k}$ to be non-negative, we further require that $a_1 - (n-1)a_2 \geq 0$ and $\forall i \geq 2$, $na_i - (n-1)a_{i+1} \geq 0$.

[3] named so because, to improve cost per iteration, we update as few coordinates as is possible.

The significance of these two sequences lies in defining the algorithm's primal-dual output pair by

$$\widetilde{\mathbf{x}}_k = \frac{1}{A_k}\sum_{i\in[k]} a_i^k \mathbf{x}_i \quad \text{and} \quad \widetilde{\mathbf{y}}_k = \frac{1}{A_k}\sum_{i\in[k]} a_i \mathbf{y}_i. \tag{8}$$

The intricate interdependence of $\{a_i\}$ and $\{a_i^k\}$ enables expressing $\widetilde{\mathbf{x}}_k$ in terms of only $\{a_i\}$. This expression further simplifies to a cheaper recursive one, which is used in Algorithm 1.

With the sequences $\{a_i\}_{i\geq 1}$ and $\{a_i^k\}_{1\leq i\leq k}$ in tow, we are now ready to show the construction of an upper bound $\mathrm{U}_k(\mathbf{v})$ on $\mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{v})$ and a lower bound $\mathrm{L}_k(\mathbf{u})$ on $\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k)$.

Upper bound. To construct an upper bound, first observe that by Eq. (3) and Eq. (8),

$$\mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{v}) = \langle \mathbf{A}\widetilde{\mathbf{x}}_k, \mathbf{v}\rangle - \frac{1}{2}\|\mathbf{v}\|_2^2 - \mathbf{1}^\top \widetilde{\mathbf{x}}_k = \frac{1}{A_k}\sum_{i\in[k]} a_i^k \Big[\langle \mathbf{A}\mathbf{x}_i, \mathbf{v}\rangle - \frac{1}{2}\|\mathbf{v}\|_2^2 - \mathbf{1}^\top \mathbf{x}_i\Big].$$

Consider the primal estimate sequence defined for $k = 0$ as $\psi_0 = 0$ and for $k \geq 1$ by

$$\psi_k(\mathbf{v}) \stackrel{\text{def}}{=} \sum_{i\in[k]} a_i^k \Big[\langle \mathbf{A}\mathbf{x}_i, \mathbf{v}\rangle - \frac{1}{2}\|\mathbf{v}\|_2^2 - \mathbf{1}^\top \mathbf{x}_i\Big], \tag{9}$$

which ensures that $\mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{v}) = \frac{1}{A_k}\psi_k(\mathbf{v})$. A key upshot of constructing $\psi_k(\mathbf{v})$ as in Eq. (9) is that the quadratic term implies A_k-strong concavity of ψ_k for $k \geq 1$, which in turn ensures that the vector $\mathbf{y}_k = \arg\max_{\mathbf{y}\in\mathbb{R}^m} \psi_k(\mathbf{y})$ from Eq. (6) is unique. This property, coupled with the first-order optimality condition in Inequality (1), gives that for any $\mathbf{y} \in \mathbb{R}^m$, $\psi_k(\mathbf{y}) \leq \psi_k(\mathbf{y}_k) - \frac{A_k}{2}\|\mathbf{y} - \mathbf{y}_k\|_2^2$. We are now ready to define the following upper bound by:

$$\mathrm{U}_k(\mathbf{v}) \stackrel{\text{def}}{=} \frac{1}{A_k}\psi_k(\mathbf{y}_k) - \frac{1}{2}\|\mathbf{v} - \mathbf{y}_k\|_2^2. \tag{10}$$

The preceding discussion immediately implies that U_k is a valid upper bound for the Lagrangian.

Lemma 3.1. *For U_k as defined in Eq. (10), Lagrangian defined in Eq. (3) and $\widetilde{\mathbf{x}}_k \in \mathbb{R}_+^n$ in Eq. (8), we have, for all $\mathbf{y} \in \mathbb{R}^m$, the upper bound $\mathrm{U}_k(\mathbf{y}) \geq \mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{y})$.*

Lower bound. Analogous to the preceding section, we now obtain a *lower bound* on the Lagrangian, completing the bound on the gap estimate. However, the construction becomes more technical. We start with the same approach as for the upper bound. Since $\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k)$ is concave in $\widetilde{\mathbf{y}}_k$, by Jensen's inequality: $\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) \geq \frac{1}{A_k}\sum_{i\in[k]} a_i\Big(\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i\rangle - \mathbf{1}^\top\mathbf{u} - \frac{1}{2}\|\mathbf{y}_i\|_2^2\Big)$.Were we to define the dual estimate sequence ϕ_k in the same way as we did for the primal estimate sequence ψ_k, we would now simply define it as A_k times the right-hand side in the last inequality. However, doing so would make ϕ_k depend on $\mathbf{y}_k$, which is updated *after* $\mathbf{x}_k$, which in Eq. (6) is defined as the minimizer of the ϕ_k. To avoid such a circular dependency, we add and subtract a linear term $\sum_{i\in[k]}\langle \mathbf{A}^\top \overline{\mathbf{y}}_{i-1}, \mathbf{u}\rangle$, where $\overline{\mathbf{y}}_{i-1}$, defined later, are extrapolation points that depend only on $\mathbf{y}_1, \ldots \mathbf{y}_{i-1}$. We thus have

$$\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) \geq \frac{1}{A_k}\sum_{i\in[k]} a_i\Big[\langle \mathbf{A}\mathbf{u}, \overline{\mathbf{y}}_{i-1}\rangle - \mathbf{1}^\top\mathbf{u} - \frac{1}{2}\|\mathbf{y}_i\|_2^2\Big] + \frac{1}{A_k}\sum_{i\in[k]} a_i\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i - \overline{\mathbf{y}}_{i-1}\rangle.$$

If we now defined ϕ_k based on the first term in the above inequality, we run into another obstacle: the linearity of the resulting estimate sequence is insufficient for cancelling all the error terms in the analysis. Hence, as is common, we introduce strong convexity by adding and subtracting an appropriate strongly convex function. Our chosen strongly convex function is motivated by the box-constrained property of the optimum from Proposition 2.1 (c) and crucial in bounding the initial gap estimate. It coincides with ϕ_0: for any $\mathbf{x} \in \mathbb{R}_+^n$, define the function

$$\phi_0(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{\Lambda}}^2. \tag{11}$$

This function is 1-strongly convex with respect to $\|\cdot\|_{\mathbf{\Lambda}}$ and used in defining ϕ_1 as:

$$\phi_1(\mathbf{u}) = a_1\langle \mathbf{A}^\top\bar{\mathbf{y}}_0 - \mathbf{1}, \mathbf{u}\rangle + \phi_0(\mathbf{u}). \tag{12}$$

The definition of ϕ_1 is driven by the purpose of cancelling initial error terms. Next, we choose ϕ_k so that for any fixed $\mathbf{u} \in \mathcal{X}$, we have

$$\mathbb{E}[\phi_k(\mathbf{u})] = \mathbb{E}\Big[\sum_{i\in[k]} a_i\langle \mathbf{A}^\top \overline{\mathbf{y}}_{i-1} - \mathbf{1}, \mathbf{u}\rangle + \phi_0(\mathbf{u})\Big], \tag{13}$$

where the expectation is with respect to all the randomness in the algorithm. This construction is used to reduce the per-iteration complexity, for which we employ a *randomized* coordinate update on $\mathbf{x}_k$ for $k \geq 2$. To support such updates, we relax the lower bound to hold only *in expectation*.

Concretely, let j_i be the coordinate sampled uniformly at random from $[n]$ in the ith iteration of SI-NNLS+, independent of history. Fix $\mathbf{y}_i$ for $i = 1, \ldots, k-1$ and for $k \geq 2$ and $\mathbf{x} \in \mathcal{X}$, define

$$\phi_k(\mathbf{x}) = \phi_1(\mathbf{x}) + \sum_{i=2}^{k} n a_i \langle \mathbf{A}^\top \overline{\mathbf{y}}_{i-1} - \mathbf{1}, x_{j_i}\mathbf{e}_{j_i}\rangle. \tag{14}$$

For $k \geq 2$, $\phi_k(\mathbf{u})$ can also be defined recursively via

$$\phi_k(\mathbf{x}) = \phi_{k-1}(\mathbf{x}) + n a_k \langle \mathbf{A}^\top \overline{\mathbf{y}}_{k-1} - \mathbf{1}, x_{j_k}\mathbf{e}_{j_k}\rangle. \tag{15}$$

The function ϕ_k inherits the strong convexity of ϕ_0. This property, together with Eq. (6) and first-order optimality from Inequality (1), give

$$\phi_k(\mathbf{x}) \geq \phi_k(\mathbf{x}_k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{\Lambda}}^2. \tag{16}$$

Along with strong convexity, our choice of ϕ_k in Eq. (15) leads to the following properties essential to our analysis: (1) ϕ_k is separable in its coordinates; (2) the primal variable $\mathbf{x}_k$ is updated only at its j_k^{th} coordinate; (3) Eq. (13) is true. These are formally stated in Proposition 6.2.

With the dual estimate sequence ϕ_k defined in Eq. (15), we now define the sequence L_k by

$$\mathrm{L}_k(\mathbf{x}) \stackrel{\text{def}}{=} \frac{\phi_k(\mathbf{x}_k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{\Lambda}}^2 - \phi_0(\mathbf{x}) + \sum_{i\in[k]} a_i(\langle \mathbf{A}\mathbf{x}, \mathbf{y}_i - \overline{\mathbf{y}}_{i-1}\rangle - \frac{1}{2}\|\mathbf{y}_i\|_2^2)}{A_k}. \tag{17}$$

We conclude this section by justifying our choice of L_k as a valid expected lower bound on $\mathbb{E}\mathcal{L}(\mathbf{x}, \widetilde{\mathbf{y}}_k)$.

Lemma 3.2. *For L_k defined in Eq. (17), for the Lagrangian in Eq. (3) and $\widetilde{\mathbf{y}}_k$ in Eq. (8), we have, for a fixed $\mathbf{u} \in \mathcal{X}$, the lower bound $\mathbb{E}\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) \geq \mathbb{E}\mathrm{L}_k(\mathbf{u})$, where the expectation is with respect to all the random choices of coordinates in Algorithm 1.*

3.2 Bounding the gap estimate

With the gap estimate G_k constructed as in the preceding section and combining Eq. (10) and Eq. (17), we now achieve our goal of bounding $A_k\mathrm{G}_k$ (to obtain a convergence rate of the order $1/A_k$) by bounding the change in $A_k\mathrm{G}_k$ and the initial scaled gap $A_1\mathrm{G}_1$.

Lemma 3.3. *Consider the iterates $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ evolving according to Algorithm 1. Let $n \geq 2$ and assume that $a_1 = \frac{1}{\sqrt{2}n^{1.5}}$ and $a_1 \geq (n-1)a_2$, while for $k \geq 3$,*

$$a_k \leq \min\Big(\frac{n a_{k-1}}{n-1}, \frac{\sqrt{A_{k-1}}}{2n}\Big). \tag{18}$$

Then, for fixed $\mathbf{u} \in \mathcal{X}$, any $\mathbf{v} \in \mathbb{R}^m$, and all $k \geq 2$, the gap estimate $\mathrm{G}_k = \mathrm{U}_k - \mathrm{L}_k$ satisfies

$$\begin{aligned}
&\mathbb{E}(A_k G_k(\mathbf{x}, \mathbf{y}) - A_{k-1}G_{k-1}(\mathbf{x}, \mathbf{y})) \\
&\leq -\mathbb{E}\left(\frac{A_k}{2}\|\mathbf{y} - \mathbf{y}_k\|_2^2 - \frac{A_{k-1}}{2}\|\mathbf{y} - \mathbf{y}_{k-1}\|_2^2\right) - \frac{1}{2}\mathbb{E}\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{\Lambda}}^2 + \frac{1}{2}\mathbb{E}\|\mathbf{x} - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2 \\
&\quad - a_k\mathbb{E}\langle \mathbf{A}(\mathbf{x} - \mathbf{x}_k), \mathbf{y}_k - \mathbf{y}_{k-1}\rangle + a_{k-1}\mathbb{E}\langle \mathbf{A}(\mathbf{x} - \mathbf{x}_{k-1}), \mathbf{y}_{k-1} - \mathbf{y}_{k-2}\rangle \\
&\quad - \frac{1}{4}A_{k-1}\mathbb{E}\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2 + \frac{1}{4}A_{k-2}\mathbb{E}\|\mathbf{y}_{k-1} - \mathbf{y}_{k-2}\|_2^2.
\end{aligned}$$

Lemma 3.4. *Given a fixed $\mathbf{u} \in \mathcal{X}$, any $\mathbf{v} \in \mathbb{R}^m$, $\bar{\mathbf{y}}_0 = \mathbf{y}_0$, and $\mathbf{x}_1$ and $\mathbf{y}_1$ from Algorithm 1, we have*

$$A_1\mathrm{G}_1(\mathbf{u}, \mathbf{v}) = a_1\langle \mathbf{A}^\top(\mathbf{y}_1 - \mathbf{y}_0), \mathbf{x}_1 - \mathbf{u}\rangle + \phi_0(\mathbf{u}) - \phi_0(\mathbf{x}_1) - \frac{1}{2}\|\mathbf{u} - \mathbf{x}_1\|_{\mathbf{\Lambda}}^2 - \frac{A_1}{2}\|\mathbf{v} - \mathbf{y}_1\|_2^2.$$

Combining the two lemmas, we now bound G_K and deduce our final result on the primal error.

Theorem 3.5. *[Main Result] Assume that $n \geq 4$. Given a matrix $\mathbf{A} \in \mathbb{R}_+^{m \times n}$, $\varepsilon > 0$, an arbitrary $\mathbf{x}_0 \in \mathcal{X}$ and $\bar{\mathbf{y}}_0 = \mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$, let $\mathbf{x}_k$ and A_k evolve according to SI-NNLS+ (Algorithm 1) for $k \geq 1$. For f defined in (2), define $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x} \geq \mathbf{0}} f(\mathbf{x})$. Then, for all $K \geq 2$, we have*

$$\mathbb{E}\Big[\langle \nabla f(\widetilde{\mathbf{x}}_K), \widetilde{\mathbf{x}}_K - \mathbf{x}^\star \rangle + \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}_K - \mathbf{x}^\star)\|^2\Big] \leq \frac{2\phi_0(\mathbf{x}^\star)}{A_K} = \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{\Lambda}}^2}{A_K}.$$

When $K \geq \frac{5}{2} n \log n$, we have $A_K \geq \frac{(K - \frac{5}{2} n \log n)^2}{36 n^2}$. If $\phi_0(\mathbf{x}^\star) \leq |f(\mathbf{x}^\star)|$, then for $K \geq \frac{5}{2} n \log n + \frac{6n}{\sqrt{\varepsilon}}$, we have $\mathbb{E}[f(\widetilde{\mathbf{x}}_K) - f(\mathbf{x}^\star)] \leq \varepsilon |f(\mathbf{x}^\star)|$. The total cost is $O\big(nnz(\mathbf{A})\big(\log n + \frac{1}{\sqrt{\varepsilon}}\big)\big)$.

The assumption $\phi_0(\mathbf{x}^\star) \leq |f(\mathbf{x}^\star)|$ above is satisfied by $\mathbf{x}_0 = \mathbf{0}$ (c.f. Proposition 2.1 and Eq. (11)).

Remark 3.6. SI-NNLS+ (Algorithm 1) and Theorem 3.5 also generalize to a mini-batch version. Increasing the batch size grows our bounds and number of data passes by a factor of at most square-root of the batch size s, by relating the spectral norm of the s columns of $\mathbf{A}$ corresponding to a batch to the Euclidean norms of individual columns of $\mathbf{A}$ from the same batch. However, due to efficient available implementations of vector operations, mini-batch variants of our algorithm with small batch sizes can have lower total runtimes on some datasets (see Section 5).

4 Adaptive restart

We now describe how SI-NNLS+ can be combined with adaptive restart to obtain linear convergence rate. To apply the restart strategy, we need suitable upper and lower bounds on the measure of convergence rate. Our measure of optimality is the *natural residual* $r(\mathbf{x}) = \|\mathbf{R}(\mathbf{x})\|_{\mathbf{\Lambda}}$ [30] for

$$\mathbf{R}(\mathbf{x}) = \mathbf{x} - \Pi_{\mathbb{R}_+^n}(\mathbf{x} - \mathbf{\Lambda}^{-1}\nabla f(\mathbf{x})) = \mathbf{x} - (\mathbf{x} - \mathbf{\Lambda}^{-1}\nabla f(\mathbf{x}))_+, \tag{19}$$

where $\Pi_{\mathbb{R}_+^n}$ is the projection operator onto $\mathbb{R}_+^n$ and $\mathbf{\Lambda}$ is as defined in Section 2. For $\mathbf{\Lambda} = \mathbf{I}$, $\mathbf{R}(\mathbf{x})$ is the natural map as defined in, e.g., [15]. Due to space constraints, we only state the main result of this section in the following theorem, while full technical details are deferred to Section 6.4.

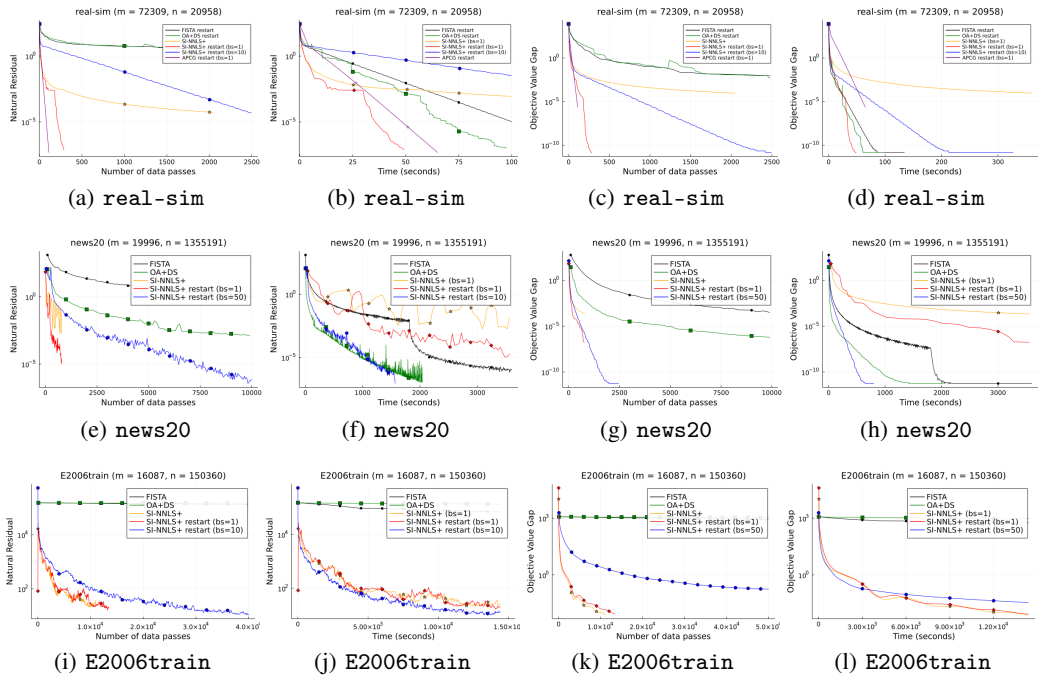
Theorem 4.1. *Given an error parameter $\varepsilon > 0$ and $\mathbf{x}_0 = \mathbf{0}$, consider the following algorithm $\mathcal{A}$:*

> $\mathcal{A}$: ***SI-NNLS+ with Restarts***
> *Initialize:* $k = 1$.
> *Initialize Lazy SI-NNLS+ at* $\mathbf{x}_{k-1}$.
> *Run Lazy SI-NNLS+ until the output* $\widetilde{\mathbf{x}}_K^k$ *satisfies* $r(\widetilde{\mathbf{x}}_K^k) \leq \frac{1}{2} r(\mathbf{x}_{k-1})$.
> *Restart Lazy SI-NNLS+ initializing at* $\mathbf{x}_k = \widetilde{\mathbf{x}}_K^k$.
> *Increment* k.
> *Repeat until* $r(\widetilde{\mathbf{x}}_K^k) \leq \varepsilon$.

Then, the expected number of arithmetic operations of $\mathcal{A}$ is $O\big(\mathrm{nnz}(\mathbf{A})\big(\log n + \frac{\sqrt{n}}{\mu}\big) \log\big(\frac{r(\mathbf{x}_0)}{\varepsilon}\big)\big)$. As a consequence, given $\bar{\varepsilon} > 0$, the total expected number of arithmetic operations until a point with $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \bar{\varepsilon}|f(\mathbf{x}^\star)|$ can be constructed by $\mathcal{A}$ is $O\Big(\mathrm{nnz}(\mathbf{A})\Big(\log n + \frac{\sqrt{n}}{\mu}\Big) \log\Big(\frac{n}{\mu\bar{\varepsilon}}\Big)\Big)$.

5 Numerical experiments and discussion

We conclude our paper by presenting the numerical performance of SI-NNLS+ and its restart versions (see the efficient implementation version in Algorithm 2) against FISTA with restart [5, 37], a general-purpose large-scale optimization algorithm, OA+DS with restart designed by [22] specifically for large-scale NNLS problems, and APCG [27] with restart. As an accelerated algorithm, FISTA has the optimal $1/k^2$ convergence rate; OA+DS, while often efficient in practice, has only an asymptotic convergence guarantee. For FISTA and APCG, we compute the tightest Lipschitz constant (i.e., the spectral norm $\|\mathbf{A}\|$); for OA+DS, we follow the best practices laid out by [22]. For our SI-NNLS+ algorithm and its restart version with batch size bs $= 1$, we follow Algorithm 2 and the restart strategy



(a) `real-sim` (b) `real-sim` (c) `real-sim` (d) `real-sim`

(e) `news20` (f) `news20` (g) `news20` (h) `news20`

(i) `E2006train` (j) `E2006train` (k) `E2006train` (l) `E2006train`

Figure 1: Algorithm comparison for NNLS+ on `real-sim`, `news20` and `E2006train` datasets.

in Section 4.[4] For the restart version with batch size larger than 1, we choose the best batch size in $\{10, 50, 300, 500\}$ and compute the block coordinate Lipschitz constants as the spectral norms of the corresponding block matrices. All algorithms were implemented in Julia.

We evaluated the performance of the algorithms on the large-scale sparse datasets `real-sim`, `news20`, and `E2006train` from the LibSVM library [9]. Both `real-sim` and `news20` datasets have non-negative data matrices, but the labels may be negative. When there exist negative labels, it is possible for the elements of $\mathbf{A}^\top \mathbf{b}$ to be negative. In such cases, per the discussion from the introduction, we can simply remove the corresponding columns of $\mathbf{A}$ and solve an equivalent problem with smaller dimension. On the other hand, the data matrix in `E2006train` dataset is not non-negative, which means that this dataset does not satisfy the assumption required for the analysis of Algorithm 1. However, Algorithm 1 can still be run by keeping only the non-negativity constraints for primal updates. This example is provided solely for illustration of empirical performance.

Results. To compare all implemented algorithms, we plot the natural residual/objective value gap versus number of data passes/time in Figure 1. Figure 1(a)-(d) shows that SI+NNLS+ is better than FISTA and OA+DS in terms of number of data passes on the `real-sim` dataset and better than APCG in terms of time. Note that both OA+DS and FISTA with restart have only sublinear convergence, but all the SI-NNLS+ algorithms with restart strategy achieve linear convergence and thus are much faster in obtaining a high-accuracy solution. Specifically, with bs $= 1$, we have a much better coordinate Lipschitz constant than the case of bs $= 10$ and thus the case of bs $= 1$ dominates bs $= 10$. As FISTA and OA+DS take less time accessing the full dataset once, they have lower runtimes than SI-NNLS+ but are beaten by SI-NNLS+ with restart and bs $= 1$. In Figure 1(e)-(h), on the `news20` dataset, in terms of number of data passes, SI-NNLS+ and its restart version with bs $= 1$ are dominant. However, as `news20` is a very sparse dataset, letting bs $= 1$ significantly increases the total time to access the full data once due to the overhead per iteration. As a result, both the SI-NNLS+ with bs $= 1$ and its restart version have the worst runtimes, while SI-NNLS+ with bs $= 10$ still attains the best runtime. Figure 1(i)-(l) shows the performance comparison on the `E2006train` dataset. On this dataset, both FISTA and OA+DS ran for 4 hours without visibly reducing the function value, whereas our (block) coordinate algorithm outperforms them in both number of data passes and time. Further experiments are left for future work.

[4] The algorithm is implemented for the non-scaled version of the problem, (P); see Section 2.

References

[1] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly linear-time packing and covering lp solvers. *Mathematical Programming*, 175(1):307–353, 2019.

[2] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119. PMLR, 2016.

[3] Anders H Andersen and William S Rayens. Structure-seeking multilinear methods for the analysis of fmri data. *NeuroImage*, 22(2):728–739, 2004.

[4] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

[5] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[6] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

[7] Rasmus Bro and Sijmen Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11:393–401, 09 1997.

[8] Alfred M Bruckstein, Michael Elad, and Michael Zibulevsky. On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Transactions on Information Theory*, 54(11):4813–4820, 2008.

[9] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[10] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[11] Monica Dessole, Marco Dell'Orto, and Fabio Marcuzzi. The lawson-hanson algorithm with deviation maximization: Finite convergence and sparse recovery. *arXiv preprint arXiv:2108.05345*, 2021.

[12] Jelena Diakonikolas and Lorenzo Orecchia. Alternating randomized block coordinate descent. In *International Conference on Machine Learning*, pages 1224–1232. PMLR, 2018.

[13] Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019.

[14] Jelena Diakonikolas, Maryam Fazel, and Lorenzo Orecchia. Fair packing and covering on a relative scale. *SIAM Journal on Optimization*, 30(4):3284–3314, 2020.

[15] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[16] Simon Foucart and David Koslicki. Sparse recovery by means of nonnegative least squares. *IEEE Signal Processing Letters*, 21(4):498–502, 2014.

[17] Nicolas Gillis. The why and how of nonnegative matrix factorization. *Connections*, 12:2–2, 2014.

[18] Takashi Isobe, Eric D Feigelson, Michael G Akritas, and Gutti Jogesh Babu. Linear regression in astronomy. *The astrophysical journal*, 364:104–113, 1990.

[19] Ethan C Jackson, James Alexander Hughes, and Mark Daley. On the generalizability of linear and non-linear region of interest-based multivariate regression models for fmri data. In *2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8. IEEE, 2018.

[20] Arun Jambulapati, Jerry Li, and Kevin Tian. Robust sub-gaussian principal component analysis and width-independent schatten packing. *Advances in Neural Information Processing Systems*, 33:15689–15701, 2020.

[21] Michael S Jennis and Joyce L Peabody. Pulse oximetry: an alternative method for the assessment of oxygenation in newborn infants. *Pediatrics*, 79(4):524–528, 1987.

[22] Dongmin Kim, Suvrit Sra, and Inderjit S Dhillon. A non-monotonic method for large-scale non-negative least squares. *Optimization Methods and Software*, 28(5):1012–1039, 2013.

[23] Richard Kueng and Peter Jung. Robust nonnegative sparse recovery and the nullspace property of 0/1 measurements. *IEEE Transactions on Information Theory*, 64(2):689–703, 2018. doi: 10.1109/TIT.2017.2746620.

[24] Abhishek Kumar, Vikas Sindhwani, and Prabhanjan Kambadur. Fast conical hull algorithms for near-separable non-negative matrix factorization. In *International Conference on Machine Learning*, pages 231–239. PMLR, 2013.

[25] Charles L. Lawson and Richard J. Hanson. *Solving least squares problems*, volume 15 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. ISBN 0-89871-356-0. Revised reprint of the 1974 original.

[26] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 ieee 54th annual symposium on foundations of computer science*, pages 147–156. IEEE, 2013.

[27] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. *Advances in Neural Information Processing Systems*, 27:3059–3067, 2014.

[28] Michael Luby and Noam Nisan. A parallel approximation algorithm for positive linear programming. In *Proc. ACM STOC'93*, 1993.

[29] Michael W Mahoney, Satish Rao, Di Wang, and Peng Zhang. Approximating the solution to mixed packing and covering LPs in parallel $\tilde{O}(\epsilon^{-3})$ time. In *Proc. ICALP'16*, 2016.

[30] Olvi L. Mangasarian and J Ren. New improved error bounds for the linear complementarity problem. *Mathematical Programming*, 66(1):241–255, 1994.

[31] Jelena Marasevic, Clifford Stein, and Gil Zussman. A fast distributed stateless algorithm for alpha-fair packing problems. In *Proc. ICALP'16*, 2016.

[32] Eduardo Martínez-Montes, José M Sánchez-Bornot, and Pedro A Valdés-Sosa. Penalized parafac analysis of spontaneous eeg recordings. *Statistica Sinica*, pages 1449–1464, 2008.

[33] Joe M Myre, Erich Frahm, David J Lilja, and Martin O Saar. TNT-NN: A fast active set method for solving large non-negative least squares problems. *Procedia Computer Science*, 108:755–764, 2017.

[34] Arkadi S Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992.

[35] Arkadii Semenovich Nemirovsky and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. 1983.

[36] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[37] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1):125–161, 2013.

[38] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Doklady AN SSSR (translated as Soviet Mathematics Doklady)*, volume 269, pages 543–547, 1983.

[39] Yurii Nesterov and Sebastian U Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

[40] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[41] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

[42] Yonatan Shadmi, Peter Jung, and Giuseppe Caire. Sparse non-negative recovery from biased subgaussian measurements using nnls. *arXiv preprint arXiv:1901.05727*, 2019.

[43] Martin Slawski and Matthias Hein. Non-negative least squares for high-dimensional linear models: consistency and sparse recovery without regularization, 2014.

[44] Chaobing Song, Stephen J Wright, and Jelena Diakonikolas. Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums. In *International Conference on Machine Learning*, 2021.

[45] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[46] Mark H Van Benthem and Michael R Keenan. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(10):441–450, 2004.

[47] Meng Wang, Weiyu Xu, and Ao Tang. A unique "nonnegative" solution to an underdetermined system: From vectors to matrices. *IEEE Transactions on Signal Processing*, 59(3):1007–1016, 2011. doi: 10.1109/TSP.2010.2089624.

[48] Michael W Wukitsch, Michael T Petterson, David R Tobler, and Jonas A Pologe. Pulse oximetry: analysis of theory, technology, and practice. *Journal of clinical monitoring*, 4(4):290–301, 1988.

[49] Neal Young. Sequential and parallel algorithms for mixed packing and covering. In *Proc. IEEE FOCS'01*, 2001.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We summarized the contributions and reviewed related work in Introduction. Section 3 and 6 contain the full technical details for our theoretical results. Section 5 provides full details for numerical experiments, while Section 7 provides full technical details for the implementation version of our algorithm, including the associated discussion about maintaining the claimed theoretical guarantees.
   (b) Did you describe the limitations of your work? [Yes] The main examples are: (i) that similar to ours (although weaker) results can be obtained with existing accelerated coordinate descent algorithms once the structural properties we prove are taken into account, in Introduction (Section 1); (ii) the discrepancy between the theoretical and empirical guarantee depending on the batch size as a consequence of modern processor architecture, in Remark 3.6 and Section 5; and (iii) that further experiments are left for future work, in Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a theoretical optimization paper for generic non-negative least squares problems, and as such it is not tied to any specific application that could lead to a negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] We stated all the assumptions in Section 2.

(b) Did you include complete proofs of all theoretical results? [Yes] We provided a detailed technical overview in Sections 3 and 4, and included complete proofs in Sections 6 and 7.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We included the code in the supplementary material. The data source used in this paper is cited in Section 5.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We explained the data preprocessing in Section 5.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] The performance of all the algorithms was consistent over many runs and thus the error bars would not have been particularly informative.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We included this information in Section 5.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Yes, we cited [9]; see Section 5.

(b) Did you mention the license of the assets? [N/A] The datasets we use are from LibSVM, which is an open source library under the 3-clause BSD license.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included the code in the supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The data we use is from LibSVM, which is an open resource database.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] According to the data's description, there is no such issue.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

6 Appendix: Omitted Technical Details

In this section, we provide proofs for all the claims made in the main body of the paper, additional supporting propositions and lemmas, and also omitted details of the implementation of our algorithm. The proofs are provided in the order in which the corresponding statement appears in the main body. In Section 6.1, we prove the properties of our problem. We again emphasize that these properties are crucial to achieving our goal of a scale-invariant algorithm for (P). Section 6.2 contains the proofs of all our results pertaining to the convergence analysis of Algorithm 1, with proofs of the growth rate of the scalar sequences $\{a_i\}, \{a_i^k\}$, and A_k grouped separately in Section 6.3, owing to their more technical nature.

6.1 Omitted Proof from Section 2: Properties of Our Objective

Proposition 2.1. *Given $f : \mathbb{R}^n_+ \to \mathbb{R}$ as defined in (2) and $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^n_+} f(\mathbf{x})$, the following statements all hold.*

a) $\nabla f(\mathbf{x}^\star) \geq \mathbf{0}$.
b) $f(\mathbf{x}^\star) = -\frac{1}{2}\|\mathbf{A}\mathbf{x}^\star\|_2^2 = -\frac{1}{2}\mathbf{1}^\top\mathbf{x}^\star$.
c) for all $j \in [n]$, we have $x_j^\star \in \left[0, \frac{1}{\|\mathbf{A}_{:j}\|_2^2}\right]$.
d) $-\frac{1}{2}\sum_{j\in[n]} \frac{1}{\|\mathbf{A}_{:j}\|_2^2} \leq f(\mathbf{x}^\star) \leq -\frac{1}{2\min_{j\in[n]}\|\mathbf{A}_{:j}\|_2^2}$.

Proof. We recall the first-order optimality condition stated in Inequality (1): for all $\mathbf{x} \geq \mathbf{0}$, we have $\langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star\rangle \geq 0$; we repeatedly invoke this inequality in the proof below.

1. Suppose there exists a coordinate j at which Proposition 2.1 (a) does not hold and instead, we have $\nabla_j f(\mathbf{x}^\star) < 0$. Consider $\mathbf{x} \geq \mathbf{0}$ such that $x_i = x_i^\star$ for all $i \neq j$ and let $x_j = x_j^\star + \varepsilon$ for some $\varepsilon > 0$. Then, Inequality (1) becomes $\nabla_j f(\mathbf{x}^\star) \cdot \varepsilon \geq 0$. Under the assumption $\nabla_j f(\mathbf{x}^\star) < 0$, this is an invalid inequality, thus contradicting our assumption.

2. From Proposition 2.1 (a), we know that $\nabla f(\mathbf{x}^\star) \geq \mathbf{0}$. If $\nabla_i f(\mathbf{x}^\star) > 0$, and if $x_i^\star > 0$, then by picking a vector $\mathbf{x}$ such that $x_j = x_j^\star$ for $j \neq i$ and $x_i = x_i^\star - \gamma$ for any $\gamma \in (0, x_i^\star)$, we violate Inequality (1). Therefore it must be the case that if $\nabla_i f(\mathbf{x}^\star) > 0$, then $x_i^\star = 0$. Thus we have
$$0 = \langle \mathbf{x}^\star, \nabla f(\mathbf{x}^\star)\rangle = \langle \mathbf{x}^\star, \mathbf{A}^\top\mathbf{A}\mathbf{x}^\star - \mathbf{1}\rangle.$$
Therefore, $f(\mathbf{x}^\star) = \frac{1}{2}\langle \mathbf{x}^\star, \mathbf{A}^\top\mathbf{A}\mathbf{x}^\star\rangle - \mathbf{1}^\top\mathbf{x}^\star = -\frac{1}{2}\langle \mathbf{x}^\star, \mathbf{A}^\top\mathbf{A}\mathbf{x}^\star\rangle = -\frac{1}{2}\mathbf{1}^\top\mathbf{x}^\star$.

3. From the proof of Proposition 2.1 (b), we have $\langle \mathbf{x}^\star, \nabla f(\mathbf{x}^\star)\rangle = 0$. We also have $\mathbf{x}^\star \geq \mathbf{0}$ and, from Proposition 2.1 (a), that $\nabla f(\mathbf{x}^\star) \geq \mathbf{0}$. Therefore, if $\mathbf{x}_i^\star > 0$ for some coordinate i then it must be that $\nabla_i f(\mathbf{x}^\star) = 0$. That is, $1 = \langle \mathbf{A}_{:i}, \mathbf{A}\mathbf{x}^\star\rangle$. Combining this equality with the fact that $\mathbf{A}$ and $\mathbf{x}^\star$ are both coordinate-wise non-negative gives
$$1 = \langle \mathbf{A}_{:i}, \mathbf{A}\mathbf{x}^\star\rangle \geq \langle \mathbf{A}_{:i}, \mathbf{A}_{:i}x_i^\star\rangle,$$
which implies $x_i^\star \leq \frac{1}{\|\mathbf{A}_{:i}\|_2^2}$ for all coordinates i.

4. The lower bound follows immediately by combining Proposition 2.1 (b) and Proposition 2.1 (c). For the upper bound, we need to find *a* feasible point $\widehat{\mathbf{x}}$ and compute the function value at $\widehat{\mathbf{x}}$, since $f(\mathbf{x}^\star) = \min_{\mathbf{y}\geq\mathbf{0}} f(\mathbf{y}) \leq f(\widehat{\mathbf{x}})$. Let $\widehat{\mathbf{x}} = \gamma\mathbf{e}_k$ for some $\gamma > 0$. Then,
$$f(\widehat{\mathbf{x}}) = \frac{1}{2}\gamma^2\|\mathbf{A}_{:k}\|_2^2 - \gamma.$$
Let $\gamma = \frac{1}{\|\mathbf{A}_{:k}\|_2^2}$. Then, $f(\widehat{\mathbf{x}}) = -\frac{1}{2\|\mathbf{A}_{:k}\|_2^2}$. We pick $k = \arg\min_{i\in[n]}\|\mathbf{A}_{:i}\|_2$, therefore $f(\mathbf{x}^\star) \leq -\frac{1}{2\min_{i\in[n]}\|\mathbf{A}_{:i}\|_2^2}$ as claimed.

□

6.2 Omitted Proofs from Section 3: Analysis of Algorithm

6.2.1 Proofs from Section 3.1: Results on Upper and Lower Estimates

We first show the results stating U_k and L_k are indeed valid upper and lower (respectively) estimates of the Lagrangian.

Lemma 3.1. *For U_k as defined in Eq. (10), Lagrangian defined in Eq. (3) and $\widetilde{\mathbf{x}}_k \in \mathbb{R}^n_+$ in Eq. (8), we have, for all $\mathbf{y} \in \mathbb{R}^m$, the upper bound $\mathrm{U}_k(\mathbf{y}) \geq \mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{y})$.*

Proof. By evaluating the Lagrangian described by Eq. (3) at $\mathbf{x} = \widetilde{\mathbf{x}}_k$, and by definition of ψ_k from Eq. (9), we obtain the following upper bound on the Lagrangian at $(\widetilde{\mathbf{x}}_k, \mathbf{y})$.

$$\begin{aligned}
\mathcal{L}(\widetilde{\mathbf{x}}_k, \mathbf{y}) &= \langle \mathbf{A}\widetilde{\mathbf{x}}_k, \mathbf{y}\rangle - \frac{1}{2}\|\mathbf{y}\|_2^2 - \mathbf{1}^\top \widetilde{\mathbf{x}}_k \\
&= \frac{1}{A_k}\sum_{i\in[k]} a_i^k \left[\langle \mathbf{A}\mathbf{x}_i, \mathbf{y}\rangle - \frac{1}{2}\|\mathbf{y}\|^2 - \mathbf{1}^\top \mathbf{x}_i\right] = \frac{1}{A_k}\psi_k(\mathbf{y}) \\
&\leq \frac{1}{A_k}\psi_k(\mathbf{y}_k) - \frac{1}{2}\|\mathbf{y} - \mathbf{y}_k\|_2^2 = \mathrm{U}_k(\mathbf{y}),
\end{aligned}$$

where the final steps are by strong convexity of ψ_k and by Eq. (10). □

We emphasize that $\mathbf{y}$ here can be random since this is a deterministic statement.

Lemma 3.2. *For L_k defined in Eq. (17), for the Lagrangian in Eq. (3) and $\widetilde{\mathbf{y}}_k$ in Eq. (8), we have, for a fixed $\mathbf{u} \in \mathcal{X}$, the lower bound $\mathbb{E}\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) \geq \mathbb{E}\mathrm{L}_k(\mathbf{u})$, where the expectation is with respect to all the random choices of coordinates in Algorithm 1.*

Proof. First, evaluating Eq. (3) at $\widetilde{\mathbf{y}}_k$ gives

$$\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) = \langle \mathbf{A}\mathbf{u}, \widetilde{\mathbf{y}}_k\rangle - \mathbf{1}^\top \mathbf{u} - \frac{1}{2}\|\widetilde{\mathbf{y}}_k\|_2^2.$$

Taking the expectation on both sides, applying the definition of $\widetilde{\mathbf{y}}_k$, convexity of $\frac{1}{2}\|\cdot\|^2$, and Jensen's inequality, and adding and subtracting $\frac{1}{A_k}\mathbb{E}\sum_{i\in[k]} a_i\langle \mathbf{A}\mathbf{x}, \overline{\mathbf{y}}_{i-1}\rangle + \frac{1}{A_k}\phi_0(\mathbf{u})$ gives

$$\begin{aligned}
\mathbb{E}\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) &\geq \frac{1}{A_k}\mathbb{E}\left[\sum_{i\in[k]} a_i\left[\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i\rangle - \mathbf{1}^\top\mathbf{u} - \frac{1}{2}\|\mathbf{y}_i\|_2^2\right]\right] \\
&= \frac{1}{A_k}\mathbb{E}\left[\sum_{i\in[k]} a_i\left[\langle \mathbf{A}\mathbf{u}, \overline{\mathbf{y}}_{i-1}\rangle - \mathbf{1}^\top\mathbf{u} - \frac{1}{2}\|\mathbf{y}_i\|_2^2\right]\right] + \frac{1}{A_k}\mathbb{E}\left[\phi_0(\mathbf{u})\right] - \frac{1}{A_k}\mathbb{E}\left[\phi_0(\mathbf{u})\right] \\
&\quad + \frac{1}{A_k}\mathbb{E}\left[\sum_{i\in[k]} a_i\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i - \overline{\mathbf{y}}_{i-1}\rangle\right].
\end{aligned}$$

We continue the analysis as

$$\begin{aligned}
\mathbb{E}\mathcal{L}(\mathbf{u}, \widetilde{\mathbf{y}}_k) &\geq \frac{1}{A_k}\mathbb{E}\left[\phi_k(\mathbf{u})\right] - \frac{1}{A_k}\mathbb{E}\left[\phi_0(\mathbf{u})\right] + \frac{1}{A_k}\sum_{i\in[k]} a_i\mathbb{E}\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i - \overline{\mathbf{y}}_{i-1}\rangle - \frac{1}{2A_k}\mathbb{E}\sum_{i\in[k]} a_i\|\mathbf{y}_i\|_2^2 \\
&\geq \frac{1}{A_k}\mathbb{E}\left[\phi_k(\mathbf{x}_k)\right] + \mathbb{E}\left[\frac{1}{2A_k}\|\mathbf{u} - \mathbf{x}_k\|_{\mathbf{\Lambda}}^2\right] - \frac{1}{A_k}\mathbb{E}[\phi_0(\mathbf{u})] + \frac{1}{A_k}\mathbb{E}\left[\sum_{i\in[k]} a_i\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i - \overline{\mathbf{y}}_{i-1}\rangle\right] \\
&\quad - \frac{1}{2A_k}\mathbb{E}\sum_{i\in[k]} a_i\|\mathbf{y}_i\|_2^2 \\
&= \mathbb{E}\mathrm{L}_k(\mathbf{u}),
\end{aligned}$$

the first step comes from Eq. (13), the second step comes from Eq. (16), and the final step comes from Eq. (17). □

6.2.2 Proofs from Section 3.2

We now describe three technical propositions that bound terms that show up in the proof of our result on bounding the scaled gap estimate.

Proposition 6.1. *For ψ_k defined in Eq. (9), with $\{a_i^k\}$ defined in Eq. (7), we have for all $k \geq 1$,*

$$\begin{aligned}\psi_k(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_{k-1}) \leq a_k^k &\left\{ \langle \mathbf{y}_k, \mathbf{A}\mathbf{x}_k \rangle - \frac{1}{2}\|\mathbf{y}_k\|_2^2 - \mathbf{1}^\top \mathbf{x}_k \right\} \\ &+ \sum_{i=1}^{k-1} (a_i^k - a_i^{k-1}) \left[\langle \mathbf{y}_k, \mathbf{A}\mathbf{x}_i \rangle - \frac{1}{2}\|\mathbf{y}_k\|_2^2 - \mathbf{1}^\top \mathbf{x}_i \right] \\ &- \frac{A_{k-1}}{2}\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2.\end{aligned}$$

Proof. Evaluating ψ_k and ψ_{k-1} as defined in Eq. (9) at $\mathbf{y}_k$ and subtracting, we have

$$\psi_k(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_k) = a_k \langle \mathbf{A}^\top \mathbf{y}_k - \mathbf{1}, n\mathbf{x}_k - (n-1)\mathbf{x}_{k-1} \rangle - \frac{a_k}{2}\|\mathbf{y}_k\|_2^2. \tag{20}$$

Next, applying strong convexity of ψ_{k-1} at $\mathbf{y}_k$ and $\mathbf{y}_{k-1}$ while using the fact that $\mathbf{y}_{k-1}$ minimizes ψ_{k-1} gives

$$\psi_{k-1}(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_{k-1}) \leq -\frac{1}{2}A_{k-1}\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2. \tag{21}$$

To complete the proof, it remains to add Eq. (20) and Inequality (21). □

Proposition 6.2. *The random function $\phi_k : \mathcal{X} \to \mathbb{R}$, $k \geq 2$, defined in Eq. (15) satisfies the following properties, with $\mathbf{x}_k$, $\mathbf{y}_k$, and $\overline{\mathbf{y}}_k$ evolving as per Algorithm 1.*

a) It is separable in its coordinates: $\phi_k(\mathbf{x}) = \sum_{j\in[n]} \phi_{k,j}(x_j)$, where, for each $j \in [n]$, we define $\phi_{0,j}(x_j) = \frac{\|\mathbf{A}_{:j}\|_2^2}{2}(x_j - [\mathbf{x}_0]_j)^2$, $\phi_{1,j}(x_j) = a_1 x_j (\mathbf{A}^\top \overline{\mathbf{y}}_0 - \mathbf{1})_j + \phi_{0,j}(x_j)$, and for $k \geq 2$,

$$\phi_{k,j}(x_j) = \phi_{k-1,j}(x_j) + n a_k \mathbf{1}_{j=j_k} \langle \mathbf{A}^\top \overline{\mathbf{y}}_{k-1} - \mathbf{1}, x_{j_k}\mathbf{e}_{j_k} \rangle. \tag{22}$$

b) The primal variable $\mathbf{x}_k$ is updated only on the j_k^{th} coordinate in each iteration: $\mathbf{x}_k = \mathbf{x}_{k-1} + \gamma \mathbf{e}_{j_k}$ for some γ, and $[\mathbf{x}_k]_j = [\mathbf{x}_{k-1}]_j$ for $j \neq j_k$.

c) For a fixed $\mathbf{x} \in \mathcal{X}$ and for $k \geq 1$, we have, over all the randomness in the algorithm,

$$\mathbb{E}\left[\phi_k(\mathbf{x})\right] = \mathbb{E}\left[\phi_0(\mathbf{x})\right] + \sum_{i\in[k]} a_i \mathbb{E}\left[\langle \mathbf{A}^\top \overline{\mathbf{y}}_{i-1} - \mathbf{1}, \mathbf{x} \rangle\right]. \tag{23}$$

Proof. In the statement of Proposition 6.2 (a), the claim about separability of ϕ_0 and ϕ_1 can be checked just from the definitions of $\phi_{0,j}$ and $\phi_{1,j}$. We prove the claim of coordinate-wise separability for $k \geq 2$ by summing over $j \in [n]$ both sides of Eq. (22). We can compute this sum via following observation, which concludes the proof of Proposition 6.2 (a).

$$\sum_{j\in[n]} a_k \mathbf{1}_{j=j_k} \langle \mathbf{A}^\top \overline{\mathbf{y}}_{k-1} - \mathbf{1}, x_{j_k}\mathbf{e}_{j_k} \rangle = a_k \langle \mathbf{A}^\top \overline{\mathbf{y}}_{k-1} - \mathbf{1}, x_{j_k}\mathbf{e}_{j_k} \rangle.$$

From Proposition 6.2 (a), we may therefore define, for $j \neq j_k$,

$$[\mathbf{x}_k]_j = \arg\min_{u\in\mathbb{R}_+} \phi_{k,j}(u) = \arg\min_{u\in\mathbb{R}_+} \phi_{k-1,j}(u) = [\mathbf{x}_{k-1}]_j.$$

Therefore, $[\mathbf{x}_k]_j = [\mathbf{x}_{k-1}]_j$ for all $j \neq j_k$, thus proving Proposition 6.2 (b). To prove Proposition 6.2 (c), we use induction on k. The base case holds for $k = 1$ by the definition of $\phi_1(\mathbf{x})$. Let Proposition 6.2 (c) hold for $k \geq 1$. Then, by the definition of ϕ_k as in Eq. (15), we have

$$\phi_k(\mathbf{x}) = \phi_{k-1}(\mathbf{x}) + n a_k \langle \mathbf{A}^\top \overline{\mathbf{y}}_{k-1} - \mathbf{1}, x_{j_k}\mathbf{e}_{j_k} \rangle, \text{ for all } k \geq 2.$$

Let $\mathcal{F}_{k-1}$ be the natural filtration, containing all the randomness in the algorithm up to and including iteration $k-1$. Taking expectations with respect to all the randomness until iteration k and invoking linearity of expectation, the inductive hypothesis, and the tower rule $\mathbb{E}[\,\cdot\,] = \mathbb{E}[\mathbb{E}[\,\cdot\,|\mathcal{F}_{k-1}]]$, we have

$$\begin{aligned}
\mathbb{E}[\phi_k(\mathbf{x})] &= \mathbb{E}[\phi_{k-1}(\mathbf{x})] + na_k\mathbb{E}\left[\left[\mathbb{E}\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, x_{j_k}\mathbf{e}_{j_k}\rangle|\mathcal{F}_{k-1}\right]\right] \\
&= \mathbb{E}\left[\phi_0(\mathbf{x})\right] + \sum_{i\in[k-1]} a_i\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{i-1} - \mathbf{1}, \mathbf{x}\rangle\right] + a_k\mathbb{E}\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, \mathbf{x}\rangle \\
&= \mathbb{E}\left[\phi_0(\mathbf{x})\right] + \sum_{i\in[k]} a_i\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{i-1} - \mathbf{1}, \mathbf{x}\rangle\right],
\end{aligned}$$

which finishes the proof of Proposition 6.2 (c). □

Proposition 6.3. *For all $k \geq 2$, the random function $\phi_k : \mathcal{X} \to \mathbb{R}$, $k \geq 2$, defined in Eq. (15) satisfies the following inequality, where $\mathbf{x}_k$ and $\overline{\mathbf{y}}_k$ evolve according to Algorithm 1.*

$$\phi_k(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_{k-1}) \geq a_k\left(n\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, [\mathbf{x}_k]_{j_k}\mathbf{e}_{j_k}\rangle\right) + \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2.$$

Proof. We have, using $\mathbf{x} = \mathbf{x}_k$ in Eq. (15), that

$$\phi_k(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_k) = na_k\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, [\mathbf{x}_k]_{j_k}\mathbf{e}_{j_k}\rangle.$$

Applying Eq. (16) to ϕ_{k-1} at $\mathbf{x}_k$ gives

$$\phi_{k-1}(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_{k-1}) \geq \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2.$$

Adding these inequalities completes the proof of the claim. □

We now use the preceding technical results to bound the change in scaled gap.

Lemma 3.3. *Consider the iterates $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ evolving according to Algorithm 1. Let $n \geq 2$ and assume that $a_1 = \frac{1}{\sqrt{2}n^{1.5}}$ and $a_1 \geq (n-1)a_2$, while for $k \geq 3$,*

$$a_k \leq \min\left(\frac{na_{k-1}}{n-1}, \frac{\sqrt{A_{k-1}}}{2n}\right). \tag{18}$$

Then, for fixed $\mathbf{u} \in \mathcal{X}$, any $\mathbf{v} \in \mathbb{R}^m$, and all $k \geq 2$, the gap estimate $\mathrm{G}_k = \mathrm{U}_k - \mathrm{L}_k$ satisfies

$$\begin{aligned}
&\mathbb{E}(A_kG_k(\mathbf{x},\mathbf{y}) - A_{k-1}G_{k-1}(\mathbf{x},\mathbf{y})) \\
&\leq -\mathbb{E}\left(\frac{A_k}{2}\|\mathbf{y}-\mathbf{y}_k\|_2^2 - \frac{A_{k-1}}{2}\|\mathbf{y}-\mathbf{y}_{k-1}\|_2^2\right) - \frac{1}{2}\mathbb{E}\|\mathbf{x}-\mathbf{x}_k\|_{\mathbf{\Lambda}}^2 + \frac{1}{2}\mathbb{E}\|\mathbf{x}-\mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2 \\
&\quad - a_k\mathbb{E}\langle \mathbf{A}(\mathbf{x}-\mathbf{x}_k), \mathbf{y}_k - \mathbf{y}_{k-1}\rangle + a_{k-1}\mathbb{E}\langle \mathbf{A}(\mathbf{x}-\mathbf{x}_{k-1}), \mathbf{y}_{k-1} - \mathbf{y}_{k-2}\rangle \\
&\quad - \frac{1}{4}A_{k-1}\mathbb{E}\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2 + \frac{1}{4}A_{k-2}\mathbb{E}\|\mathbf{y}_{k-1} - \mathbf{y}_{k-2}\|_2^2.
\end{aligned}$$

Proof. Using $\mathrm{G}_k = \mathrm{U}_k - \mathrm{L}_k$, U_k from Eq. (10), and L_k from Eq. (17), we have

$$\begin{aligned}
A_k\mathrm{G}_k(\mathbf{u},\mathbf{v}) &= \psi_k(\mathbf{y}_k) - \phi_k(\mathbf{x}_k) + \phi_0(\mathbf{u}) \\
&\quad - \frac{A_k}{2}\|\mathbf{v}-\mathbf{y}_k\|_2^2 - \frac{1}{2}\|\mathbf{u}-\mathbf{x}_k\|_{\mathbf{\Lambda}}^2 - \sum_{i\in[k]} a_i\langle \mathbf{A}\mathbf{u}, \mathbf{y}_i - \overline{\mathbf{y}}_{i-1}\rangle + \sum_{i\in[k]}\frac{a_i}{2}\|\mathbf{y}_i\|_2^2.
\end{aligned}$$

Therefore, the difference in scaled gap between successive iterations is

$$\begin{aligned}
A_k\mathrm{G}_k(\mathbf{u},\mathbf{v}) - A_{k-1}\mathrm{G}_{k-1}(\mathbf{u},\mathbf{v}) = {} & [\psi_k(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_{k-1})] - [\phi_k(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_{k-1})] \\
& - \frac{A_k}{2}\|\mathbf{v}-\mathbf{y}_k\|_2^2 + \frac{A_{k-1}}{2}\|\mathbf{v}-\mathbf{y}_{k-1}\|_2^2 \\
& - \frac{1}{2}\|\mathbf{u}-\mathbf{x}_k\|_{\mathbf{\Lambda}}^2 + \frac{1}{2}\|\mathbf{u}-\mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2 \\
& - a_k\langle \mathbf{A}\mathbf{u}, \mathbf{y}_k - \overline{\mathbf{y}}_{k-1}\rangle + \frac{a_k}{2}\|\mathbf{y}_k\|_2^2.
\end{aligned} \tag{24}$$

Based on the above expression, to prove the lemma, it suffices to bound the expectation of

$$T_k(\mathbf{u}) \stackrel{\text{def}}{=} [\psi_k(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_{k-1})] - [\phi_k(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_{k-1})] - a_k\langle \mathbf{A}\mathbf{u}, \mathbf{y}_k - \overline{\mathbf{y}}_{k-1}\rangle + \frac{a_k}{2}\|\mathbf{y}_k\|_2^2. \tag{25}$$

First, we take expectations on both sides of the inequality in Proposition 6.3 by invoking $\mathbb{E}[\,\cdot\,] = \mathbb{E}[\mathbb{E}[\,\cdot\,|\mathcal{F}_{k-1}]]$ as before, where $\mathcal{F}_{k-1}$ denotes the natural filtration. By using the fact that $\mathbf{x}_{k-1}$ is updated only at coordinate j_k (as stated in Proposition 6.2 (b)), we observe the following for the term from the right hand side of Proposition 6.3.

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, [\mathbf{x}_k]_{j_k}\mathbf{e}_{j_k}\rangle\right] \\
=&\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, \mathbf{x}_k - \mathbf{x}_{k-1}\rangle\right] + \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, [\mathbf{x}_{k-1}]_{j_k}\mathbf{e}_{j_k}\rangle|\mathcal{F}_{k-1}\right]\right] \\
=&\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, \mathbf{x}_k - \mathbf{x}_{k-1}\rangle\right] + \frac{1}{n}\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, \mathbf{x}_{k-1}\rangle\right] \\
=&\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, \mathbf{x}_k - (1-1/n)\,\mathbf{x}_{k-1}\rangle\right].
\end{aligned} \tag{26}$$

Therefore, we have from Proposition 6.3 and scaling Eq. (26) by $-na_k$ that

$$\begin{aligned}
-\mathbb{E}\left[\phi_k(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_{k-1})\right] \leq\ & -\frac{1}{2}\mathbb{E}\left[\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2\right] \\
& - a_k\mathbb{E}\left[\langle \mathbf{A}^\top\overline{\mathbf{y}}_{k-1} - \mathbf{1}, n\mathbf{x}_k - (n-1)\mathbf{x}_{k-1}\rangle\right].
\end{aligned} \tag{27}$$

We now bound the expectation of the term involving differences of ψ_k by taking expectations of both sides of Proposition 6.1.

$$\begin{aligned}
\mathbb{E}\left[\psi_k(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_{k-1})\right] \leq & -\frac{A_{k-1}}{2}\mathbb{E}\left[\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2\right] - \frac{a_k}{2}\mathbb{E}\left[\|\mathbf{y}_k\|_2^2\right] \\
& + a_k\mathbb{E}\left[\langle \mathbf{A}^\top\mathbf{y}_k - \mathbf{1}, n\mathbf{x}_k - (n-1)\mathbf{x}_{k-1}\rangle\right].
\end{aligned} \tag{28}$$

By taking expectations on both sides of Eq. (25), we have

$$\begin{aligned}
\mathbb{E}[T_k(\mathbf{u})] = & \mathbb{E}\left[\psi_k(\mathbf{y}_k) - \psi_{k-1}(\mathbf{y}_{k-1})\right] - \mathbb{E}\left[\phi_k(\mathbf{x}_k) - \phi_{k-1}(\mathbf{x}_{k-1})\right] \\
& - a_k\mathbb{E}\left[\langle \mathbf{A}\mathbf{u}, \mathbf{y}_k - \overline{\mathbf{y}}_{k-1}\rangle\right] + \frac{a_k}{2}\mathbb{E}\left[\|\mathbf{y}_k\|_2^2\right].
\end{aligned} \tag{29}$$

Combining Eq. (27), Eq. (28), and Eq. (29) then gives

$$\begin{aligned}
\mathbb{E}[T_k(\mathbf{u})] \leq & -\frac{A_{k-1}}{2}\mathbb{E}\left[\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2\right] - \frac{1}{2}\mathbb{E}\left[\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2\right] \\
& + a_k\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_k - \overline{\mathbf{y}}_{k-1}), n\mathbf{x}_k - (n-1)\mathbf{x}_{k-1} - \mathbf{u}\rangle\right].
\end{aligned} \tag{30}$$

Recall that by the assumption in the statement of the lemma,

$$\overline{\mathbf{y}}_{k-1} = \mathbf{y}_{k-1} + \frac{a_{k-1}}{a_k}(\mathbf{y}_{k-1} - \mathbf{y}_{k-2}).$$

Plugging into Eq. (30) and rearranging, we have

$$\begin{aligned}
\mathbb{E}[T_k(\mathbf{u})] \leq & -\frac{A_{k-1}}{2}\mathbb{E}\left[\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2\right] - \frac{1}{2}\mathbb{E}\left[\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2\right] \\
& + (n-1)a_k\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_k - \mathbf{y}_{k-1}), \mathbf{x}_k - \mathbf{x}_{k-1}\rangle\right] - na_{k-1}\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_{k-1} - \mathbf{y}_{k-2}), \mathbf{x}_k - \mathbf{x}_{k-1}\rangle\right] \\
& + a_k\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_k - \mathbf{y}_{k-1}), \mathbf{x}_k - \mathbf{u}\rangle\right] - a_{k-1}\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_{k-1} - \mathbf{y}_{k-2}), \mathbf{x}_{k-1} - \mathbf{u}\rangle\right].
\end{aligned} \tag{31}$$

To complete the proof, we need to bound the terms from the first two lines on the right-hand side of Eq. (31). First, observe that, by the coordinate update of $\mathbf{x}_k$ and Young's inequality, $\forall\beta > 0$,

$$\begin{aligned}
\langle \mathbf{A}^\top(\mathbf{y}_k - \mathbf{y}_{k-1}), \mathbf{x}_k - \mathbf{x}_{k-1}\rangle &= \langle \mathbf{y}_k - \mathbf{y}_{k-1}, \mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1})\rangle \\
&= \langle \mathbf{y}_k - \mathbf{y}_{k-1}, \mathbf{A}_{:j_k}([\mathbf{x}_k]_{j_k} - [\mathbf{x}_{k-1}]_{j_k})\rangle \\
&\leq \frac{\beta}{2}\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2 + \frac{1}{2\beta}\|\mathbf{A}_{:j_k}\|_2^2|[\mathbf{x}_k]_{j_k} - [\mathbf{x}_{k-1}]_{j_k}|^2 \\
&= \frac{\beta}{2}\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2 + \frac{1}{2\beta}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2.
\end{aligned} \tag{32}$$

By the same token, $\forall \gamma > 0$,

$$-\langle \mathbf{A}^\top(\mathbf{y}_{k-1} - \mathbf{y}_{k-2}), \mathbf{x}_k - \mathbf{x}_{k-1}\rangle \leq \frac{\gamma}{2}\|\mathbf{y}_{k-1} - \mathbf{y}_{k-2}\|_2^2 + \frac{1}{2\gamma}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\mathbf{\Lambda}}^2. \tag{33}$$

Recalling that, by the choice of step sizes, $(n-1)a_k \leq na_{k-1}$ and $a_k \leq \frac{\sqrt{A_{k-1}}}{2n}$, we can verify that for $\beta = 2(n-1)a_k$ and $\gamma = 2na_{k-1}$, the following inequalities hold:

$$\begin{aligned} &(n-1)a_k\beta - A_{k-1} \leq -\frac{A_{k-1}}{2}, \\ &\frac{(n-1)a_k}{\beta} + \frac{na_{k-1}}{\gamma} \leq 1. \end{aligned} \tag{34}$$

Combining Equations (31)–(34),

$$\begin{aligned} \mathbb{E}[T_k(\mathbf{u})] \leq &-\frac{A_{k-1}}{4}\mathbb{E}[\|\mathbf{y}_k - \mathbf{y}_{k-1}\|_2^2] + n^2 {a_{k-1}}^2\mathbb{E}[\|\mathbf{y}_{k-1} - \mathbf{y}_{k-2}\|_2^2] \\ &+ a_k\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_k - \mathbf{y}_{k-1}), \mathbf{x}_k - \mathbf{u}\rangle\right] - a_{k-1}\mathbb{E}\left[\langle \mathbf{A}^\top(\mathbf{y}_{k-1} - \mathbf{y}_{k-2}), \mathbf{x}_{k-1} - \mathbf{u}\rangle\right]. \end{aligned} \tag{35}$$

It remains to combine Eq. (24), Eq. (25), and Eq. (35). □

Lemma 3.4. *Given a fixed* $\mathbf{u} \in \mathcal{X}$*, any* $\mathbf{v} \in \mathbb{R}^m$*,* $\bar{\mathbf{y}}_0 = \mathbf{y}_0$*, and* $\mathbf{x}_1$ *and* $\mathbf{y}_1$ *from Algorithm 1, we have*

$$A_1\mathrm{G}_1(\mathbf{u}, \mathbf{v}) = a_1\langle \mathbf{A}^\top(\mathbf{y}_1 - \mathbf{y}_0), \mathbf{x}_1 - \mathbf{u}\rangle + \phi_0(\mathbf{u}) - \phi_0(\mathbf{x}_1) - \frac{1}{2}\|\mathbf{u} - \mathbf{x}_1\|_{\mathbf{\Lambda}}^2 - \frac{A_1}{2}\|\mathbf{v} - \mathbf{y}_1\|_2^2.$$

Proof. Evaluating Eq. (10) and Eq. (17) at $k = 1$ gives

$$A_1U_1(\mathbf{v}) = \psi_1(\mathbf{y}_1) - \frac{A_1}{2}\|\mathbf{v} - \mathbf{y}_1\|_2^2, \tag{36}$$

$$A_1L_1(\mathbf{u}) = \phi_1(\mathbf{x}_1) + \frac{1}{2}\|\mathbf{u} - \mathbf{x}_1\|_{\mathbf{\Lambda}}^2 - \phi_0(\mathbf{u}) + a_1\langle \mathbf{A}\mathbf{u}, \mathbf{y}_1 - \overline{\mathbf{y}}_0\rangle - \frac{a_1}{2}\|\mathbf{y}_1\|_2^2. \tag{37}$$

By definition of ψ_1 from Eq. (9), ϕ_1 from Eq. (12), and the assignment $a_1^1 = a_1$, we have

$$\begin{aligned} \psi_1(\mathbf{y}_1) - \phi_1(\mathbf{x}_1) &= -\frac{a_1}{2}\|\mathbf{y}_1\|_2^2 + a_1\langle \mathbf{y}_1, \mathbf{A}\mathbf{x}_1\rangle - a_1\mathbf{1}^\top\mathbf{x}_1 - \phi_0(\mathbf{x}_1) - a_1\langle \mathbf{A}^\top\overline{\mathbf{y}}_0 - \mathbf{1}, \mathbf{x}_1\rangle \\ &= -\frac{a_1}{2}\|\mathbf{y}_1\|_2^2 + a_1\langle \mathbf{A}(\mathbf{y}_1 - \mathbf{y}_0), \mathbf{x}_1\rangle - \phi_0(\mathbf{x}_1), \end{aligned}$$

where we have used that $\bar{\mathbf{y}}_0 = \mathbf{y}_0$, which holds by assumption. To complete the proof, it remains to subtract Eq. (37) from Eq. (36) and combine with the last equality. □

Theorem 3.5. *[Main Result] Assume that* $n \geq 4$*. Given a matrix* $\mathbf{A} \in \mathbb{R}_+^{m\times n}$*,* $\varepsilon > 0$*, an arbitrary* $\mathbf{x}_0 \in \mathcal{X}$ *and* $\bar{\mathbf{y}}_0 = \mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$*, let* $\mathbf{x}_k$ *and* A_k *evolve according to SI-NNLS+ (Algorithm 1) for* $k \geq 1$*. For* f *defined in (2), define* $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x}\geq\mathbf{0}} f(\mathbf{x})$*. Then, for all* $K \geq 2$*, we have*

$$\mathbb{E}\Big[\langle \nabla f(\widetilde{\mathbf{x}}_K), \widetilde{\mathbf{x}}_K - \mathbf{x}^\star\rangle + \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}_K - \mathbf{x}^\star)\|^2\Big] \leq \frac{2\phi_0(\mathbf{x}^\star)}{A_K} = \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{\Lambda}}^2}{A_K}.$$

When $K \geq \frac{5}{2}n\log n$*, we have* $A_K \geq \frac{(K - \frac{5}{2}n\log n)^2}{36n^2}$*. If* $\phi_0(\mathbf{x}^\star) \leq |f(\mathbf{x}^\star)|$*, then for* $K \geq \frac{5}{2}n\log n + \frac{6n}{\sqrt{\varepsilon}}$*, we have* $\mathbb{E}[f(\widetilde{\mathbf{x}}_K) - f(\mathbf{x}^\star)] \leq \varepsilon|f(\mathbf{x}^\star)|$*. The total cost is* $O\big(nnz(\mathbf{A})\big(\log n + \frac{1}{\sqrt{\varepsilon}}\big)\big)$*.*

Proof. Observe that, by the choice of step sizes, $n^2a_{k-1}^2 \leq \frac{A_{k-2}}{4}$, $\forall k \geq 3$. Thus, telescoping the bound in Lemma 3.3 and combining with Lemma 3.4, we have

$$\begin{aligned} &\mathbb{E}[A_K\mathrm{G}_K(\mathbf{u}, \mathbf{v})] \\ \leq &\phi_0(\mathbf{u}) - \frac{A_K}{2}\mathbb{E}[\|\mathbf{v} - \mathbf{y}_K\|_2^2] - \frac{1}{2}\mathbb{E}[\|\mathbf{u} - \mathbf{x}_K\|_{\mathbf{\Lambda}}^2] - a_K\mathbb{E}[\langle \mathbf{A}(\mathbf{u} - \mathbf{x}_K), \mathbf{y}_K - \mathbf{y}_{K-1}\rangle] \\ &- \frac{A_{K-1}}{4}\mathbb{E}[\|\mathbf{y}_K - \mathbf{y}_{K-1}\|_2^2] + n^2{a_1}^2\mathbb{E}[\|\mathbf{y}_1 - \mathbf{y}_0\|_2^2] - \mathbb{E}[\phi_0(\mathbf{x}_1)]. \end{aligned} \tag{38}$$

We first show how to cancel out the inner product term with the negative quadratic terms. Observe that, $\forall \beta > 0$,

$$\begin{aligned}-a_K\langle \mathbf{A}(\mathbf{u}-\mathbf{x}_K), \mathbf{y}_K - \mathbf{y}_{K-1}\rangle =& - a_K \sum_{j=1}^{n} (\mathbf{y}_K - \mathbf{y}_{K-1})^\top \mathbf{A}_{:j}(u_j - [\mathbf{x}_K]_j) \\ \leq& a_K\Big(\frac{n\beta}{2}\|\mathbf{y}_K - \mathbf{y}_{K-1}\|_2^2 + \frac{1}{2\beta}\|\mathbf{u}-\mathbf{x}_K\|_{\mathbf{\Lambda}}^2\Big),\end{aligned}$$

where the last line is by Young's inequality. In particular, choosing $\beta = 2a_K$, we have $\frac{1}{2}a_K n\beta = na_K^2$, which is at most $\frac{A_{K-1}}{4}$, by the choice of step sizes in SI-NNLS+. Thus, since $\phi_0(\mathbf{x}_1) = \frac{1}{2}\|\mathbf{x}_1 - \mathbf{x}_0\|_{\mathbf{\Lambda}}^2$, Eq. (38) simplifies to

$$\begin{aligned}&\mathbb{E}[A_K \mathrm{G}_K(\mathbf{u},\mathbf{v})] \\ \leq& \phi_0(\mathbf{u}) - \frac{A_K}{2}\mathbb{E}[\|\mathbf{v}-\mathbf{y}_K\|_2^2] - \frac{1}{4}\mathbb{E}[\|\mathbf{u}-\mathbf{x}_K\|_{\mathbf{\Lambda}}^2] + n^2{a_1}^2\mathbb{E}[\|\mathbf{y}_1-\mathbf{y}_0\|_2^2] - \mathbb{E}[\phi_0(\mathbf{x}_1)].\end{aligned} \tag{39}$$

Since $-\frac{1}{4}\mathbb{E}[\|\mathbf{u}-\mathbf{x}_K\|_{\mathbf{\Lambda}}^2] \leq 0$, we can ignore it. Let us now bound $n^2{a_1}^2\|\mathbf{y}_1 - \mathbf{y}_0\|_2^2 - \phi_0(\mathbf{x}_1)$. By definition of $\mathbf{x}_1$ and ϕ_1, we have $\mathbf{x}_1 = \mathbf{x}_0 - a_1\mathbf{\Lambda}^{-1}(\mathbf{A}^\top\mathbf{y}_0 - \mathbf{1})$. Further, from Eq. (9) and Eq. (6), as we have $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$, we can simplify the terms to bound as follows.

$$\begin{aligned}&n^2a_1^2\|\mathbf{y}_1-\mathbf{y}_0\|_2^2 - \phi_0(\mathbf{x}_1) \\ =& {a_1}^2\Big(n^2{a_1}^2\|\mathbf{A}\mathbf{\Lambda}^{-1}(\mathbf{A}^\top\mathbf{y}_0-\mathbf{1})\|_2^2 - \frac{1}{2}\|\mathbf{\Lambda}^{-1}(\mathbf{A}^\top\mathbf{y}_0-\mathbf{1})\|_{\mathbf{\Lambda}}^2\Big) \\ \leq& {a_1}^2\Big(n^2{a_1}^2\|\mathbf{\Lambda}^{-1/2}\mathbf{A}^\top\mathbf{A}\mathbf{\Lambda}^{-1/2}\|_2\|\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}^\top\mathbf{y}_0-\mathbf{1})\|_2^2 - \frac{1}{2}\|\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}^\top\mathbf{y}_0-\mathbf{1})\|_2^2\Big) \\ \leq& {a_1}^2\|\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}^\top\mathbf{y}_0-\mathbf{1})\|_2^2\Big(n^3{a_1}^2 - \frac{1}{2}\Big),\end{aligned}$$

where the reasoning behind the first inequality follows from the definition of spectral norm and that $\|\mathbf{A}\mathbf{\Lambda}^{-1/2}\|_2^2 = \lambda_{\max}(\mathbf{\Lambda}^{-1/2}\mathbf{A}^\top\mathbf{A}\mathbf{\Lambda}^{-1/2})$; the last inequality follows as the matrix $\mathbf{\Lambda}^{-1/2}\mathbf{A}^\top\mathbf{A}\mathbf{\Lambda}^{-1/2}$ has all ones on the main diagonal, and thus its trace is at most n, and since it is positive semidefinite, its spectral norm is at most its trace. As $a_1 = \frac{1}{\sqrt{2}n^{1.5}}$, we conclude that $n^2{a_1}^2\|\mathbf{y}_1 - \mathbf{y}_0\|_2^2 - \phi_0(\mathbf{x}_1) \leq 0$. Thus, Eq. (39) simplifies to

$$\mathbb{E}[A_K \mathrm{G}_K(\mathbf{u},\mathbf{v})] \leq \mathbb{E}[\phi_0(\mathbf{u})] - \frac{A_K}{2}\mathbb{E}[\|\mathbf{v}-\mathbf{y}_K\|_2^2]. \tag{40}$$

By construction, $\mathrm{Gap}_{\mathcal{L}}^{\mathbf{u},\mathbf{v}}(\widetilde{\mathbf{x}}_K, \widetilde{\mathbf{y}}_K) \leq \mathrm{G}_K(\mathbf{u},\mathbf{v})$. Further, by Inequality (5), $f(\widetilde{\mathbf{x}}_K) - f(\mathbf{x}^\star) + \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}_K - \mathbf{x}^\star)\|^2 \leq \mathrm{Gap}_{\mathcal{L}}^{\mathbf{u},\mathbf{v}}(\widetilde{\mathbf{x}}_K, \widetilde{\mathbf{y}}_K)$. Hence, we can conclude from Eq. (40) that

$$\mathbb{E}\Big[f(\widetilde{\mathbf{x}}_K) - f(\mathbf{x}^\star) + \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}_K - \mathbf{x}^\star)\|^2\Big] \leq \frac{\phi_0(\mathbf{x}^\star)}{A_K} = \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{\Lambda}}^2}{2A_K}. \tag{41}$$

On the other hand, for $\mathbf{u} = \mathbf{x}^\star$ and $\mathbf{v} = \mathbf{y}^\star = \mathbf{A}\mathbf{x}^\star$, $\mathrm{Gap}_{\mathcal{L}}^{\mathbf{u},\mathbf{v}}(\widetilde{\mathbf{x}}_K, \widetilde{\mathbf{y}}_K) \geq 0$, and, recalling from Eq. (6), Eq. (8), and Eq. (9) that $\mathbf{y}_K = \mathbf{A}\widetilde{\mathbf{x}}_K$, we can also conclude from Eq. (40) that

$$\mathbb{E}\Big[\frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}_K - \mathbf{x}^\star)\|_2^2\Big] \leq \frac{\phi_0(\mathbf{x}^\star)}{A_K} = \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{\Lambda}}^2}{2A_K}. \tag{42}$$

By Proposition 2.1 (b), $f(\mathbf{x}^\star) = -\frac{1}{2}\mathbf{1}^\top\mathbf{x}^\star = -\frac{1}{2}\|\mathbf{A}\mathbf{x}^\star\|_2^2$. Using this identity, one can verify that, $\forall \mathbf{x}$,

$$\begin{aligned}f(\mathbf{x}) - f(\mathbf{x}^\star) + \frac{1}{2}\|\mathbf{A}(\mathbf{x}-\mathbf{x}^\star)\|_2^2 &= \langle \mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{1}, \mathbf{x} - \mathbf{x}^\star\rangle \\ &= \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}^\star\rangle.\end{aligned}$$

Hence, summing Eq. (41) and Eq. (42), we also have

$$\mathbb{E}\Big[\langle \nabla f(\widetilde{\mathbf{x}}_K), \widetilde{\mathbf{x}}_K - \mathbf{x}^\star\rangle + \frac{1}{2}\|\mathbf{A}(\widetilde{\mathbf{x}}_K - \mathbf{x}^\star)\|^2\Big] \leq \frac{2\phi_0(\mathbf{x}^\star)}{A_K} = \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{\Lambda}}^2}{A_K}.$$

Finally, the bound on the rate of growth of A_k is provided in Appendix 6.3. □

6.3 Omitted Proofs from Section 3: Growth of Scalar Sequences

In this section, we use the properties of $\{a_i\}$ and $\{a_i^k\}$ to obtain our claimed rate of growth of A_k. Note that in any iteration $k \geq 2$ of Algorithm 1, there are two possible updates to a_k, which we name as follows.

$$\text{Type I update: } a_{k+1} = \frac{na_k}{n-1} \tag{43}$$

$$\text{Type II update: } a_{k+1} = \frac{\sqrt{A_k}}{2n} \tag{44}$$

Obtaining a handle on the growth rate of A_k requires controlling the number of updates of both types specified above. At a high level, the idea behind obtaining such a bound is that if the algorithm had only Type II updates, we would have $A_k \geq \Omega(\frac{k^2}{n^2})$; we then go on to show that we cannot have more than $\frac{5}{2}n \log n$ Type I updates since those make a_k grow too fast. We formalize this intuition in the following lemmas.

Lemma 6.4. *In Algorithm 1, we have, for $k \geq 2$, that $a_{k+1} \geq a_k$ and $A_{k+1} > A_k$.*

Proof. Notice that for all k, we have $a_k > 0$, which implies that $A_{k+1} \stackrel{\text{def}}{=} A_k + a_{k+1}$ satisfies $A_{k+1} > A_k$. To check the non-decreasing nature of a_k, we recall that $a_{k+1} = \min\left(\frac{na_k}{n-1}, \frac{\sqrt{A_k}}{2n}\right)$. In the case that $\frac{\sqrt{A_k}}{2n} \geq \frac{na_k}{n-1}$, we have $a_{k+1} = \frac{na_k}{n-1} > a_k$, as claimed. Consider the other case with $a_{k+1} = \frac{\sqrt{A_k}}{2n}$, and suppose, for the sake of contradiction, that $a_{k+1} < a_k$. Chaining this inequality with the assumed expression for a_{k+1}, scaling appropriately, and squaring both sides gives $A_k < 4n^2 a_k^2$. Plugging this into $A_k = A_{k-1} + a_k$ and solving for a_k from this quadratic inequality (and further invoking the nonnegativity of a_k), yields $a_k > \frac{1+\sqrt{1+16n^2 A_{k-1}}}{8n^2} > \frac{\sqrt{A_{k-1}}}{2n}$. However, this contradicts $a_k = \min\left(\frac{na_{k-1}}{n-1}, \frac{\sqrt{A_{k-1}}}{2n}\right) \leq \frac{\sqrt{A_{k-1}}}{2n}$. □

Lemma 6.5. *Consider the iterations $\{s_k\}$ in which Algorithm 1 performs a Type II update $a_{s_k+1} = \frac{\sqrt{A_{s_k}}}{2n}$. Then we have $A_{s_k} \geq \frac{k^2}{c_1 n^2}$ and $a_{s_k} \geq \frac{k-1}{2\sqrt{c_1}n^2}$ for $c_1 = 36$.*

Proof. We prove this claim by induction. First, notice that $s_k \geq k+1$ for any k. Recall our initialization $a_1 = A_1 = \frac{1}{\sqrt{2}n^{1.5}}$. By combining this with the monotonicity property stated in Lemma 6.4, we have $a_{s_1} \geq a_2 = \frac{1}{\sqrt{2}n^{2.5}} \geq 0$. By using Lemma 6.4 again, we have, in a similar fashion, that $A_{s_1} \geq A_2 = \frac{1}{\sqrt{2}n^{2.5}} + \frac{1}{\sqrt{2}n^{1.5}} \geq \frac{1}{c_1 n^2}$, which proves the base case for induction. Assume that for some $k > 1$, we have the induction hypothesis $A_{s_k} \geq \frac{k^2}{c_1 n^2}$ and $a_{s_k} \geq \frac{k-1}{2\sqrt{c_1}n^2}$. Then, combining the monotonicity of A_k from Lemma 6.4 with the fact that the algorithm performs a Type II update on a_{s_k}, we have $a_{s_{k+1}} = \frac{\sqrt{A_{s_{k+1}-1}}}{2n} \geq \frac{\sqrt{A_{s_k}}}{2n} \geq \frac{k}{2\sqrt{c_1}n^2}$. By again applying monotonicity of A_k and the induction hypothesis about a_k, we have $A_{s_{k+1}} = A_{s_{k+1}-1} + a_{s_{k+1}} \geq A_{s_k} + a_{s_{k+1}} \geq \frac{2k^2+\sqrt{c_1}k}{2c_1 n^2} > \frac{(k+1)^2}{c_1 n^2}$. □

Lemma 6.6. *If at some k_0^{th} iteration of Algorithm 1, we have that*

$$a_{k_0} > \frac{n-1}{2\sqrt{c_1}n^2}; \quad A_{k_0} \geq \frac{1}{c_1} \tag{45}$$

then for all $k \geq k_0$, we have that

$$a_k \geq \frac{k-1-k_0+n}{2\sqrt{c_1}n^2}; \quad A_k \geq \frac{(k-k_0+n)^2}{c_1 n^2} \tag{46}$$

for $c_1 = 36$.

Proof. We prove the claim by induction. First, the base case is true for $k = k_0$ by our assumption on a_{k_0} and A_{k_0}. Assume the induction hypothesis $a_k \geq \frac{k-1-k_0+n}{2\sqrt{c_1}n^2}$ and $A_k \geq \frac{(k-k_0+n)^2}{c_1 n^2}$ for $k \geq k_0$. We now discuss how a_k changes with the two types of updates.

If the algorithm performs a Type I update on a_k, then, by definition, $a_{k+1} = \frac{na_k}{n-1}$. Now applying the assumed lower bound on a_k, we have, when $k > k_0$, that

$$a_{k+1} = \frac{na_k}{n-1} \geq \frac{k-1-k_0+n}{2\sqrt{c_1}n(n-1)} \geq \frac{k-k_0+n}{2\sqrt{c_1}n^2}.$$

Similarly, given that $A_k \geq \frac{(k-k_0+n)^2}{c_1 n^2}$, we have,

$$A_{k+1} = A_k + a_{k+1} \geq \frac{(k-k_0+n)^2}{c_1 n^2} + \frac{k-k_0+n}{2\sqrt{c_1}n^2} \geq \frac{(k+1-k_0+n)^2}{c_1 n^2}.$$

If, on the other hand, the algorithm performs a Type II update on a_k, then we have

$$a_{k+1} = \frac{\sqrt{A_k}}{2n} \geq \frac{k-k_0+n}{2\sqrt{c_1}n^2}.$$

This completes the induction. □

As we saw in Lemma 6.6, after the k_0^{th} iteration - starting at which Inequality (45) holds - A_k grows fast. We therefore need to estimate the number of Type I updates *before* the k_0^{th} iteration.

Lemma 6.7. *There are at most $\frac{3}{2}n\log n$ Type I updates (Equation (43)) performed before the k_0^{th} iteration (the first iteration at which Inequality (45) holds).*

Proof. Suppose there are n_1 Type I updates performed by Algorithm 1 before the k_0^{th} iteration, when Inequality (45) starts to hold. Further, by Lemma 6.4, a_k is monotonically increasing (for both types of updates). Then, when considering Type I updates (Equation (43)), we have $a_{k_0} \geq \left(\frac{n}{n-1}\right)^{n_1} a_2 = \left(\frac{n}{n-1}\right)^{n_1} \cdot \frac{1}{\sqrt{2}n^{2.5}}$. In order for $a_{k_0} > \frac{n-1}{12n^2}$, we only need to have $n_1 > \log_{\frac{n}{n-1}}\left(\frac{\sqrt{n}(n-1)}{6\sqrt{2}}\right)$. In a similar fashion, combining the monotonicity of A_k from Lemma 6.4 with the Type I update rule, we have

$$A_{k_0} \geq a_2\left(1 + \frac{n}{n-1} + \left(\frac{n}{n-1}\right)^2 + \cdots + \left(\frac{n}{n-1}\right)^{n_1}\right) > \frac{(\frac{n}{n-1})^{n_1}}{\sqrt{2}n^{2.5}}.$$

So, in order to have $A_{k_0} > \frac{1}{36}$ per Inequality (45), we only need to have $n_1 \geq \log_{\frac{n}{n-1}}(\frac{n^{2.5}}{18\sqrt{2}})$. By using the approximation $1 + x \leq e^x$ and combining the above two bounds, we get as soon as $n_1 \geq \frac{3}{2}n\log n$, the inequality (45) holds. □

Proposition 6.8. *[Rate of change of A_k] When $k \geq \frac{5}{2}n\log n$, we have $A_k \geq \frac{(k-\frac{5}{2}n\log n)^2}{36n^2}$.*

Proof. Let there be t_1 Type I updates and t_2 Type II updates before the first iteration at which Inequality (45) holds, and let us call this iteration k_0. By the result of Lemma 6.7, we have $t_1 \leq \frac{3}{2}n\log n$. By the result of Lemma 6.5, we must have $A_{k_0} \geq \frac{t_2^2}{c_1 n^2}$ and $a_{k_0} \geq \frac{t_2-1}{2\sqrt{c_1}n^2}$. To meet the requirement in Inequality (45) then, we can see that $t_2 \leq n$. Therefore, $k_0 = t_1 + t_2 \leq \frac{3}{2}n\log n + n \leq \frac{5}{2}n\log n$. Having reached the k_0^{th} iteration, the result of Lemma 6.6 applies, and we have $A_k \geq \frac{(k-k_0)^2}{c_1 n^2}$. □

6.4 Omitted Proofs from Section 4: Restart Strategy

To establish local error bounds, we start with the observation that (P) is equivalent to a linear complementarity problem.

Proposition 6.9. *Problem* (P) *is equivalent to the following linear complementarity problem, denoted by* $\mathrm{LCP}(\mathbf{M}, \mathbf{q})$.

$$\mathbf{M}\mathbf{x} + \mathbf{q} \geq \mathbf{0},\ \mathbf{x} \geq \mathbf{0},\ \langle \mathbf{x}, \mathbf{M}\mathbf{x} + \mathbf{q} \rangle = 0, \tag{47}$$

where $\mathbf{\Lambda}^{-1}\mathbf{M} = \mathbf{A}^\top \mathbf{A}$ *and* $\mathbf{q} = -\mathbf{\Lambda}^{-1}\mathbf{1}$.

Proof. Observe first that, as $\mathbf{\Lambda}^{-1}$ is a diagonal matrix with positive elements on the diagonal, the stated linear complementarity problem is equivalent to

$$\nabla f(\mathbf{x}) \geq \mathbf{0},\ \mathbf{x} \geq \mathbf{0},\ \langle \nabla f(\mathbf{x}), \mathbf{x} \rangle = 0. \tag{48}$$

By Proposition 2.1, these conditions hold for any solution of (P). In the opposite direction, suppose that the conditions from Eq. (48) hold for some $\mathbf{x}$. Then applying these conditions for any $\mathbf{u} \geq \mathbf{0}$ gives

$$\langle \nabla f(\mathbf{x}), \mathbf{u} - \mathbf{x} \rangle = \langle \nabla f(\mathbf{x}), \mathbf{u} \rangle \geq 0.$$

But $\langle \nabla f(\mathbf{x}), \mathbf{u} - \mathbf{x} \rangle \geq 0$ is the first-order optimality condition for (P), and so $\mathbf{x}$ solves (P). □

For $r(\mathbf{x}) = \|\mathbf{R}(\mathbf{x})\|_{\mathbf{\Lambda}}$, a quantity termed *natural residual* [30], local error bound is obtained as a corollary of the following theorem.

Theorem 6.10 ([30], Theorem 2.1)**.** *Let* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *be such that* $\mathrm{LCP}(\mathbf{M}, \mathbf{0})$ *has* $\mathbf{0}$ *as its unique solution. Then there exists* $\mu > 0$ *such that for each* $\mathbf{x} \in \mathbb{R}^n$*, we have* $r(\mathbf{x}) \geq \mu \|\mathbf{x} - \mathbf{x}^\star\|$*, where* $\mathbf{x}^\star$ *is a solution to* $\mathrm{LCP}(\mathbf{M}, \mathbf{q})$ *that is closest to* $\mathbf{x}$ *under the norm* $\|\cdot\|$.

Theorem 6.10 applies to our problem due to the nonnegativity (and nondegeneracy) of $\mathbf{A}$ and choosing $\|\cdot\| = \|\cdot\|_{\mathbf{\Lambda}}$. By arguing that Theorem 3.5 provides an upper bound on $r(\widetilde{\mathbf{x}}_K)$, in expectation, we then obtain our final result below.

Proposition 6.11. *For any* $\mathbf{x} \in \mathbb{R}^n_+$, $r(\mathbf{x}) \leq \sqrt{2n(f(\mathbf{x}) - f(\mathbf{x}^\star))}$*, where* $\mathbf{x}^\star \in \mathrm{argmin}_{\mathbf{u} \in \mathbb{R}^n_+} f(\mathbf{u})$.

Proof. Given $\mathbf{x} \in \mathbb{R}^n_+$, consider $\hat{\mathbf{x}}$ defined as $\hat{x}_{j^\star} = x_{j^\star} - \mathbf{R}_{j^\star}(\mathbf{x})$, where $j^\star = \mathrm{argmax}_{1 \leq j \leq n} |\mathbf{R}_j(\mathbf{x})| \cdot \|\mathbf{A}_{:j}\|_2$, and $\hat{x}_j = x_j$ for $j \neq j^\star$. Then observing that

$$\begin{aligned} f(\hat{\mathbf{x}}) - f(\mathbf{x}) &= \nabla_{j^\star} f(\mathbf{x})([\hat{\mathbf{x}}]_{j^\star} - [\mathbf{x}]_{j^\star}) + \frac{\|\mathbf{A}_{:j^\star}\|_2^2}{2} |[\hat{\mathbf{x}}]_{j^\star} - [\mathbf{x}]_{j^\star}|^2 \\ &\leq -\frac{1}{2} |\mathbf{R}_{j^\star}(\mathbf{x})|^2 \|\mathbf{A}_{:j^\star}\|_2^2 \leq -\frac{1}{2n} \|\mathbf{R}(\mathbf{x})\|_{\mathbf{\Lambda}}^2, \end{aligned}$$

and combining with $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}^\star)$, $r(\mathbf{x}) = \|\mathbf{R}(\mathbf{x})\|_{\mathbf{\Lambda}}$, the claimed bound follows after a simple rearrangement. □

Theorem 4.1. *Given an error parameter* $\varepsilon > 0$ *and* $\mathbf{x}_0 = \mathbf{0}$*, consider the following algorithm* $\mathcal{A}$:

> $\mathcal{A}$: ***SI-NNLS+ with Restarts***
> *Initialize:* $k = 1$.
> *Initialize Lazy SI-NNLS+ at* $\mathbf{x}_{k-1}$.
> *Run Lazy SI-NNLS+ until the output* $\widetilde{\mathbf{x}}_K^k$ *satisfies* $r(\widetilde{\mathbf{x}}_K^k) \leq \frac{1}{2} r(\mathbf{x}_{k-1})$.
> *Restart Lazy SI-NNLS+ initializing at* $\mathbf{x}_k = \widetilde{\mathbf{x}}_K^k$.
> *Increment* k.
> *Repeat until* $r(\widetilde{\mathbf{x}}_K^k) \leq \varepsilon$.

Then, the expected number of arithmetic operations of $\mathcal{A}$ *is* $O\big(\mathrm{nnz}(\mathbf{A})\big(\log n + \frac{\sqrt{n}}{\mu}\big) \log\big(\frac{r(\mathbf{x}_0)}{\varepsilon}\big)\big)$. *As a consequence, given* $\bar{\varepsilon} > 0$*, the total expected number of arithmetic operations until a point with* $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \bar{\varepsilon} |f(\mathbf{x}^\star)|$ *can be constructed by* $\mathcal{A}$ *is* $O\Big(\mathrm{nnz}(\mathbf{A})\Big(\log n + \frac{\sqrt{n}}{\mu}\Big) \log\Big(\frac{n}{\mu \bar{\varepsilon}}\Big)\Big)$.

Proof. Because each restart halves the natural residual $r(\mathbf{x})$, it is immediate that the total number of restarts until $r(\widetilde{\mathbf{x}}_K^k) \leq \varepsilon$ is bounded by $\log(\frac{r(\mathbf{x}_0)}{\varepsilon})$. Thus, to prove the first (and main) part of the theorem, we only need to bound the number of iterations (and the overall number of arithmetic operations) of (Lazy) SI-NNLS+ in expectation. Hence, in the following, we only consider one run

of SI-NNLS+ until the natural residual is halved. To keep the notation simple, we let $\mathbf{x}_0$ denote the initial point of SI-NNLS+ and $\widetilde{\mathbf{x}}_k$ denote the output of SI-NNLS+ at iteration k. If $r(\mathbf{x}_0) = 0$, $\mathcal{A}$ halts immediately and the bound on the number of iterations holds trivially, so assume $r(\mathbf{x}_0) > 0$. Using Theorem 3.5, we have that $\forall k \geq 2$,

$$\mathbb{E}[A_k r^2(\widetilde{\mathbf{x}}_k)] \leq n\|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{\Lambda}}^2 \leq \frac{n}{\mu^2} r^2(\mathbf{x}_0). \tag{49}$$

As $r^2(\cdot)$ is nonnegative, we can use Markov's inequality to bound the total number of iterations K until $r(\widetilde{\mathbf{x}}_K) \leq \frac{r(\mathbf{x}_0)}{2}$. In particular, using Eq. (49), we get by Markov's inequality that $\Pr[K > k] \leq \Pr[r^2(\widetilde{\mathbf{x}}_k) \geq \frac{r^2(\mathbf{x}_0)}{4}] \leq \frac{4n}{\mu^2 A_k}$. As K is nonnegative, we can estimate its expectation using

$$\begin{aligned}
\mathbb{E}[K] &= \sum_{i=0}^{\infty} \Pr[K > i] \leq \sum_{i=0}^{\infty} \min\left\{1, \frac{4n}{\mu^2 A_i}\right\} \\
&\leq \sum_{i=0}^{\lceil 12n\sqrt{n}/\mu + \frac{5}{2}n\log n\rceil} 1 + \sum_{\lceil 12n\sqrt{n}/\mu + \frac{5}{2}n\log n\rceil + 1}^{\infty} \frac{4n}{\mu^2 A_i} \\
&\leq 24n\sqrt{n}/\mu + \frac{5}{2}n\log n + 2,
\end{aligned}$$

where in the last inequality we use the rate of A_k from Proposition 6.8.

In the lazy implementation of SI-NNLS+, as argued in Appendix 7, the expected cost of an iteration is $\frac{\mathrm{nnz}(\mathbf{A})}{n}$, which leads to the claimed bound on the number of arithmetic operations until $r(\mathbf{x}) \leq \varepsilon$.

By using that $r(\mathbf{x}_0) \leq \sqrt{2n(f(\mathbf{x}_0) - f(\mathbf{x}^\star))} = \sqrt{2n|f(\mathbf{x}^\star)|}$, $f\left(\widetilde{\mathbf{x}}^{K_1} - \mathbf{R}(\widetilde{\mathbf{x}}^{K_1})\right) - f(\mathbf{x}^\star) \leq \left((n-1) + \frac{n+1}{\mu}\right) r^2(\widetilde{\mathbf{x}}^{K_1})$ (argued below), the bound on the number of iterations until $f\left(\widetilde{\mathbf{x}}^{K_1} - \mathbf{R}(\widetilde{\mathbf{x}}^{K_1})\right) - f(\mathbf{x}^\star) \leq \bar{\varepsilon}|f(\mathbf{x}^\star)|$ have

$$\begin{aligned}
f\left(\widetilde{\mathbf{x}}^{K_1} - \mathbf{R}(\widetilde{\mathbf{x}}^{K_1})\right) - f(\mathbf{x}^\star) &\leq \left((n-1) + \frac{n+1}{\mu}\right) r^2(\widetilde{\mathbf{x}}^{K_1}) \\
&\leq \left((n-1) + \frac{n+1}{\mu}\right) \frac{1}{2^{2K_1}} r^2(\mathbf{x}_0) \\
&\leq \left((n-1) + \frac{n+1}{\mu}\right) \frac{1}{2^{2K_1}} 2n|f(\mathbf{x}^\star)|
\end{aligned}$$

and by setting $K_1 = \frac{1}{2}\log_2 \frac{2n\left((n-1) + \frac{n+1}{\mu}\right)}{\bar{\varepsilon}}$, we have this bound.

Finally, it remains to argue that $f\left(\widetilde{\mathbf{x}}^{K_1} - \mathbf{R}(\widetilde{\mathbf{x}}^{K_1})\right) - f(\mathbf{x}^\star) \leq \left((n-1) + \frac{n+1}{\mu}\right) r^2(\widetilde{\mathbf{x}}^{K_1})$. Observe that the definition of $\mathbf{R}(\mathbf{x})$ is equivalent to $\mathbf{x} - \bar{\mathbf{x}}$, where

$$\bar{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{u} \in \mathbb{R}_+^n} \left\{ \langle \nabla f(\mathbf{x}), \mathbf{u} - \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|_{\mathbf{\Lambda}}^2 \right\}.$$

By the first-order optimality of $\bar{\mathbf{x}}$ based on the equivalent definition of $\mathbf{R}(\mathbf{x})$ above, we have $\langle \nabla f(\mathbf{x}) + \mathbf{\Lambda}(\bar{\mathbf{x}} - \mathbf{x}), \mathbf{x}^\star - \bar{\mathbf{x}} \rangle \geq 0$. Rearranging, and using the definition of convexity of f, we have

$$\begin{aligned}
f(\bar{\mathbf{x}}) - f(\mathbf{x}^\star) &\leq \langle \nabla f(\bar{\mathbf{x}}), \bar{\mathbf{x}} - \mathbf{x}^\star \rangle \\
&\leq \langle \nabla f(\mathbf{x}) - \nabla f(\bar{\mathbf{x}}) + \mathbf{\Lambda}(\bar{\mathbf{x}} - \mathbf{x}), \mathbf{x}^\star - \bar{\mathbf{x}} \rangle \\
&= \langle (\mathbf{A}^\top \mathbf{A} - \mathbf{\Lambda})(\mathbf{x} - \bar{\mathbf{x}}), \mathbf{x}^\star - \bar{\mathbf{x}} \rangle \\
&= \langle (\mathbf{A}^\top \mathbf{A} - \mathbf{\Lambda})\mathbf{R}(\mathbf{x}), \mathbf{R}(\mathbf{x}) \rangle + \langle (\mathbf{A}^\top \mathbf{A} - \mathbf{\Lambda})\mathbf{R}(\mathbf{x}), \mathbf{x}^\star - \mathbf{x} \rangle \\
&= \langle (\mathbf{A}^\top \mathbf{A} - \mathbf{\Lambda})\mathbf{R}(\mathbf{x}), \mathbf{R}(\mathbf{x}) \rangle + \langle \mathbf{A}^\top \mathbf{A}\mathbf{R}(\mathbf{x}), \mathbf{x}^\star - \mathbf{x} \rangle - \langle \mathbf{\Lambda}\mathbf{R}(\mathbf{x}), \mathbf{x}^\star - \mathbf{x} \rangle \\
&\leq (n-1)\|\mathbf{R}(\mathbf{x})\|_{\mathbf{\Lambda}}^2 + (n+1)\|\mathbf{R}(\mathbf{x})\|_{\mathbf{\Lambda}}\|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{\Lambda}} \\
&\leq \left((n-1) + \frac{n+1}{\mu}\right) r^2(\mathbf{x}),
\end{aligned}$$

where in the last inequality we have used the error bound from Theorem 6.10. □

7 Implementation Version of SI-NNLS+

Since Algorithm 1 explicitly updates $\widetilde{\mathbf{x}}_k$ and $\widetilde{\mathbf{y}}_k$ (of lengths n and m respectively), the per iteration cost is $O(m+n)$, which is unnecessarily high when the matrix $\mathbf{A}$ is sparse. In this section, we show that by using a *lazy* update strategy, we can efficiently implement Algorithm 1 with overall complexity independent of the ambient dimension. To attain this result, we maintain implicit representations for $\widetilde{\mathbf{x}}_k$, $\mathbf{y}_k$, and $\bar{\mathbf{y}}_k$ by introducing two auxiliary variables that are amenable to efficient updates.

Efficiently Updating the Primal Variable. In Lemma 7.1, we show that we can work with an implicit representation of $\widetilde{\mathbf{x}}_k$ by introducing $\mathbf{r}_k$.

Lemma 7.1. *For $\{\widetilde{\mathbf{x}}_k\}$ defined in Eq. (8) (and simplified in Algorithm 1), we have, for $k \geq 1$,*

$$\widetilde{\mathbf{x}}_k = \mathbf{x}_k + \frac{1}{A_k}\mathbf{r}_k, \tag{50}$$

where $\mathbf{x}_k$ evolves as per Algorithm 1, $\mathbf{r}_1 = \mathbf{0}$ and, when $k \geq 1$, $\mathbf{r}_k = \mathbf{r}_{k-1} + ((n-1)a_k - A_{k-1})(\mathbf{x}_k - \mathbf{x}_{k-1})$.

Proof. We prove the lemma by induction. Using the facts that $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{x}_1 = \widetilde{\mathbf{x}}_1$, $\mathbf{s}_1 = \mathbf{0}$, $a_1 = A_1$, and $A_0 = 0$, we have

$$\begin{aligned}\widetilde{\mathbf{x}}_1 &= \frac{1}{A_1}\left(A_0\widetilde{\mathbf{x}}_0 + a_1\big(n\mathbf{x}_1 - (n-1)\mathbf{x}_0\big)\right)\\ &= \frac{1}{A_1}\left(a_1\mathbf{x}_1 + (n-1)a_1(\mathbf{x}_1 - \mathbf{x}_0)\right)\\ &= \mathbf{x}_1 + ((n-1)a_1 - A_0)(\mathbf{x}_1 - \mathbf{x}_0).\end{aligned} \tag{51}$$

Assume for certain $k \geq 2$, that Eq. (50) holds for $k-1$. Then, using the recursion of $\tilde{\mathbf{x}}_k$ in Algorithm 1, we have that for $k \geq 3$,

$$\begin{aligned}A_k\widetilde{\mathbf{x}}_k &= A_{k-1}\widetilde{\mathbf{x}}_{k-1} + a_k\mathbf{x}_k + (n-1)a_k(\mathbf{x}_k - \mathbf{x}_{k-1})\\ &= A_{k-1}\mathbf{x}_{k-1} + \mathbf{r}_{k-1} + a_k\mathbf{x}_k + (n-1)a_k(\mathbf{x}_k - \mathbf{x}_{k-1})\\ &= A_{k-1}(\mathbf{x}_{k-1} - \mathbf{x}_k + \mathbf{x}_k) + \mathbf{r}_{k-1} + a_k\mathbf{x}_k + (n-1)a_k(\mathbf{x}_k - \mathbf{x}_{k-1})\\ &= A_{k-1}(\mathbf{x}_{k-1} - \mathbf{x}_k) + A_{k-1}\mathbf{x}_k + \mathbf{r}_{k-1} + a_k\mathbf{x}_k + (n-1)a_k(\mathbf{x}_k - \mathbf{x}_{k-1})\\ &= A_k\mathbf{x}_k + \mathbf{r}_{k-1} + ((n-1)a_k - A_{k-1})(\mathbf{x}_k - \mathbf{x}_{k-1})\\ &= A_k\mathbf{x}_k + \mathbf{r}_k,\end{aligned}$$

as required. □

The expression for $\mathbf{r}_k$ in Lemma 7.1 shows that it can be updated at cost $O(1)$ as $\mathbf{x}_k$ differs from $\mathbf{x}_{k-1}$ only at one coordinate. Therefore, by Eq. (50) we need not compute $\tilde{\mathbf{x}}_k$ in all iterations and can instead maintain $\mathbf{r}_k$. Along the same lines, we give an efficient implementation strategy for $\mathbf{y}_k$ and $\bar{\mathbf{y}}_k$ in the following discussion.

Efficiently Updating the Dual Variable. We now show how to update the dual variable efficiently.

Lemma 7.2. *Consider $\{\mathbf{y}_k\}$ and $\{\mathbf{x}_k\}$ evolving as per Algorithm 1. Then, for $k = 1$, we have $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1$; for $k \geq 2$, we have*

$$\mathbf{y}_k = \frac{A_{k-1}}{A_k}\mathbf{y}_{k-1} + \frac{a_k}{A_k}\mathbf{A}\mathbf{x}_k + \frac{(n-1)a_k}{A_k}\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}), \tag{52}$$

$$\tag{53}$$

Proof. The proof is directly from the definition of $\mathbf{y}_k$ in Algorithm 1. □

Lemma 7.3. *Consider $\{\mathbf{y}_k\}$ and $\{\mathbf{x}_k\}$ evolving as per Algorithm 1. Then for all $k \geq 1$, we have*

$$\mathbf{y}_k = \mathbf{A}\mathbf{x}_k + \frac{1}{A_k}\mathbf{s}_k, \tag{54}$$

where $\mathbf{s}_1 = 0$ and $\mathbf{s}_k = \mathbf{s}_{k-1} + ((n-1)a_k - A_{k-1})\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1})$ when $k \geq 2$.

Proof. We prove the lemma by induction. For the base case of $k = 1$, we have, by the choice of $\mathbf{s}_1 = \mathbf{0}$, that $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 = \mathbf{A}\mathbf{x}_1 + \frac{1}{A_1}\mathbf{s}_1$. Then for some $k \geq 2$, assume Eq. (54) holds for $k-1$, then we have,

$$\begin{aligned}
A_k\mathbf{y}_k &= A_{k-1}\mathbf{y}_{k-1} + a_k\mathbf{A}\mathbf{x}_k + (n-1)a_k\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) \\
&= A_{k-1}\mathbf{A}\mathbf{x}_{k-1} + \mathbf{s}_{k-1} + a_k\mathbf{A}\mathbf{x}_k + (n-1)a_k\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) \\
&= A_{k-1}\mathbf{A}(\mathbf{x}_{k-1} - \mathbf{x}_k + \mathbf{x}_k) + \mathbf{s}_{k-1} + a_k\mathbf{A}\mathbf{x}_k + (n-1)a_k\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) \\
&= A_k\mathbf{A}\mathbf{x}_k + \mathbf{s}_{k-1} + ((n-1)a_k - A_{k-1})\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) \\
&= A_k\mathbf{A}\mathbf{x}_k + \mathbf{s}_k, \qquad (55)
\end{aligned}$$

where the first step is by Lemma 7.2, second step is by the induction hypothesis, third step is by adding and subtracting $A_{k-1}\mathbf{A}\mathbf{x}_k$, fourth step is by rearranging terms appropriately, and the final step uses the recursive definition of $\mathbf{s}_k$ stated in the lemma. Dividing throughout by A_k then finishes the proof. □

Algorithm 2 SI-NNLS+ (Implementation)

Input: Matrix $\mathbf{A} \in \mathbb{R}_+^{m\times n}$ with $n \geq 4$, accuracy ε
Output: Vector $\widetilde{\mathbf{x}}_K \in \mathbb{R}_+^n$ such that $f(\widetilde{\mathbf{x}}_K) \leq (1-\varepsilon)f(\mathbf{x}^\star)$.
Initialize: $a_1 = \frac{1}{n-1}$, $a_2 = \frac{n}{n-1}$, $A_1 = a_1$, $\phi_0(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{A}}^2$, $\overline{\mathbf{y}}_0 = \mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$, $\mathbf{p}_0 = \mathbf{0}, \mathbf{q}_0 = \mathbf{A}\mathbf{x}_0, \mathbf{t}_0 = \mathbf{0}, \mathbf{s}_1 = \mathbf{0}, \mathbf{r}_1 = \mathbf{0}$.
for $k = 1, 2, \ldots, K$ **do**
  Sample j_k uniformly at random from $\{1, 2, \ldots, n\}$
  if $k = 1$ **then**
    $\bar{\mathbf{y}}_0 = \mathbf{q}_0$
  else if $k = 2$ **then**
    $\bar{\mathbf{y}}_1 = \mathbf{q}_1 + \frac{a_1}{a_2}\mathbf{t}_1$
  else if $k \geq 3$ **then**
    $\bar{\mathbf{y}}_{k-1} = \mathbf{q}_{k-1} + \frac{1}{A_{k-1}}\Big(1 - \frac{a_{k-1}^2}{a_k A_{k-2}}\Big)\mathbf{s}_{k-1} + \frac{(n-1)a_{k-1}^2}{a_k A_{k-2}}\mathbf{t}_{k-1}$
  end if
  $p_{k,i} = \begin{cases} p_{k-1,i}, & i \neq j_k \\ p_{k-1,i} + na_k\big(\mathbf{A}_{:i}^T\bar{\mathbf{y}}_{k-1} - 1\big), & i = j_k. \end{cases}$
  $x_{k,i} = \begin{cases} x_{k-1,i}, & i \neq j_k \\ \max\big\{0, \min\big\{x_{0,i} - \frac{1}{\|\mathbf{A}_{:i}\|^2}\cdot p_{k,i}, \frac{1}{\|\mathbf{A}_{:i}\|^2}\big\}\big\}, & i = j_k \end{cases}$
  $\mathbf{t}_k = \mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1})$
  if $k \geq 2$ **then**
    $\mathbf{r}_k = \mathbf{r}_{k-1} + ((n-1)a_k - A_{k-1})(\mathbf{x}_k - \mathbf{x}_{k-1})$
    $\mathbf{s}_k = \mathbf{s}_{k-1} + ((n-1)a_k - A_{k-1})\mathbf{t}_k$
  end if
  $\mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{t}_k$
  $A_{k+1} = A_k + a_{k+1}$
  $a_{k+2} = \min\{\frac{na_{k+1}}{n-1}, \frac{\sqrt{A_{k+1}}}{2n}\}$
end for
return $\mathbf{x}_K + \frac{1}{A_K}\mathbf{r}_K$

Lemma 7.4. *Consider* $\{\mathbf{x}_k\}$, $\{\mathbf{y}_k\}$, *and* $\{\overline{\mathbf{y}}_k\}$ *evolving as per Algorithm 1. Then we have that*

$$\bar{\mathbf{y}}_1 = \mathbf{A}\mathbf{x}_1 + \frac{a_1}{a_2}\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_0). \qquad (56)$$

and

$$\bar{\mathbf{y}}_k = \mathbf{A}\mathbf{x}_k + \frac{1}{A_k}\Big(1 - \frac{a_k^2}{a_{k+1}A_{k-1}}\Big)\mathbf{s}_k + \frac{(n-1)a_k^2}{a_{k+1}A_{k-1}}\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}). \qquad (57)$$

Proof. From the definition of $\bar{\mathbf{y}}_k$, the initializations for $\mathbf{x}_0, \mathbf{y}_0$, and $\overline{\mathbf{y}}_0$, and Lemma 7.2, we have

$$\bar{\mathbf{y}}_1 = \mathbf{y}_1 + \frac{a_1}{a_2}(\mathbf{y}_1 - \mathbf{y}_0) = \mathbf{A}\mathbf{x}_1 + \frac{a_1}{a_2}\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_0).$$

For $k \geq 2$, by Lemma 7.2, we have

$$A_k \mathbf{y}_k - A_{k-1}\mathbf{y}_{k-1} = a_k \mathbf{A}\mathbf{x}_k + (n-1)a_k \mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}).$$

As a result,

$$A_{k-1}(\mathbf{y}_k - \mathbf{y}_{k-1}) = a_k \mathbf{A}\mathbf{x}_k + (n-1)a_k \mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) - a_k \mathbf{y}_k. \tag{58}$$

So for $k \geq 2$, it follows that

$$\begin{aligned}
\bar{\mathbf{y}}_k &= \mathbf{y}_k + \frac{a_k}{a_{k+1}}(\mathbf{y}_k - \mathbf{y}_{k-1}) \\
&= \mathbf{y}_k + \frac{a_k}{a_{k+1}}\Big(\frac{a_k}{A_{k-1}}\mathbf{A}\mathbf{x}_k + \frac{(n-1)a_k}{A_{k-1}}\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) - \frac{a_k}{A_{k-1}}\mathbf{y}_k\Big) \\
&= \Big(1 - \frac{a_k^2}{a_{k+1}A_{k-1}}\Big)\mathbf{y}_k + \frac{a_k^2}{a_{k+1}A_{k-1}}\mathbf{A}\mathbf{x}_k + \frac{(n-1)a_k^2}{a_{k+1}A_{k-1}}\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) \\
&= \mathbf{A}\mathbf{x}_k + \frac{1}{A_k}\Big(1 - \frac{a_k^2}{a_{k+1}A_{k-1}}\Big)\mathbf{s}_k + \frac{(n-1)a_k^2}{a_{k+1}A_{k-1}}\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}),
\end{aligned}$$

where the first step is by the definition of $\overline{\mathbf{y}}_k$ in Algorithm 1, the second step is by Eq. (58), the third step is by rearranging, and the final step is by Lemma 7.3. □

Based on the above lemmas, we give our efficient lazy implementation version of Algorithm 1 in Algorithm 2. In Algorithm 2, we also introduce other auxiliary variables $\mathbf{p}_k, \mathbf{q}_k$ and $\mathbf{t}_k$. Based on Lemmas 7.1-7.4, it is easy to verify the equivalence between Algorithms 1 and 2. With this implementation, by updating only the dual coordinates corresponding to the nonzero coordinates of the selected column of $\mathbf{A}$, the per-iteration cost is proportional to the number of nonzero elements of the selected row in the iteration. As a result, the overall complexity result will depend only on the number of nonzero elements of $\mathbf{A}$.