Journal of Machine Learning Research 25 (2024) 1-46 Submitted 7/21; Revised 1/24; Published 2/24

Convergence for nonconvex ADMM, with applications to CT imaging

Rina Foygel Barber RINA@UCHICAGO.EDU
Department of Statistics
University of Chicago
Chicago, IL 60637, USA

Emil Y. Sidky SIDKY@UCHICAGO.EDU
Department of Radiology
University of Chicago
Chicago, IL 60637, USA

Editor: Genevera Allen

Abstract

The alternating direction method of multipliers (ADMM) algorithm is a powerful and flexible tool for complex optimization problems of the form $\min\{f(x) + g(y) : Ax + By = c\}$. ADMM exhibits robust empirical performance across a range of challenging settings including nonsmoothness and nonconvexity of the objective functions f and g, and provides a simple and natural approach to the inverse problem of image reconstruction for computed tomography (CT) imaging. From the theoretical point of view, existing results for convergence in the nonconvex setting generally assume smoothness in at least one of the component functions in the objective. In this work, our new theoretical results provide convergence guarantees under a restricted strong convexity assumption without requiring smoothness or differentiability, while still allowing differentiable terms to be treated approximately if needed. We validate these theoretical results empirically, with a simulated example where both f and g are nondifferentiable—and thus outside the scope of existing theory—as well as a simulated CT image reconstruction problem.

Keywords: CT imaging, ADMM, nonconvex optimization

1. Introduction

In this work, we consider optimization problems of the form

$$\text{Minimize } f(x) + g(y) \text{ subject to the constraint that } Ax + By = c. \tag{1}$$

Problems of this form arise in many applications throughout the physical and biological sciences. In particular, we are interested in optimization problems pertaining to computed tomography (CT) imaging, which, as we will see later on, can often be expressed in this type of formulation.

Solving the optimization problem (1) can be computationally challenging even when the functions f and g are both convex. Challenges in the convex setting may include high dimensionality of the variables x and y, nondifferentiability of f and/or g, or poor conditioning of the linear transformations A, B or the functions f, g. If one or both functions are nonconvex, this brings an additional level of difficulty to the optimization problem.

In this work, we study a linearized form of the alternating directions method of multipliers (ADMM) algorithm, in the setting where f and g may both be nonconvex and nonsmooth. While variants of this algorithm are very well known in the literature, existing theoretical results have typically been restricted to narrower settings (e.g., assuming that at least one of the two functions f, g must be smooth), and thus cannot be applied to guarantee convergence for many settings arising in modern high dimensional optimization and data analysis.

Outline In Section 2, we describe the method of nonconvex ADMM with linear approximations, and review known results in the literature on the convergence properties of this type of algorithm in various settings. In Section 3 we present our new convergence result, which addresses a more flexible setting allowing both f and g to be potentially nonconvex and nonsmooth. We demonstrate the performance of the algorithm on a simple simulated quantile regression problem in Section 4, and present an application to computed tomography (CT) imaging in Section 5. Finally, some future directions and implications of this work are discussed in Section 6. Some proofs and additional technical details are deferred to the Appendix.

2. Setting and background

Consider the optimization problem

$$\text{Minimize } f(x) + g(y) \ : \ x \in \mathbb{R}^d, y \in \mathbb{R}^m \text{ such that } Ax + By = c \tag{2}$$

where the functions f on $\mathbb{R}^d$ and g on $\mathbb{R}^m$ are potentially nonconvex and/or nondifferentiable, while $A \in \mathbb{R}^{k \times d}$, $B \in \mathbb{R}^{k \times m}$, and $c \in \mathbb{R}^k$ define linear constraints on the variables. In this work, we will consider functions f and g that can be decomposed as

$$f(x) = f_c(x) + f_d(x), \quad g(y) = g_c(y) + g_d(y)$$

where f_c is convex (possibly nondifferentiable) and f_d is twice differentiable (possibly nonconvex), and similarly for g_c and g_d. This decomposition allows us to take linear approximations to the differentiable terms f_d and g_d, where needed, to ensure simple calculations for each update step of our iterative algorithm.

We will assume that f_c and g_c are *proper functions*. Formally, this means that we can write

$$f_c : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\},$$

with nonempty domain $\text{dom}(f_c) := \{x \in \mathbb{R}^d : f(x) < +\infty\}$ (and similarly for g_c). We also assume that f_c and g_c are lower semi-continuous. The differentiable component f_d is assumed to be defined on all of $\mathbb{R}^d$, i.e.,

$$f_d : \mathbb{R}^d \to \mathbb{R},$$

and similarly for g_d on $\mathbb{R}^m$. Putting these assumptions together, we see that f and g are also proper functions, with domains $\text{dom}(f) = \text{dom}(f_c)$ and $\text{dom}(g) = \text{dom}(g_c)$ (note that convexity of f_c, g_c ensures that these domains are also convex). Finally, we assume that the feasible set

$$\big(\text{dom}(f) \times \text{dom}(g)\big) \cap \Big\{(x, y) \in \mathbb{R}^d \times \mathbb{R}^m : Ax + By = c\Big\}$$

is nonempty. We will say that a point (x, y) is *feasible* for this optimization problem if it lies in this feasible set, i.e., $x \in \mathrm{dom}(f)$, $y \in \mathrm{dom}(g)$, and the constraint $Ax + By = c$ is satisfied.

2.1 Background and prior work

2.1.1 ADMM for convex optimization problems

The alternating directions method of multipliers (ADMM) algorithm is a method for solving problems of the form (2). It was developed initially for the setting where f and g are both convex, and operates by reformulating the optimization problem (2) with an augmented Lagrangian,

$$\min_{x,y} \max_{u} \{\mathcal{L}_\Sigma(x, y, u)\},$$

where the augmented Lagrangian is defined as

$$\mathcal{L}_\Sigma(x, y, u) = f(x) + g(y) + \langle u, Ax + By - c\rangle + \frac{1}{2}\|Ax + By - c\|_\Sigma^2, \tag{3}$$

for some positive definite penalty matrix $\Sigma \succ 0$. (Most commonly, Σ is taken to be a multiple of the identity.) See Boyd et al. (2011) for a review of the motivation and performance of ADMM for the convex setting, including the long history of this algorithm and many of its variants.

The ADMM algorithm solves this optimization problem as follows: initializing at some x_0, u_0, y_0, for all $t \geq 0$ we run the steps:

$$\begin{cases} x_{t+1} = \arg\min_x \{\mathcal{L}_\Sigma(x, y_t, u_t)\}, \\ y_{t+1} = \arg\min_y \{\mathcal{L}_\Sigma(x_{t+1}, y, u_t)\}, \\ u_{t+1} = u_t + \Sigma(Ax_{t+1} + By_{t+1} - c). \end{cases} \tag{4}$$

Adding step size matrices In some cases, adding step size matrices $H_f \succeq 0$ for the x update and $H_g \succeq 0$ for the y update can improve the convergence behavior and/or may allow for easier calculation of the update steps:

$$\begin{cases} x_{t+1} = \arg\min_x \left\{\mathcal{L}_\Sigma(x, y_t, u_t) + \frac{1}{2}\|x - x_t\|_{H_f}^2\right\}, \\ y_{t+1} = \arg\min_y \left\{\mathcal{L}_\Sigma(x_{t+1}, y, u_t) + \frac{1}{2}\|y - y_t\|_{H_g}^2\right\}, \\ u_{t+1} = u_t + \Sigma(Ax_{t+1} + By_{t+1} - c). \end{cases} \tag{5}$$

(Here $\succeq$ denotes the positive semidefinite ordering on matrices, i.e., $H_f \succeq 0$ means that H_f is positive semidefinite.)

In many cases, choosing H_f so that $D_f := H_f + A^\top \Sigma A$ is diagonal, or is a multiple of the identity, may be convenient for calculating the x update step—this is because the x update step is a minimization problem of the form $\arg\min_x \left\{f(x) + \frac{1}{2}x^\top D_f x - x^\top v_t\right\}$, where v_t is a vector that depends on the previous iteration. Specifically, this type of choice for H_f can be helpful when the function f separates over the entries of x, $f(x) = \sum_i f_i(x_i)$, so that now the x update step separates completely over the entries of x. Another setting where this type of modification is commonly used is when f is equipped with an inexpensive proximal

map (the map $z \mapsto \arg\min\{f(x) + \frac{1}{2}\|x - z\|_2^2\}$)—for example, the ℓ_1 norm, $f(x) = \|x\|_1$, or the (squared) ℓ_2 norm, $f(x) = \|x\|_2^2$, are both commonly used regularization functions that have simple proximal maps. (Without the matrix H_f, the x update step is of the form $\arg\min_x \left\{f(x) + \frac{1}{2}x^\top A^\top \Sigma A x - x^\top v_t\right\}$, which may be substantially more challenging to compute if $A^\top \Sigma A$ is a dense matrix.) Similarly we may choose H_g with these types of considerations in mind for the y update step. For further details, see Wang and Banerjee (2014, Eqn. (17)), where this type of modification is referred to as a "linearization" of the quadratic penalty term.

This type of modification of ADMM is closely linked to related algorithms for composite optimization problems of the form $f(Ax) + g(x)$, studied via primal-dual methods by, e.g., Chen and Teboulle (1994); Chambolle and Pock (2011); He and Yuan (2012); Valkonen (2014), among many others, and has been applied to convex versions of the CT image reconstruction problem (see, e.g., Nien and Fessler (2014)).

Linear approximations For many optimization problems, even with the modification of a step size matrix as in (5) above, it may still be challenging to compute the x update step if the function f is difficult to minimize (and similarly, the y step with the function g). In particular, if the x update step itself can only be solved with an iterative procedure, this type of "inner loop" will drastically slow down the convergence of ADMM.

An alternative is to replace the function f with an approximation at each step. In particular, consider our earlier decomposition, $f = f_c + f_d$, where f_c is convex while f_d is twice differentiable. Taking a linear approximation to f_d, at the current iteration x_t, we can approximate the function f as

$$f(x) \approx f_c(x) + \big(f_d(x_t) + \langle \nabla f_d(x_t), x - x_t\rangle\big).$$

Although this inexact calculation of the x update may lead to slower convergence in terms of the total number of iterations, this may be outweighed if this approximation allows the cost of each single iteration to be substantially reduced. We can make the analogous modification for the y update step. This type of modification has been commonly used in both the convex and nonconvex settings, particularly in settings where f itself is twice differentiable so we can take $f_d = f$ and $f_c \equiv 0$. For instance, Wang and Banerjee (2014, Eqn. (21)) study this modification for the convex setting, where this type of approach is referred to as "linearization" of the target function; see also the references described below for the nonconvex setting.

For completeness, Algorithm 1 presents this modified form of ADMM (combining both linear approximations to f_d and g_d, and the addition of step size matrices described above). This is the version of the algorithm that we will study in our work.

2.1.2 Nonconvex ADMM

Next we turn to the nonconvex setting, where the functions f and/or g are no longer required to be convex. In many optimization problems, the ADMM algorithm (possibly with the addition of step size matrices H_f, H_g and/or linear approximations to f_d, g_d) has been observed to perform well, converging successfully and avoiding issues such as saddle points or local minima. The convergence properties in a nonconvex setting have been studied extensively. For example, Wang et al. (2014); Magnússon et al. (2015); Hong et al. (2016);

Algorithm 1 ADMM with linear approximations

Input: Functions $f = f_c + f_d$ and $g = g_c + g_d$, with f_c, g_c convex, f_d, g_d twice differentiable; matrices A, B; vector c; penalty matrix $\Sigma \succ 0$; step size matrices $H_f, H_g \succeq 0$.

Initialize: x_0, y_0, u_0.

for $t = 0, 1, 2, \ldots$ **do**

$$\text{Update } x\text{:} \quad x_{t+1} = \arg\min_x \Big\{ f_c(x) + \langle x, \nabla f_d(x_t) + A^\top u_t \rangle + \frac{1}{2}\|Ax + By_t - c\|_\Sigma^2 + \frac{1}{2}\|x - x_t\|_{H_f}^2 \Big\}.$$

$$\text{Update } y\text{:} \quad y_{t+1} = \arg\min_y \Big\{ g_c(y) + \langle y, \nabla g_d(y_t) + B^\top u_t \rangle + \frac{1}{2}\|Ax_{t+1} + By - c\|_\Sigma^2 + \frac{1}{2}\|y - y_t\|_{H_g}^2 \Big\}.$$

$$\text{Update } u\text{:} \quad u_{t+1} = u_t + \Sigma(Ax_{t+1} + By_{t+1} - c).$$

until some convergence criterion is reached.

Guo et al. (2017); Wang et al. (2018, 2019); Themelis et al. (2020) study the performance of ADMM with f and g update steps calculated exactly (in some cases, extending the algorithm to handle more than two variable blocks), while Li and Pong (2015); Lanza et al. (2017); Jiang et al. (2019); Liu et al. (2019) study the algorithm with linear approximations to (parts of) f and/or g. All of these works prove results of one of the two following types:

- Assume that either f or g is differentiable and has a Lipschitz gradient, and establish convergence guarantees;
- Assume that the algorithm converges (or, more weakly, assume only that the dual variable u_t converges), and establish optimality properties of the limit point.

It is important to note that neither type of existing result verifies that convergence is guaranteed in a nonconvex setting where both f and g are nondifferentiable.

A different type of nonconvexity that is studied in the literature is where f and g are both convex, but the constraint on (x, y) is nonconvex (e.g., $y = A(x)$ for a nonlinear operator A); this type of problem is studied by Valkonen (2014); Ochs et al. (2015), among others. Bolte et al. (2018) allow for nonconvexity both in the functions (f and/or g) and in the constraint on (x, y); as with many of the methods above, the results of this paper require that either f or g is differentiable and has a Lipschitz gradient.

2.1.3 The MOCCA algorithm

Our own earlier work on this problem (Barber and Sidky, 2016) proposed the Mirrored Convex/Concave algorithm (MOCCA), which solves problems of the form (1). At a high level, the MOCCA algorithm can be viewed as a version of Algorithm 1 with a key modification: rather than taking a new linear approximation to f_d and g_d at each iteration t (i.e., computing the gradients $\nabla f_d(x_t)$ and $\nabla g_d(y_t)$), the MOCCA algorithm requires an "inner

loop", where we cycle L_t many times through the variable update steps before re-calculating the linear approximations to f_d and g_d.

In (Barber and Sidky, 2016), two versions of the MOCCA algorithm are proposed:

- The "stable" version (Barber and Sidky, 2016, Algorithm 2), where at each iteration t of the outer loop, we run $L_t \gg 1$ many iterations of the inner loop, and require $L_t \to \infty$.

- The "simple" version (Barber and Sidky, 2016, Algorithm 1), with no inner loop (or equivalently, with $L_t = 1$ for each t).

The theoretical guarantee given in (Barber and Sidky, 2016) proves a convergence result for the "stable" version. To our knowledge, this was a unique result in that it ensured convergence without requiring either f or g to have a Lipschitz gradient (in comparison to the literature on ADMM in the nonconvex setting as discussed above), requiring instead a restricted strong convexity type condition (see Section 3.2 below). However, the theoretical result has the drawback of requiring the inner loop, with $L_t \to \infty$. This requirement contradicts the empirical performance of the algorithm: the empirical results in (Barber and Sidky, 2016) actually implemented the "simple" version of MOCCA, with no inner loop, and the algorithm typically showed convergence even though no theoretical justification was known.

The ADMM algorithm studied in the present work, Algorithm 1, is in fact essentially equivalent to the "simple" version of MOCCA (with a few changes in the details; e.g., in MOCCA, the matrix B was required to be the identity). The novelty of the present work, then, is not in the algorithm itself, but rather in the fact that the theoretical guarantees established in this paper apply to the actual algorithm being run in practice (Algorithm 1, or equivalently, the "simple" version of MOCCA), rather than applying only to a more computationally inefficient version of this algorithm (the "stable" version of MOCCA, as in the theoretical results of (Barber and Sidky, 2016)).

2.2 Preview of new results

In the present work, we establish a convergence guarantee for Algorithm 1 in the nonconvex setting, with no "inner loop" needed in the theory, substantially closing the gap between the theoretical results and our empirical observations for this algorithm. As byproducts of this new analysis, we uncover an additional interesting finding that better explains the dependence of performance on step size parameters. Moreover, our new work allows for a more direct connection to CT imaging—we are able to apply our algorithm, exactly as defined and with no modifications, to simulated CT image reconstruction problems, obtaining very clean results. (For real CT data, issues of scanner calibration, non-random noise, etc., require a more careful application of the algorithm, which we address in separate work, but we mention here that this algorithm has been very successful on real CT data, e.g., Rizzo et al. (2022); Schmidt et al. (2023); Rizzo et al. (2023).)

3. Convergence guarantee

We will prove a convergence result under an additional condition requiring approximate convexity of the problem. If the optimization problem were strongly convex, we would expect that our optimization algorithm would converge to the unique minimizer—which, under strong convexity, would be a point satisfying first-order optimality conditions. In more challenging settings, however, strong convexity may not hold and we will need to relax our goal for convergence.

In this work, we will consider a setting where there is a feasible point $(\tilde{x}, \tilde{y})$ that is *approximately* first-order optimal, around which the optimization problem satisfies a *relaxed* version of strong convexity. Since these conditions will only be required to hold approximately, the point $(\tilde{x}, \tilde{y})$ may in general be nonunique; feasible points $(\tilde{x}', \tilde{y}')$ sufficiently close to $(\tilde{x}, \tilde{y})$ might also satisfy the conditions. This is not a contradiction, however, since our theoretical results will only guarantee convergence to within some neighborhood of $(\tilde{x}, \tilde{y})$.

In the remainder of this section, we will define our assumptions more formally and will state the theoretical guarantee, but we first need to review the definition of the subdifferential in this nonconvex setting.

3.1 Subdifferentials of f and g

Since f and g are not necessarily convex, we pause here to define the notation $\partial f(x)$ and $\partial g(y)$, which is a generalization of the usual subdifferential for convex functions. Here, for any $x \in \mathrm{dom}(f)$, we will use the definition

$$\partial f(x) = \left\{ \xi : \lim_{t \to 0} \frac{f(x + tw) - f(x)}{t} \geq \langle \xi, w \rangle \text{ for all } w \in \mathbb{R}^d \right\}$$

and similarly for g. This definition is illustrated in Figure 3.1.

In particular, given the convex-plus-differentiable decomposition $f = f_c + f_d$, we can write

$$\partial f(x) = \{\xi + \nabla f_d(x) : \xi \in \partial f_c(x)\} \quad \text{and} \quad \partial g(y) = \{\zeta + \nabla g_d(y) : \zeta \in \partial g_c(y)\},$$

where $\partial f_c(x)$ and $\partial g_c(y)$ are the usual subdifferentials of the convex functions f_c and g_c, i.e., for all $x \in \mathrm{dom}(f)$ we define

$$\partial f_c(x) = \left\{ \xi : f(x + w) - f(x) \geq \langle \xi, w \rangle \text{ for all } w \in \mathbb{R}^d \right\},$$

and similarly for g_c.

From this point on, for any $x \in \mathrm{dom}(f)$ and any $y \in \mathrm{dom}(g)$, ξ_x always denotes an element of $\partial f(x)$, and ζ_y always denotes an element of $\partial g(y)$.

3.2 Restricted strong convexity

We will assume a restricted strong convexity (RSC) condition, which at a high level is a relaxation of imposing a strong convexity condition on the constrained optimization problem. This type of convexity condition has been extensively studied in the high-dimensional statistics literature. For background, the condition was proposed initially by Negahban

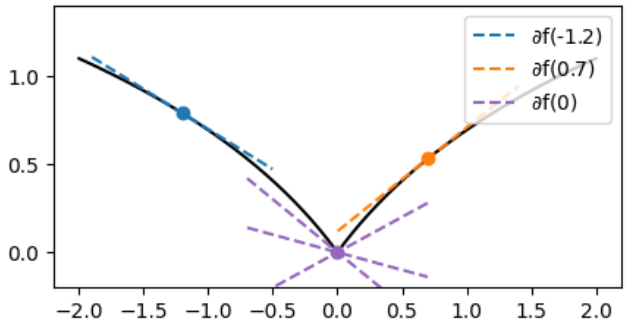

Figure 1: Illustration of the subdifferential $\partial f(t)$, for the function $f(t) = \log(1 + |t|)$. For any $t \neq 0$, the function is differentiable at t, and the subdifferential is a singleton set containing only this derivative, $\partial f(t) = \{f'(t)\} = \{\text{sign}(t)/(1 + |t|)\}$. This is illustrated in the figure for two nonzero values of t. At $t = 0$, the function is nondifferentiable, and the subdifferential is given by $\partial f(0) = [-1, 1]$. This is illustrated in the figure by showing several elements of $\partial f(0)$.

et al. (2012), and was studied by Loh and Wainwright (2015) in the setting of nonconvex loss functions. This type of condition is known to characterize many settings where accurate signal recovery is possible in spite of the "curse of dimensionality", and over recent years has been studied in many settings, e.g., (Jain et al., 2014; Gunasekar et al., 2015; Elenberg et al., 2018).

We will assume the following condition, for some constants $\varepsilon \geq 0$, $\alpha_f, \alpha_g \geq 0$, and $c_f, c_g \in (0, +\infty]$, and some positive definite matrix $\Sigma \succ 0$:

Assumption 1 (Restricted Strong Convexity) *There exists a feasible point $(\tilde{x}, \tilde{y})$ and subgradients $\xi_{\tilde{y}} \in \partial f(\tilde{x}), \zeta_{\tilde{y}} \in \partial g(\tilde{y})$, such that*

$$\left\langle \begin{pmatrix} x - \tilde{x} \\ y - \tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq \alpha_f \min\{\|x - \tilde{x}\|_2^2, c_f \|x - \tilde{x}\|_2\} + \alpha_g \min\{\|y - \tilde{y}\|_2^2, c_g \|y - \tilde{y}\|_2\} - \frac{1}{2}\|Ax + By - c\|_\Sigma^2 - \varepsilon^2, \quad (6)$$

for all $x \in \text{dom}(f)$, $y \in \text{dom}(g)$, $\xi_x \in \partial f(x)$, and $\zeta_y \in \partial g(y)$.

Motivation To motivate this condition, consider a first-order optimal point $(\tilde{x}, \tilde{y})$. We first observe that if the functions f and g were α_f-strongly convex and α_g-strongly convex, respectively, then we would have

$$\left\langle \begin{pmatrix} x - \tilde{x} \\ y - \tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq \alpha_f \|x - \tilde{x}\|_2^2 + \alpha_g \|y - \tilde{y}\|_2^2 \ \forall x \in \text{dom}(f), y \in \text{dom}(g), \forall \xi_x, \zeta_y.$$

If instead f and/or g does not satisfy strong convexity (or may even be nonconvex) but strong convexity is regained once we impose the constraint $Ax + By = c$, we might instead

have a bound of the form

$$\left\langle \begin{pmatrix} x-\tilde{x} \\ y-\tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq \alpha_f \|x-\tilde{x}\|_2^2 + \alpha_g \|y-\tilde{y}\|_2^2 \ \forall \text{ feasible } (x,y),\ \forall \xi_x, \zeta_y.$$

This is strictly weaker than requiring f and g to each be strongly convex; here, the requirement of strong convexity is restricted to the subspace defined by the constraint $Ax+By=c$.

To accommodate the setting of ADMM, where the constraint $Ax+By=c$ is not satisfied exactly at finite iterations, we will need to extend the statement above to allow for points that violate this constraint. This is the motivation for subtracting the term $\frac{1}{2}\|Ax+By-c\|_\Sigma^2$ on the right-hand side of (6), which allows the strong convexity requirement to be relaxed outside of the subspace where the constraint holds. Finally, the additional term ε^2 subtracted on the right-hand side is typically a very small positive constant, allowing for minor violations of the RSC property—we will return to the meaning and interpretation of this term below.

Parameters for the RSC condition We next examine the choices of constants $\alpha_f, \alpha_g, c_f, c_g$, the penalty matrix Σ, and the "tolerance" term ε, in this condition.

- **Constants $\alpha_f, \alpha_g, c_f, c_g$.** As seen earlier, in some cases the objective function may offer strong convexity in feasible directions (i.e., (x,y) such that $Ax+By=c$). In such a case, we would take $c_f = c_g = +\infty$ (and $\varepsilon = 0$). In other settings, however, it may not be possible to guarantee this type of strong curvature, but we can ensure a weaker property by taking finite c_f, c_g. This would arise if, e.g., f is a logistic loss function, which is convex globally but is strongly convex only locally; moreover, in Section 4.2, we will also see this type of weaker convexity guarantee for a sparse quantile regression problem. It may also be the case that the objective function offers strong convexity in the x direction but may not be strongly convex in the y direction (or vice versa), in which case we might have $\alpha_f > 0$ but $\alpha_g = 0$, for example.

- **Penalty matrix Σ.** The matrix Σ appears in both the RSC assumption and in the ADMM algorithm, where it enforces the constraint $Ax+By=c$. In other words, our assumption is that RSC holds with the same matrix Σ as the one used in ADMM. The RSC property therefore provides some insight into the role of the ADMM step size parameter. We can see that, in the presence of nonconvexity—or even if the problem is convex, but not globally strongly convex—the RSC property may fail if the ADMM parameter Σ is chosen to be too small.

  While for specific problems we may have theoretical results that guide our choice of Σ (as for the quantile regression example—see Section 4.2), more generally in practice we may need to tune Σ to achieve good convergence of ADMM. It is common to choose a multiple of the identity, i.e., $\Sigma = \sigma \mathbf{I}_k$, so that we only have a single scalar parameter $\sigma > 0$ to tune. (In the ADMM literature, this parameter is typically denoted by ρ.) In our theory, we allow for a general Σ rather than requiring a multiple of the identity, since in certain settings it may be advantageous to choose a different form for Σ; we will see an example of this in the CT imaging application, in Section 5.1.

- **Tolerance level ε.** Finally we discuss the role of the scalar $\varepsilon \geq 0$. This parameter allows for the condition to hold up to a small tolerance level, and is typically taken

to be vanishing, or even zero. We will see in our theoretical convergence guarantee below, that the RSC property with a nonzero ε only guarantees convergence to within distance $\asymp \varepsilon$ of $(\tilde{x}, \tilde{y})$.

For example, if the optimization problem arises from a statistical question where we would like to estimate some true distribution parameters based on a sample of size n, then often the function f or g reflects an empirical loss that is a random perturbation of some underlying "true" loss function. Allowing for $\varepsilon \asymp n^{-1/2}$ means that the RSC property can hold even if the strong convexity properties of the underlying true loss are not preserved exactly by the empirical loss. The fact that the RSC property only guarantees convergence to within distance ε of the true parameters, is not worrisome in this statistical setting, because convergence beyond the accuracy level $\varepsilon \asymp n^{-1/2}$ is not informative—this is because a sample of size n can only recover parameters up to errors of order $n^{-1/2}$ even with limitless computational resources (see, e.g., Loh and Wainwright (2015, Section 4.1) for further discussion of the role of the ε term in RSC type results for high-dimensional statistics). As an example, the scaling $\varepsilon \asymp n^{-1/2}$ arises in the sparse quantile regression application, for which the RSC property is studied in Section 4.2.

In Appendix A.1, we give some additional intuition and interpretations for the RSC property, for the Σ in particular, showing how RSC relates to the convexity of the augmented Lagrangian $\mathcal{L}_\Sigma$ defined in (3).

3.3 First-order conditions

A first-order stationary point (FOSP) of the optimization problem is a feasible point (x, y) such that, for any feasible (x', y'), it holds that

$$\left\langle \begin{pmatrix} x' - x \\ y' - y \end{pmatrix}, \begin{pmatrix} \xi_x \\ \zeta_y \end{pmatrix} \right\rangle \geq 0 \tag{7}$$

for some $\xi_x \in \partial f(x)$ and some $\zeta_y \in \partial g(y)$. In particular, for any triple $(x, y, u) \in \mathrm{dom}(f) \times \mathrm{dom}(g) \times \mathbb{R}^k$, if it holds that

$$\begin{cases} Ax + By = c, \\ -A^\top u \in \partial f(x), \\ -B^\top u \in \partial g(y), \end{cases} \tag{8}$$

then we can verify that (x, y) is a FOSP (by taking $\xi_x = -A^\top u$ and $\zeta_y = -B^\top u$ in (7)).

To prove (approximate) convergence to the target $(\tilde{x}, \tilde{y})$, we will need to assume that this point is (approximately) first-order optimal.

Assumption 2 *For some $\varepsilon_{\mathrm{FOSP}} \geq 0$, the point $(\tilde{x}, \tilde{y})$ satisfies*

$$\begin{cases} A\tilde{x} + B\tilde{y} = c, \\ \|-A^\top \tilde{u} - \xi_{\tilde{x}}\|_2 \leq \min\left\{\frac{\alpha_f c_f}{2}, \sqrt{\alpha_f} \cdot \varepsilon_{\mathrm{FOSP}}\right\}, \\ \|-B^\top \tilde{u} - \zeta_{\tilde{y}}\|_2 \leq \min\left\{\frac{\alpha_g c_g}{2}, \sqrt{\alpha_g} \cdot \varepsilon_{\mathrm{FOSP}}\right\}, \end{cases} \tag{9}$$

for some $\tilde{u} \in \mathbb{R}^k$, where constants $\alpha_f, \alpha_g, c_f, c_g$ and subgradients $\xi_{\tilde{x}}, \zeta_{\tilde{y}}$ are the same as the ones appearing in Assumption 1.

For intuition, we can see that if $(\tilde{x}, \tilde{y}, \tilde{u})$ were to satisfy the conditions (8) exactly, then this assumption would hold with $\varepsilon_{\text{FOSP}} = 0$.

Analogous to the role of ε in the restricted strong convexity condition, here $\varepsilon_{\text{FOSP}}$ is a tolerance level, allowing the first-order optimality conditions to hold only approximately. We will see that convergence is then guaranteed only up to an accuracy level that scales with these tolerance parameters ε and $\varepsilon_{\text{FOSP}}$.

A key motivation can again be found by considering a statistical setting, where we are minimizing a loss derived from a finite sample of size n (e.g., empirical risk minimization), then we would expect the true parameters $(\tilde{x}, \tilde{y})$ to be approximately first-order optimal with $\varepsilon_{\text{FOSP}} \asymp n^{-1/2}$, reflecting the usual error rates obtained with a sample size n.

3.4 Main result: convergence guarantee

Our main result proves that the ADMM iterates (x_t, y_t, u_t) converge to $(\tilde{x}, \tilde{y}, \tilde{u})$ (up to a tolerance level determined by ε and $\varepsilon_{\text{FOSP}}$), as long as we choose the step size matrices H_f, H_g to satisfy

$$\begin{cases} H_f \succeq 0,\ H_f + A^\top \Sigma A \succ 0, \text{ and } H_f \succeq \nabla^2 f_d(x) \text{ for all } x \in \text{dom}(f), \\ H_g \succeq 0,\ H_g + B^\top \Sigma B \succ 0, \text{ and } H_g \succeq \nabla^2 g_d(y) \text{ for all } y \in \text{dom}(g). \end{cases} \tag{10}$$

We note that, if f_d (respectively g_d) is concave and $A^\top \Sigma A$ (respectively $B^\top \Sigma B$) is full-rank, then the corresponding step size matrix H_f (respectively H_g), can be chosen to be zero. However, even in such a setting, we may prefer to take a nonzero step size matrix for easier update step calculations, as discussed above. We can also observe that the condition $H_f + A^\top \Sigma A \succ 0$, together with the assumption that f_c is convex, proper, and lower semi-continuous, ensures that x_{t+1} is unique and well-defined (i.e., the subproblem for the x update step has a unique minimum), and similarly the condition $H_g + B^\top \Sigma B \succ 0$ ensures the same for the y update step.

Theorem 1 *Suppose that the point $(\tilde{x}, \tilde{y})$ is feasible, satisfies Assumption 1 (restricted strong convexity), and satisfies Assumption 2 (approximate first-order optimality) for some $\tilde{u} \in \mathbb{R}^k$. Suppose that the nonconvex ADMM algorithm given in Algorithm 1 is run with the penalty matrix Σ chosen according to the restricted strong convexity property* (6)*, with step size matrices H_f, H_g satisfying* (10)*, and initialized at an arbitrary point $(x_0, y_0, u_0) \in \text{dom}(f) \times \text{dom}(g) \times \mathbb{R}^k$.*

Define

$$\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t \text{ and } \bar{y}_T = \frac{1}{T}\sum_{t=1}^{T} y_t,$$

where x_t, y_t are the iterates of the nonconvex ADMM algorithm. Then for all $T \geq 1$,

$$\alpha_f \min\left\{\|\bar{x}_T - \tilde{x}\|_2^2, c_f\|\bar{x}_T - \tilde{x}\|_2\right\} + \alpha_g \min\left\{\|\bar{y}_T - \tilde{y}\|_2^2, c_g\|\bar{y}_T - \tilde{y}\|_2\right\} \leq \frac{C(\tilde{x}, \tilde{y}, \tilde{u}; x_0, y_0, u_0)}{T} + 4(\varepsilon^2 + \varepsilon_{\text{FOSP}}^2).$$

The function C appearing in the upper bound is defined explicitly in the proof, and does not depend on the iteration number T.

An important observation is that convergence is guaranteed only up to the error level scaling as $\varepsilon^2 + \varepsilon^2_{\text{FOSP}}$—these terms do not vanish as $T \to \infty$. To understand why this is exactly as expected, we can again consider a statistical setting, where the true parameters $(\tilde{x}, \tilde{y})$ are estimated by minimizing a loss derived from a finite sample of size n; in this type of setting, convergence can only be expected to recover $(\tilde{x}, \tilde{y})$ up to some accuracy level. Indeed, even if we were able to compute the global minimizer of the optimization problem, we would still expect nonzero error in recovering $(\tilde{x}, \tilde{y})$. In particular, as described above, in such settings we expect the RSC property and the approximate first-order optimality property to hold with $\varepsilon, \varepsilon_{\text{FOSP}} \asymp n^{-1/2}$; this then implies that, for sufficiently large T, we have $\|\bar{x}_T - \tilde{x}\|_2 \lesssim n^{-1/2}$. As discussed earlier, since this is the expected rate for parameter estimation based on a sample of size n (in particular, even the *global* minimizer of the optimization problem will have this same error rate), we cannot hope for a better guarantee.

Comparison to related work In Section 2.1.2, we discussed prior work on different variants of the nonconvex ADMM algorithm (with or without linear approximations to the differentiable components f_d and g_d of the objective function). These existing results all require that at least one of the two functions (f or g) must be smooth, or alternatively proves a weaker convergence result, establishing properties of the limit point under the assumption that the algorithm converges (without proving that convergence must occur). The related MOCCA algorithm, discussed in Section 2.1.3, does allow for both f and g to be nonsmooth, but the convergence guarantee comes at the cost of an "inner loop" in the algorithm that increases in length with every iteration, which would be extremely inefficient in practice. The contribution of Theorem 1 is that we can be assured that, with the RSC assumption, the nonconvex ADMM algorithm will converge even when both f and g are nonsmooth.

3.5 Proof of Theorem 1

Fix any point (x, y, u) satisfying $Ax + By = c$. In Appendix A.2, we will prove that the assumption (10) on the step size matrices H_f, H_g ensures that, for all $T \geq 1$, there exist some $\xi_{x_2}, \ldots, \xi_{x_{T+1}}$ and some $\zeta_{y_1}, \ldots, \zeta_{y_T}$ such that

$$\sum_{t=0}^{T-1} \left\langle \begin{pmatrix} x_{t+2} - x \\ y_{t+1} - y \end{pmatrix}, \begin{pmatrix} \xi_{x_{t+2}} + A^\top u \\ \zeta_{y_{t+1}} + B^\top u \end{pmatrix} \right\rangle + \frac{1}{2} \sum_{t=0}^{T-1} \|Ax_{t+2} + By_{t+1} - c\|^2_\Sigma \leq C_1(x, y, u; x_0, y_0, u_0). \quad (11)$$

The function C_1 will be defined in the Appendix (see (27)).

Moreover, applying the restricted strong convexity assumption (Assumption 1), we have

$$\left\langle \begin{pmatrix} x_{t+2} - \tilde{x} \\ y_{t+1} - \tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_{x_{t+2}} - \xi_{\tilde{x}} \\ \zeta_{y_{t+1}} - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq \alpha_f \min\{\|x_{t+2} - \tilde{x}\|_2^2, c_f \|x_{t+2} - \tilde{x}\|_2\} + \alpha_g \min\{\|y_{t+1} - \tilde{y}\|_2^2, c_g \|y_{t+1} - \tilde{y}\|_2\} - \frac{1}{2}\|Ax_{t+2} + By_{t+1} - c\|^2_\Sigma - \varepsilon^2 \quad (12)$$

for each $t = 0, \ldots, T-1$.

Combining all of these calculations with the bound (11) above applied to $(x, y, u) = (\tilde{x}, \tilde{y}, \tilde{u})$, and rearranging terms, we obtain

$$\sum_{t=0}^{T-1} \left(\alpha_f \min\{\|x_{t+2} - \tilde{x}\|_2^2, c_f\|x_{t+2} - \tilde{x}\|_2\} + \alpha_g \min\{\|y_{t+1} - \tilde{y}\|_2^2, c_g\|y_{t+1} - \tilde{y}\|_2\})\right)$$
$$\leq \sum_{t=0}^{T-1} \left\langle \begin{pmatrix} x_{t+2} - \tilde{x} \\ y_{t+1} - \tilde{y} \end{pmatrix}, \begin{pmatrix} -A^\top \tilde{u} - \xi_{\tilde{x}} \\ -B^\top \tilde{u} - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle + C_1(\tilde{x}, \tilde{y}, \tilde{u}; x_0, y_0, u_0) + T\varepsilon^2.$$

Next for each t, we apply Assumption 2 to calculate

$$\begin{aligned}\langle x_{t+2} - \tilde{x}, -A^\top \tilde{u} - \xi_{\tilde{x}} \rangle &\leq \|x_{t+2} - \tilde{x}\|_2 \cdot \|-A^\top \tilde{u} - \xi_{\tilde{x}}\|_2 \\ &\leq \min\left\{\frac{\alpha_f c_f}{2} \cdot \|x_{t+2} - \tilde{x}\|_2, \sqrt{\alpha_f} \cdot \varepsilon_{\text{FOSP}} \cdot \|x_{t+2} - \tilde{x}\|_2\right\} \\ &\leq \min\left\{\frac{\alpha_f c_f}{2} \cdot \|x_{t+2} - \tilde{x}\|_2, \frac{\alpha_f}{2}\|x_{t+2} - \tilde{x}\|_2^2 + \frac{\varepsilon_{\text{FOSP}}^2}{2}\right\},\end{aligned}$$

and similarly for the y term. Therefore, we can rearrange the above to

$$\sum_{t=0}^{T-1} \left(\frac{\alpha_f}{2} \min\{\|x_{t+2} - \tilde{x}\|_2^2, c_f\|x_{t+2} - \tilde{x}\|_2\} + \frac{\alpha_g}{2} \min\{\|y_{t+1} - \tilde{y}\|_2^2, c_g\|y_{t+1} - \tilde{y}\|_2\})\right)$$
$$\leq C_1(\tilde{x}, \tilde{y}, \tilde{u}; x_0, y_0, u_0) + T\varepsilon^2 + T\varepsilon_{\text{FOSP}}^2.$$

Next, noting that x_1 is a deterministic function of (x_0, y_0, u_0), we define

$$C(x, y, u; x_0, y_0, u_0) = 4C_1(x, y, u; x_0, y_0, u_0) + 2\alpha_f \min\{\|x_1 - \tilde{x}\|_2^2, c_f\|x_1 - \tilde{x}\|_2\}.$$

We can then relax the bound above to

$$\sum_{t=0}^{T-1} \left(\frac{\alpha_f}{2} \min\{\|x_{t+1} - \tilde{x}\|_2^2, c_f\|x_{t+1} - \tilde{x}\|_2\} + \frac{\alpha_g}{2} \min\{\|y_{t+1} - \tilde{y}\|_2^2, c_g\|y_{t+1} - \tilde{y}\|_2\})\right)$$
$$\leq \frac{1}{4} C(\tilde{x}, \tilde{y}, \tilde{u}; x_0, y_0, u_0) + T\varepsilon^2 + T\varepsilon_{\text{FOSP}}^2. \quad (13)$$

Next we will use the following elementary fact: for any nonnegative $c, r_1, \ldots, r_n$,

$$\sum_{i=1}^n \min\{r_i^2, cr_i\} \geq \frac{n}{2} \min\left\{\left(\frac{1}{n}\sum_{i=1}^n r_i\right)^2, c\left(\frac{1}{n}\sum_{i=1}^n r_i\right)\right\}.$$

Therefore, applying this with $\|x_{t+1} - \tilde{x}\|_2$ in place of the r_i terms, we have

$$\sum_{t=0}^{T-1} \frac{\alpha_f}{2} \min\{\|x_{t+1} - \tilde{x}\|_2^2, c_f\|x_{t+1} - \tilde{x}\|_2\} \geq \frac{T\alpha_f}{4} \min\left\{\|\bar{x}_T - \tilde{x}\|_2^2, c_f\|\bar{x}_T - \tilde{x}\|_2\right\},$$

where the last step holds since $\frac{1}{T}\sum_{t=0}^{T-1}\|x_{t+1} - \tilde{x}\|_2 \geq \|\bar{x}_T - \tilde{x}\|_2$ by convexity. An analogous bound holds for the y term. Combining this with (13) completes the proof.

4. Example: sparse high-dimensional quantile regression

In this section, we will develop a concrete example of our framework, to illustrate the empirical performance and convergence properties of our method. Consider a regression setting where

$$w_i = \phi_i^\top \tilde{x} + (\text{noise}), \quad i = 1, \dots, n,$$

for a sparse true signal $\tilde{x} \in \mathbb{R}^d$. The response variables $w_i \in \mathbb{R}$ and the sensing matrix $\Phi = (\phi_1, \dots, \phi_n)^\top \in \mathbb{R}^{n \times d}$ are observed, and our goal is to recover $\tilde{x}$. If the noise is heavy-tailed, then a standard least-squares regression may perform poorly, and we may prefer the more robust properties of a quantile regression. Specifically, for any desired quantile $q \in (0, 1)$, consider the quantile loss

$$\ell_q(t) = q \cdot \max\{t, 0\} + (1 - q) \cdot \max\{-t, 0\}.$$

Then if we seek to minimize

$$\frac{1}{n} \sum_{i=1}^{n} \ell_q\big(w_i - \phi_i^\top x\big)$$

over $x \in \mathbb{R}^d$, this loss corresponds to aiming for $\phi_i^\top x$ to equal the q-th quantile of w_i. (Note that for the special case $q = 0.5$, i.e., median regression, this loss is equal to the ℓ_1 norm, up to rescaling.)

In the high-dimensional setting where $n < d$, minimizing this loss is not meaningful (in general, we can always find a vector $x \in \mathbb{R}^d$ that interpolates the data, i.e., $\phi_i^\top x = w_i$ for all i, which clearly leads to overfitting). We will therefore consider a penalized version of this loss:

$$\underset{x \in \mathbb{R}^d}{\arg\min} \big\{\text{Loss}(x)\big\} \text{ where } \text{Loss}(x) = \frac{1}{n} \sum_{i=1}^{n} \ell_q\big(w_i - \phi_i^\top x\big) + \lambda \sum_{j=1}^{d} \beta \log(1 + |x_j|/\beta). \tag{14}$$

The last term is a nonconvex regularizer that encourages a sparse solution; see Fazel et al. (2003); Candès et al. (2008) for background. For $\beta = +\infty$, the regularizer is equal to the ℓ_1 norm, a standard convex penalty for recovering sparse signals, while $\beta < +\infty$ leads to a nonconvex penalty. Smaller values of β correspond to greater nonconvexity, which makes the optimization problem more challenging but comes with the benefit of less shrinkage on the nonzero values in the signal vector x (see Figure 4).

To enable theoretical guarantees, we will add one small modification to this optimization problem, and will instead solve

$$\underset{x \in \mathbb{R}^d : \|x\|_2 \le R}{\arg\min} \big\{\text{Loss}(x)\big\} \tag{15}$$

for a large radius R, where this constraint is added to ensure that the iterations x do not diverge to infinity. We will see in our theoretical results that we can set R to be extremely large without compromising the convergence guarantee; in practice, therefore, we would expect that iteratively solving (15) would be indistinguishable from iteratively solving the unconstrained version (14), since the constraint $\|x\|_2 \le R$ would likely never be active.

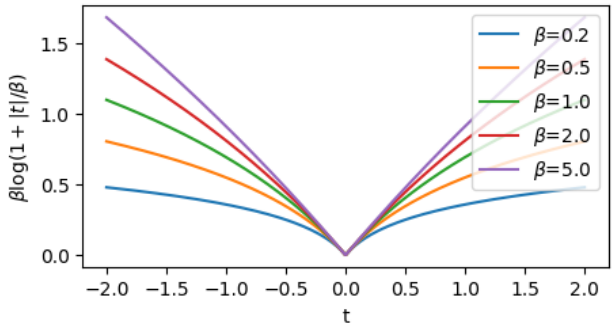

Figure 2: Illustration of the nonconvex sparsity-promoting penalty $\sum_j \beta \log(1+|x_j|/\beta)$ that appears in the objective function (14) for the sparse high-dimensional quantile regression example. The figure plots the function $t \mapsto \beta \log(1+|t|/\beta)$, for a range of values of β. The functions are all nondifferentiable at $t = 0$, and are similar to the absolute value function for $t \approx 0$, but smaller values of β correspond to greater nonconvexity as $|t|$ increases.

4.1 Implementing nonconvex ADMM

For the sparse quantile regression problem (15), we will introduce an additional variable y (with the constraint $y = \Phi x$) so that the optimization problem can be solved with Algorithm 1—we will minimize

$$\underset{x\in\mathbb{R}^d, y\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{n}\sum_{i=1}^{n} \ell_q\big(w_i - y_i\big) + \lambda \sum_{j=1}^{d} \beta \log(1+|x_j|/\beta) \;:\; y = \Phi x, \|x\|_2 \le R \right\}.$$

To solve (15), we define $A = \Phi$, $B = -\mathbf{I}_n$, and $c = 0$, and run Algorithm 1 with parameters $\Sigma = \sigma \mathbf{I}_n$ (for a chosen value of the tuning parameter $\sigma > 0$), $H_f = \sigma(\gamma \mathbf{I}_d - \Phi^\top \Phi)$ (with $\gamma = \|\Phi\|^2$ so that $H_f \succeq 0$), and $H_g = 0$, and with functions

$$f_c(x) = \lambda\|x\|_1 + \delta_{\|x\|_2\le R}, \quad f_d(x) = \lambda \sum_{j=1}^{d} \left(\beta \log(1+|x_j|/\beta) - |x_j|\right),$$

where $\delta_{\|x\|_2\le R}$ is the convex indicator function (i.e., $\delta_{\|x\|_2\le R} = 0$ if $\|x\|_2 \le R$, and $\delta_{\|x\|_2\le R} = +\infty$ otherwise), and with

$$g_c(y) = \frac{1}{n}\sum_{i=1}^{n} \ell_q\big(w_i - y_i\big), \quad g_d(y) \equiv 0.$$

The update steps for Algorithm 1 can be calculated in closed form (details are given in Appendix A.4). We note that the function f_d is concave and twice differentiable, with $\nabla^2 f_d(x) \succeq -\lambda\beta^{-1}\mathbf{I}_d$ for all x, so its concavity is bounded.

4.2 Theoretical results

Our theoretical results guarantee convergence for the nonconvex ADMM algorithm as long as the RSC property (6) and the the approximate first-order optimality property (9) both hold, to verify the assumptions of Theorem 1. In particular, RSC-type properties for sparse high-dimensional quantile regression have been studied in the literature, e.g., see Zhao et al. (2014, Lemma C.3) or Belloni and Chernozhukov (2011, Lemma 4). The conditions proved in the literature appear in a different form than the RSC property studied here, so we verify that the property (6) holds under some mild assumptions. The following result is proved in Appendix A.5.

Proposition 2 *Suppose that the observations are given by*

$$w_i = \phi_i^\top \tilde{x} + z_i, \quad i = 1, \dots, n$$

for some sample size $n \geq 4$, and let $\tilde{y} = \Phi \tilde{x}$. Assume that:

- *The feature vectors $\phi_i \in \mathbb{R}^d$ are i.i.d. with distribution $\mathcal{D}_\phi$, where for $\phi \sim \mathcal{D}_\phi$, it holds that $\|\phi\|_\infty \leq B_\phi$ almost surely, and that $\mathbb{E}\left[|\phi^\top u|^2\right] \geq a_\phi$ and $\mathbb{E}\left[|\phi^\top u|^3\right] \leq b_\phi$ for any fixed unit vector $u \in \mathbb{R}^d$;*
- *The noise terms $z_i \in \mathbb{R}$ are drawn independently from the feature vectors ϕ_i, and moreover are i.i.d. with density h_z, for which $z = 0$ is the q-th quantile, and which satisfies $h_z(t) \geq c_z$ for all $|t| \leq t_z$, for some $c_z, t_z > 0$;*
- *The true vector $\tilde{x}$ has at most s_* nonzero entries, where*

  $$1 \leq s_* \leq C_0 \cdot \frac{n}{\log(nd)}$$

  for a constant $C_0 > 0$ that depends only on $c_z, t_z, a_\phi, b_\phi, B_\phi$;
- *The parameters λ, β, R are chosen to satisfy*

  $$\lambda = C_\lambda \sqrt{\frac{\log(nd)}{n}} \text{ for some } C_\lambda \in \left[C_1, C_1 \sqrt{\frac{C_0 \cdot \frac{n}{\log(nd)}}{s_*}} \right]$$

  and

  $$R \geq \|\tilde{x}\|_2 \quad \text{and} \quad \beta \geq C_\lambda \max\{1, R\} \cdot C_2 \sqrt{\frac{\log(nd)}{n}},$$

  for constants $C_1, C_2 > 0$ that depend only on $c_z, t_z, a_\phi, b_\phi, B_\phi$.

Then, for any $\sigma > 0$, with probability at least $1 - (nd)^{-1}$, the RSC property (6) holds with

$$\alpha_f = C_3, \quad \alpha_g = 0, \quad c_f = c_g = 1, \quad \Sigma = \sigma \mathbf{I}_n, \quad \varepsilon^2 = C_4 \max\{1, \sigma^{-1}\} \cdot \frac{s_* \log(nd)}{n},$$

and the approximate first-order optimality property (9) holds with

$$\varepsilon_{\text{FOSP}}^2 = C_5 \cdot \frac{s_* \log(nd)}{n},$$

where $C_3, C_4, C_5 > 0$ are constants that depend only on $c_z, t_z, a_\phi, b_\phi, B_\phi$ and on C_λ.

With this result in place, if λ, β, R are chosen appropriately, then Theorem 1 ensures that, after T iterations of ADMM, the estimate $\bar{x}_T$ will satisfy

$$\min\{\|\bar{x}_T - \tilde{x}\|_2^2, \|\bar{x}_T - \tilde{x}\|_2\} \leq \mathcal{O}\left(\frac{1}{T} + \frac{s_* \log(nd)}{n}\right),$$

which we can simplify to

$$\|\bar{x}_T - \tilde{x}\|_2 \leq \mathcal{O}\left(\sqrt{\frac{1}{T} + \frac{s_* \log(nd)}{n}}\right).$$

In contrast, the minimax error rate for estimating $\tilde{x}$, in this high-dimensional sparse regression setting, is $\mathcal{O}\left(\sqrt{\frac{s_* \log(d/s_*)}{n}}\right)$ (Raskutti et al., 2011, Theorem 1(b)). This shows that, up to a slightly different log factor, the error of $\bar{x}_T$ matches the minimax rate once T is sufficiently large.

Comparing to existing theory As discussed in Section 2.1.2, previous results establishing convergence for nonconvex ADMM assume, at minimum, that either f or g is differentiable and has a Lipschitz gradient. We can see immediately that this property is violated for the sparse quantile regression problem (14) (or for its constrained version (15)), since the functions f and g are both nondifferentiable. In contrast, our new RSC-based framework is able to provide a guarantee, and so this example illustrates the flexibility and broad applicability of RSC type assumptions, as compared to other assumptions in the literature.

4.3 Empirical results

We next demonstrate the performance of our algorithm on the sparse quantile regression problem. Code reproducing the simulation and all figures is available at `https://github.com/rinafb/ADMM_CT`.

We choose dimension $d = 2500$ and sample size $n = 2000$ for a challenging high-dimensional setting. The matrix $\Phi \in \mathbb{R}^{n \times d}$ is constructed with i.i.d. $\mathcal{N}(0,1)$ entries. We define

$$w_i = \phi_i^\top \tilde{x} + z_i,$$

where ϕ_i is the ith row of Φ, and the true signal is given by $\tilde{x} = (1, \ldots, 1, 0, \ldots, 0)$, with $s_* = 10$ nonzero entries. The noise terms z_i are drawn i.i.d. from t_5, the standard t distribution with 5 degrees of freedom, which is a heavy-tailed distribution. We choose the quantile $q = 0.5$ (i.e., a median regression). For the penalty term, we choose $\lambda = 0.1$ and $\beta = 0.5$; this small value of β means that the penalty has substantial nonconvexity (see Figure 4). The parameter σ controlling the enforcement of the constraint in ADMM (i.e., with $\Sigma = \sigma \mathbf{I}_d$ in Algorithm 1) is varied as $\sigma \in \{0.00005, 0.0001, 0.0002, 0.0005\}$.

The results after running Algorithm 1 for 1000 iterations are displayed in Figure 3. The plot displays the loss, $\text{Loss}(x_t)$ at each iteration t, where $\text{Loss}(\cdot)$ is the objective function defined in (14), as well as the root-mean-square-error (RMSE), $\frac{1}{\sqrt{d}}\|x_t - \tilde{x}\|_2$. (We do not impose a constraint $\|x\|_2 \leq R$, since as mentioned above, the theory allows for R to be extremely large, and the iterations x_t do not violate this constraint in practice.) The plot also shows $\text{Loss}(\bar{x}_t)$ and $\frac{1}{\sqrt{d}}\|\bar{x}_t - \tilde{x}\|_2$, the loss and RMSE of the running average of the

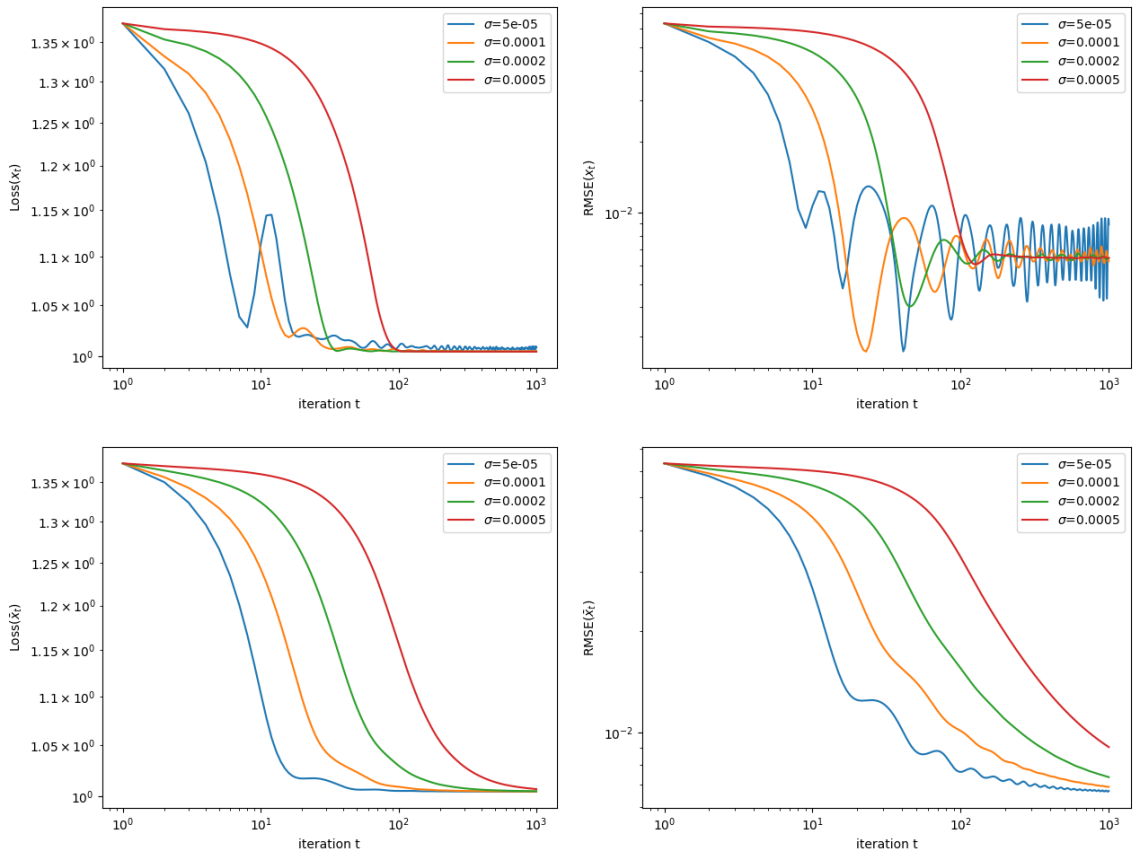

Figure 3: Results for the sparse quantile regression example (see Section 4.3). The figure shows the value of the objective function (14) over iteration $t = 1, \ldots, 500$ of the algorithm, run with various values of the parameter σ as shown. The top row shows the loss function value for x_t (the estimate at time t), as well as its root-mean-square-error (RMSE) $\frac{1}{\sqrt{d}}\|x_t - \tilde{x}\|_2$, while the bottom plot shows the loss and the RMSE for $\bar{x}_t$ (the running average). All axes are on the log scale.

estimates, $\bar{x}_t = \frac{1}{t}\sum_{t'=1}^{t} x_{t'}$. The convergence of the loss and RMSE for $\bar{x}_t$ across all σ values is supported by our theoretical result, Proposition 2, which shows that the RSC property holds (with high probability) for *any* $\sigma > 0$, as long as the tolerance term ε is adjusted accordingly. Note that the RMSE (for both x_t and $\bar{x}_t$) does not converge to zero, but instead appears to be converging to a small but positive value; this is due to the noise in the data.

Interestingly, we see that overly small values of σ lead to some instability in the convergence of the loss and the RMSE, suggesting that the RSC property may not be sufficient to ensure convergence of the iterates themselves (the x_t's) rather than the running averages (the $\bar{x}_t$'s).[1] On the other hand, overly large values of σ may lead to somewhat slower convergence; intuitively, enforcing the constraint $y = \Phi x$ with too strong of a penalty will make it difficult for the algorithm to make fast progress with alternating updates of x and y.

5. Application: CT imaging

We next apply our algorithm and convergence results to the problem of image reconstruction in computed tomography (CT) imaging, which is the motivating application for this work. In CT, we would like to reconstruct an image of an unknown object x (e.g., produce a 3D image of a patient's head or abdomen, in the setting of medical CT). The available measurements obtained from the CT scanner consist of measuring the intensity of an X-ray beam passing through the unknown object. A lower intensity of the beam when it reaches the detector indicates higher density in the unknown object along that ray.

We now introduce some notation to make this problem more precise. We will begin with a simple version of the problem, and then will add additional components step by step to build intuition. Let $x = (x_k) \in \mathbb{R}^{n_k}$ denote the unknown image, where $k = 1, \ldots, n_k$ indexes pixels (or voxels), after we have discretized to a 2D (or 3D) grid—for example, in two dimensions, $n_k = N_x \cdot N_y$ for an $N_x \times N_y$ grid.

To obtain an image, the scanner sends an X-ray beam along n_ℓ many rays. For example, for many clinical scanners in a medical setting, the device rotates around the patient, taking images from N_{img} many angles; for each of these images, there are N_{cell} many detector cells measuring the intensity of the beam after it passes through the patient's body. This leads to $n_\ell = N_{\text{img}} \cdot N_{\text{cell}}$ many rays $\ell = 1, \ldots, n_\ell$ along which measurements are taken.

Now let $P = (P_{\ell k}) \in \mathbb{R}^{n_\ell \times n_k}$ be the projection matrix, with $P_{\ell k}$ measuring the length of the intersection between ray ℓ and pixel k. The product $Px \in \mathbb{R}^{n_\ell}$ measures the projection of the object x, where $(Px)_\ell$ measures the total amount of material that lies along ray ℓ (see Figure 4 for a schematic). The attenuation (i.e., the loss of intensity) of the X-ray beam that travels along ray ℓ depends on $(Px)_\ell$. In particular, ignoring photon scattering and other sources of noise, the measurements follow a model of the form

$$\frac{\text{Intensity of the beam after passing through the object along ray } \ell}{\text{Intensity of the beam entering the object along ray } \ell} \approx e^{-\mu \cdot (Px)_\ell},$$

1. An alternative explanation for this empirical result is simply that the parameter ε in the RSC property (6) is larger, when σ is chosen to be smaller, as in Proposition 2; since convergence is only guaranteed up to the tolerance level ε in Theorem 1, this may explain the apparent lack of convergence for x_t when σ is chosen to be very small.

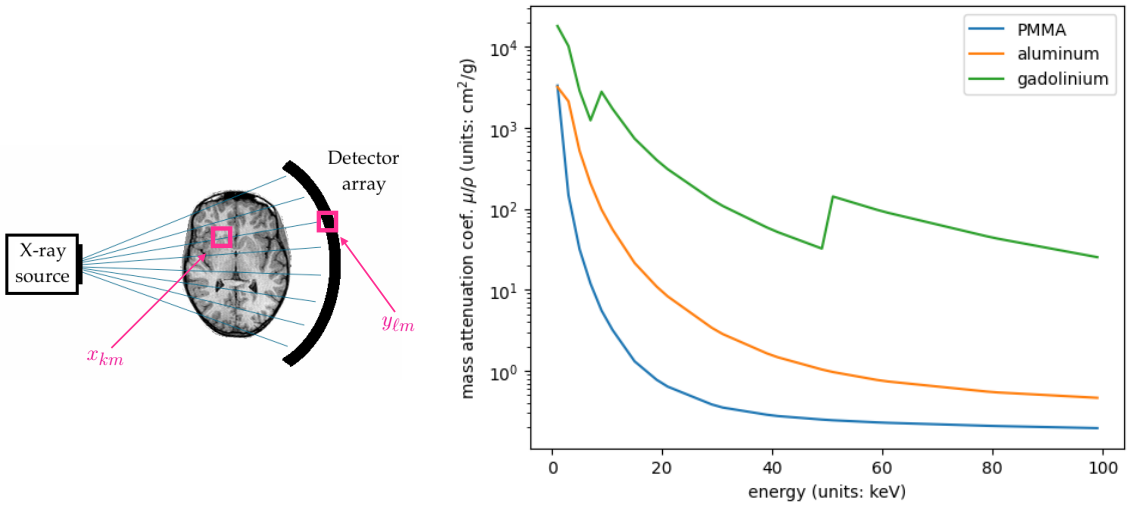

Figure 4: Left: schematic of the projection operator. Here x_{km} is the amount of material m present at pixel k, while $y_{\ell m} = (Px)_{\ell m}$ is the total amount of material m present along ray ℓ of the scan. Right: attenuation curves for several common materials.

where $\mu > 0$ is called the linear attenuation coefficient. While most clinical scanners measure the total energy of the beam when it reaches the detector, here we consider a different type of hardware, *photon counting CT*, where the measurement is a count of the number of photons reaching the detector. In this case, we can model this count as

$$C_\ell \sim \text{Poisson}\left(S \cdot \exp\{-\mu \cdot (Px)_\ell\}\right),$$

where S is the number of photons incident on the detector pixel (characterizing the intensity of the X-ray beam for a fixed time-duration scan), and C_ℓ is the number of photons reaching detector after passing through the object along ray ℓ.

In fact, since different detector cells may have slightly different sensitivities, a more accurate model is

$$C_\ell \sim \text{Poisson}\left(S_\ell \cdot \exp\{-\mu \cdot (Px)_\ell\}\right), \tag{16}$$

where the scalar term S_ℓ combines beam intensity with detector sensitivity for ray ℓ.

Multiple materials In practice, the unknown object can consist of multiple materials, which each behave differently in terms of the attenuation of the beam. Let $m = 1, \ldots, n_m$ index the materials that make up the object—for example, in a simple medical setting we might have $n_m = 3$ with bone, soft tissue, and an injected contrast material such as a gadolinium or iodine compound. The goal is now to reconstruct the image $x = (x_{km}) \in \mathbb{R}^{n_k \times n_m}$, where, for each pixel k, x_{km} is the proportion of that pixel that is occupied by each material. We can update our model (16) above to

$$C_\ell \sim \text{Poisson}\left(S_\ell \cdot \exp\left\{-\sum_m \mu_m \cdot (Px)_{\ell m}\right\}\right), \tag{17}$$

where now $\mu_m > 0$ is the (known) linear attenuation coefficient for material m.

A non-monochromatic beam Thus far, the Poisson model for CT image reconstruction does not introduce nonconvexity—maximizing the log-likelihood of the Poisson model given in (17) is a convex problem. However, this model ignores the nature of the X-ray beam used in practice, for which the photons are distributed across a spectrum of energies. The attenuation coefficient for a material m in fact depends on the energy of the photon, with each material exhibiting its own attenuation curve across the range of energies—see Figure 4 for an example. In particular, in medical applications, contrast materials such as gadolinium or iodine are used for their unique attenuation curves, which make these materials easier to distinguish from surrounding soft tissue in a CT scan.

Our model can now be updated to the following:

$$C_\ell \sim \text{Poisson}\left(\sum_i S_{\ell i} \cdot \exp\left\{-\sum_m \mu_{mi}(Px)_{\ell m}\right\}\right), \tag{18}$$

where $i = 1, \ldots, n_i$ is the index over a discretized grid of the range of energies in the X-ray beam, while $S_{\ell i}$ is the intensity of the X-ray beam (combined with detector sensitivity) for energy level i and ray ℓ, and μ_{mi} is the attenuation coefficient for material m at energy level i. The photons measured by the detector may come from any energy level in the spectrum (i.e., the measurements C_ℓ are a combination of photons from each energy level i). The resulting log-likelihood maximization problem is no longer a convex function, which is a core challenge of CT image reconstruction.

Spectral CT In spectral CT, the hardware of the scanner allows partial identification of the photon energies, making the reconstruction problem somewhat easier. Specifically, the detectors are programmed with several thresholds, separating the range of energies of the beam into "windows" $w = 1, \ldots, n_w$ (for example, 2 windows in some current clinical scanners, or 3–5 windows in current research prototypes). The measurements are now indexed by $C_{w\ell}$, the number of photons in energy window w measured along ray ℓ. In theory, the windows form a partition of the energy range, but in practice there is some noise at the boundaries between windows (that is, a photon with energy near the chosen threshold has some chance of being detected in either window). To quantify this, let $S_{w\ell i}$ incident photon spectral density at energy i, multiplied by the probability of a photon at energy i being detected in window w (for the detector sensitivity corresponding to ray ℓ). These values are typically estimated ahead of time with a calibration process. Then the model for our measurements $C_{w\ell}$ is given by

$$C_{w\ell} \sim \text{Poisson}\left(\sum_i S_{w\ell i} \cdot \exp\left\{-\sum_m \mu_{mi}(Px)_{\ell m}\right\}\right). \tag{19}$$

We can estimate the image x by maximum likelihood estimation, but as before in (18), maximizing the log-likelihood is a non-convex problem. (See Barber et al. (2016) for more details on this model.)

5.1 Image reconstruction with nonconvex ADMM

We now consider the image reconstruction problem: given observations (photon counts) $C_{w\ell}$, we would like to solve

$$\tilde{x} = \underset{x \in \mathbb{R}^{n_k \times n_m}}{\arg\min}\ \mathrm{Loss}(Px), \tag{20}$$

where $\mathrm{Loss}(y)$ is the negative log-likelihood of the Poisson model for spectral CT (19) given the projected object $y = Px \in \mathbb{R}^{n_\ell \times n_m}$:

$$\mathrm{Loss}(y) = \sum_{w\ell} \left[\sum_i S_{w\ell i} \exp\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\} - C_{w\ell} \log\left(\sum_i S_{w\ell i} \exp\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\} \right) \right].$$

We note that the first term of this loss is convex in y (and therefore, in x), while the second term is concave.

Modifying the exp function Under a well-specified model, the true image x and its projection $y = Px$ must both consist of nonnegative values. However, model misspecification, or inaccurate estimates of x and/or y at early stages of the iterative algorithm, can lead to negative values. Examining the loss function, we can see that this issue may pose problems for optimization, since $t \mapsto \exp\{t\}$ has high curvature at large values of t. To resolve this, we replace the $\exp\{\cdot\}$ function with the approximation:

$$\mathrm{qexp}\{t\} = \begin{cases} \exp\{t\}, & t \le 0, \\ 1 + t + \frac{1}{2}t^2, & t \ge 0. \end{cases}$$

The "q" in the name of this modified function refers to the fact that, for positive values of t we replace $\exp\{t\}$ with a quadratic approximation, by taking the Taylor expansion at $t = 0$. For negative values of t, the function is unchanged. This choice means that the function $\mathrm{qexp}\{t\}$ is continuously twice differentiable and is equal to $\exp\{t\}$ at all negative values of t (i.e., for any feasible nonnegative image x), while at the same time ensuring a bounded second derivative to avoid problems in the optimization. We will therefore work with a modified loss function,

$$\mathrm{Loss}(y) = \sum_{w\ell} \left[\sum_i S_{w\ell i} \,\mathrm{qexp}\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\} - C_{w\ell} \log\left(\sum_i S_{w\ell i} \,\mathrm{qexp}\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\} \right) \right].$$

It is important to note that, for CT imaging, if the model is well specified then the argument to $\exp\{\cdot\}$ or to $\mathrm{qexp}\{\cdot\}$ should always be nonpositive at the true $\tilde{y} = P\tilde{x}$ (i.e., $\sum_m \mu_{mi} y_{\ell m}$ should be nonnegative at the true $\tilde{y}$), and therefore, $\mathrm{qexp}\{\cdot\}$ should be identical to $\exp\{\cdot\}$ in the relevant range of values. Empirically, however, the convergence behavior of the optimization problem is often helped by allowing both positive and negative values, particularly in early iterations, and this can also provide useful flexibility in the case of model misspecification.

Running nonconvex ADMM To reformulate the minimization problem (20) into the setting of nonconvex ADMM, we will solve the equivalent problem

$$\tilde{x}, \tilde{y} = \underset{\substack{x \in \mathbb{R}^{n_k \times n_m} \\ y \in \mathbb{R}^{n_\ell \times n_m}}}{\arg\min}\ \{\mathrm{Loss}(y)\ :\ Px = y\}. \tag{21}$$

Now define $f(x) = f_c(x) = f_d(x) \equiv 0$, and write $g(y) = g_c(y) + g_d(y)$ where

$$g_c(y) = \sum_{w\ell} \sum_i S_{w\ell i} \operatorname{qexp}\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\} \tag{22}$$

and

$$g_d(y) = -\sum_{w\ell} C_{w\ell} \log\left(\sum_i S_{w\ell i} \operatorname{qexp}\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\} \right).$$

Then $\text{Loss}(y) = g(y)$, and we have therefore reformulated the spectral CT maximum likelihood estimation problem into the form of our nonconvex ADMM algorithm, i.e., $\min_{x,y}\{f(x) + g(y) : Px = y\}$, minimizing a composite objective function under a linear constraint. In particular, converting the matrix variables $x \in \mathbb{R}^{n_k \times n_m}$ and $y \in \mathbb{R}^{n_\ell \times n_m}$ to vectorized variables $\operatorname{vec}(x) \in \mathbb{R}^{n_k n_m}$ and $\operatorname{vec}(y) \in \mathbb{R}^{n_\ell n_m}$, the constraint $Px = y$ can be rewritten as $A\operatorname{vec}(x) + B\operatorname{vec}(y) = c$ where $A = P \otimes \mathbf{I}_{n_m} \in \mathbb{R}^{n_\ell n_m \times n_k n_m}$, $B = -\mathbf{I}_{n_\ell n_m}$, and $c = 0$ (here $\otimes$ denotes the matrix Kronecker product).

We can therefore implement Algorithm 1 for solving this optimization problem. To run Algorithm 1 for the CT image reconstruction problem (21), we need to choose the step size matrices H_f, H_g and the penalty matrix Σ. Following the construction proposed by Pock and Chambolle (2011) (for the convex setting), we begin by selecting a parameter $\sigma > 0$. We will choose step size matrix $H_g = 0$ for y, while for the variable x our step size matrix H_f will be equal to $H_f = (Q_f \otimes \mathbf{I}_{n_m}) - (P \otimes \mathbf{I}_{n_m})^\top \cdot (\tilde{\Sigma} \otimes \mathbf{I}_{n_m}) \cdot (P \otimes \mathbf{I}_{n_m})$, and the penalty parameter matrix Σ will be defined as $\Sigma = \tilde{\Sigma} \otimes \mathbf{I}_{n_m}$, where $Q_f \in \mathbb{R}^{n_k \times n_k}$ and $\tilde{\Sigma} \in \mathbb{R}^{n_\ell \times n_\ell}$ are diagonal matrices with entries

$$(Q_f)_{kk} = \sigma \sum_\ell P_{\ell k}, \quad \tilde{\Sigma}_{\ell\ell} = \frac{\sigma}{\sum_k P_{\ell k}}.$$

With these constructions, H_f is positive semidefinite as required (Pock and Chambolle, 2011, Lemma 2). The update steps for the nonconvex ADMM algorithm are computed in Appendix A.3.

5.2 CT simulation

To demonstrate the algorithm's performance on the nononvex CT image reconstruction problem, we carry out a small-scale simulation in Python. (Performance of these methods on a large scale requires more careful implementation, and is addressed in our application specific work in Barber et al. (2016); Schmidt et al. (2020).) Code reproducing the simulation and all figures is available at `https://github.com/rinafb/ADMM_CT`.

The ground truth, shown in Figure 6, is a 10cm×10cm two-dimensional image discretized to a 25×25 grid, for a total of $n_k = 25^2 = 625$ pixels. The image consists of $n_m = 3$ materials—polymethyl methacrylate (PMMA), aluminum, and gadolinium. As shown in Figure 4, PMMA has low attenuation coefficients as it is a plastic, while aluminum, like other metals, has higher attenuation coefficients as it is more difficult for the beam to pass through. Gadolinium is a contrast material used in clinical CT—its non-monotone attenuation curve allows for it to be easily identified in the presence of other materials. The simulated CT scanner has 50 detector cells, and takes images from 50 angles spaced evenly

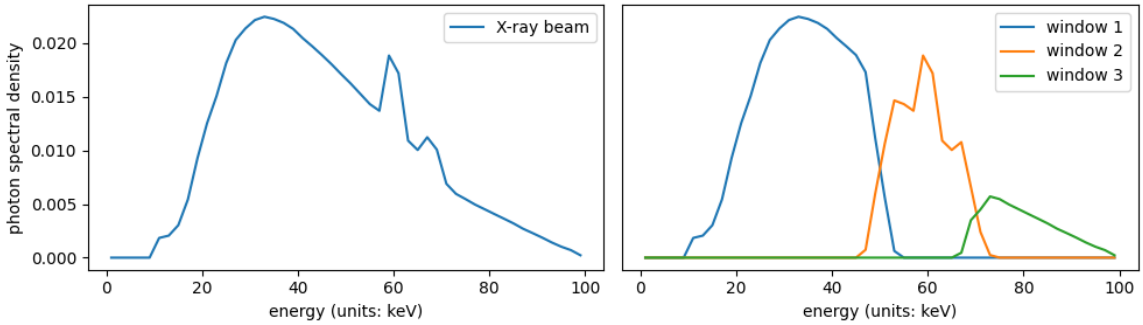

Figure 5: Left: the X-ray beam spectrum. This figure displays the density of the distribution of energies in the beam, i.e., how the total intensity of the beam is split across the energy spectrum. Right: for each energy window w, the displayed curve is proportional to the spectral response parameters $S_{w\ell i}$. These values are set to be constant across all rays ℓ, and so the figure plots the value across all energy levels i for each detector window w, rescaled so that the sum of the three response curves is equal to the density plot of the X-ray beam spectrum on the left.

around the unit circle, for a total of $n_\ell = 50^2 = 2500$ rays along which measurements are taken. The beam intensity is set to 10^6 photons, and there are $n_w = 3$ energy windows, forming a blurry partition of the energy range (see Figure 5).

Figure 6 displays the estimated image (shown at iteration 1000, at each value of the ADMM parameter $\sigma \in \{1, 10, 100\}$). In Figure 7 we show the loss function $\text{Loss}(Px_t)$, and the RMSE $\frac{1}{\sqrt{n_k}}\|x_t - \tilde{x}\|_2$, at each iteration $t = 1, \ldots, 1000$. As expected, due to the noise in the measurements, the RMSE converges to a small but positive value. We can see that the algorithm converges steadily towards minimizing the loss and reducing the RMSE, and its performance is reasonably stable and robust across a wide range of values of the tuning parameter σ.

Extensions The objective function, and accompanying algorithm, that we have presented here, can easily be modified to incorporate additional components such as regularization or constraints. In particular, total variation regularization can also be incorporated into the framework of Algorithm 1.[2]. Another possible modification is adding a preconditioning step to improve the conditioning in the n_m-dimensional material space—since the attenuation curves for the three materials are quite similar (see Figure 4), adding a preconditioning step can improve convergence substantially for the image reconstruction problem (see Sidky et al. (2018) for more details). The algorithm, together with these extensions, has been implemented for large-scale CT data, and has achieved promising empirical results for both

2. Details and a demonstration can be found with the code accompanying this paper (`https://github.com/rinafb/ADMM_CT`), alongside the basic non-regularized simulation setting presented here. In addition, this code also shows results from an experiment in a noisier setting, with beam intensity set to 10^5 rather than 10^6 for a lower signal-to-noise ratio

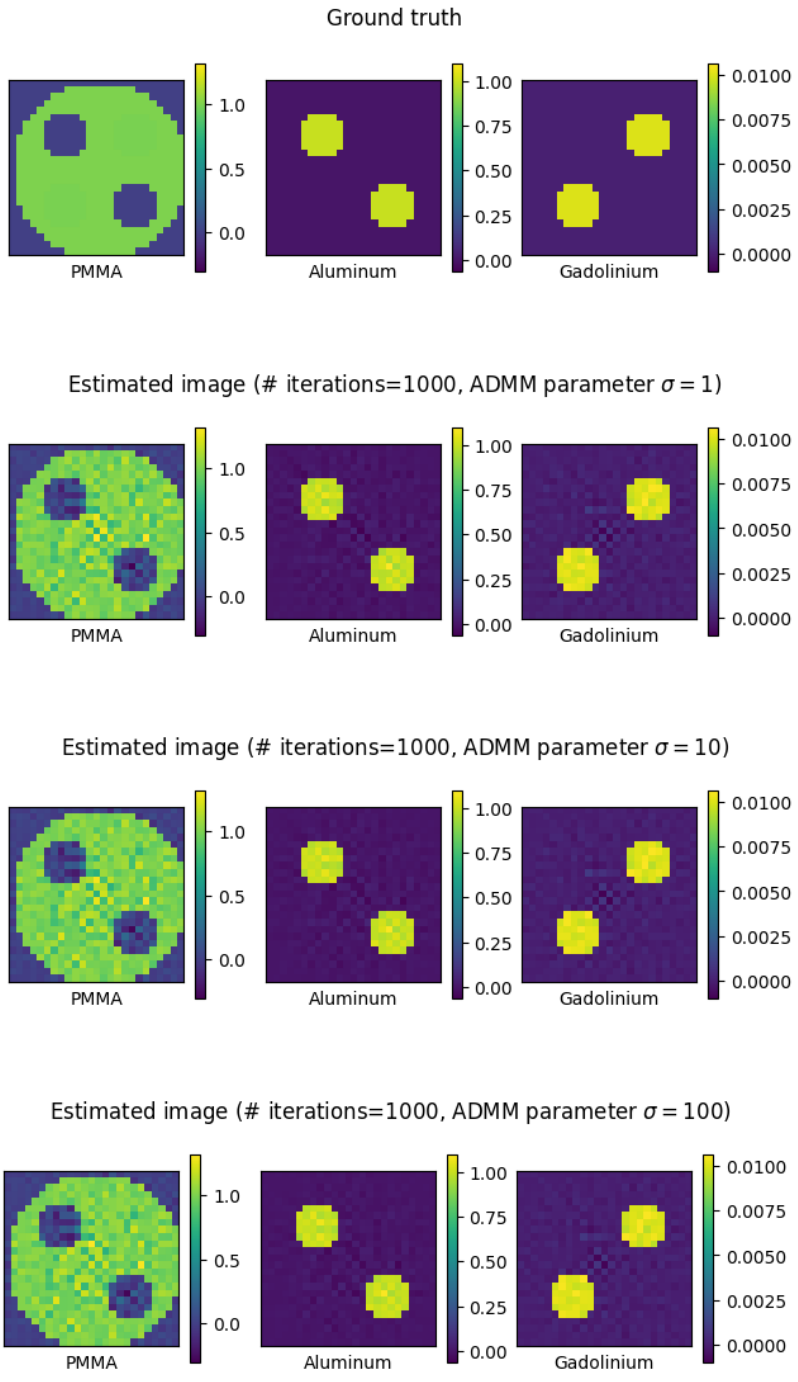

Figure 6: The true image in the simulation (top), followed by the reconstructed image (at iteration 1000) with each value of the ADMM parameter σ. Each row of images displays the values of x_{km} for each pixel k and each material m, for $x = \tilde{x}$ (for the ground truth) or $x = x_t$ (for the estimates).

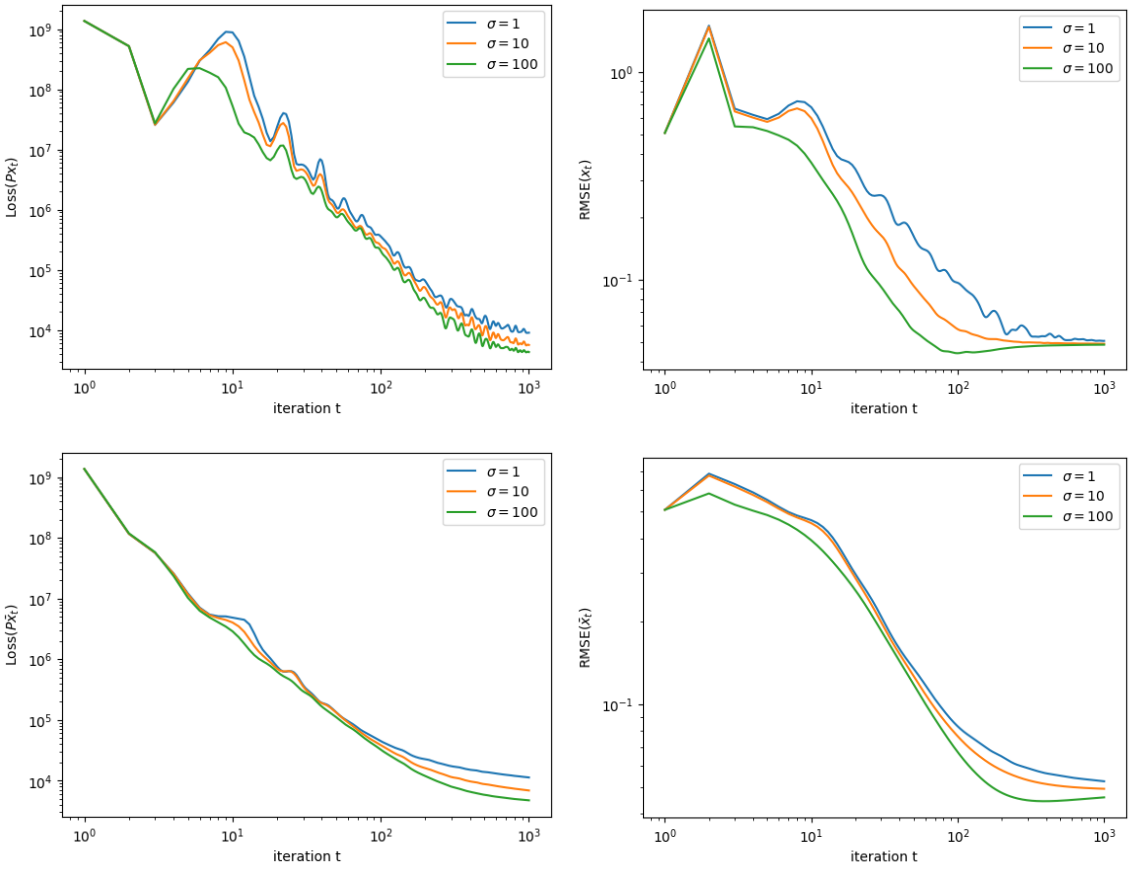

Figure 7: Convergence results for the CT image reconstruction simulation. The top row of the figure shows the value of the objective function $\text{Loss}(Px_t)$, and the RMSE $\frac{1}{\sqrt{n_k}}\|x_t - \tilde{x}\|_2$, over iteration $t = 1, \ldots, 1000$ of the algorithm, run with various values of the parameter σ as shown. The bottom row shows the same for the running average $\bar{x}_t$ in place of x_t. All axes are on the log scale.

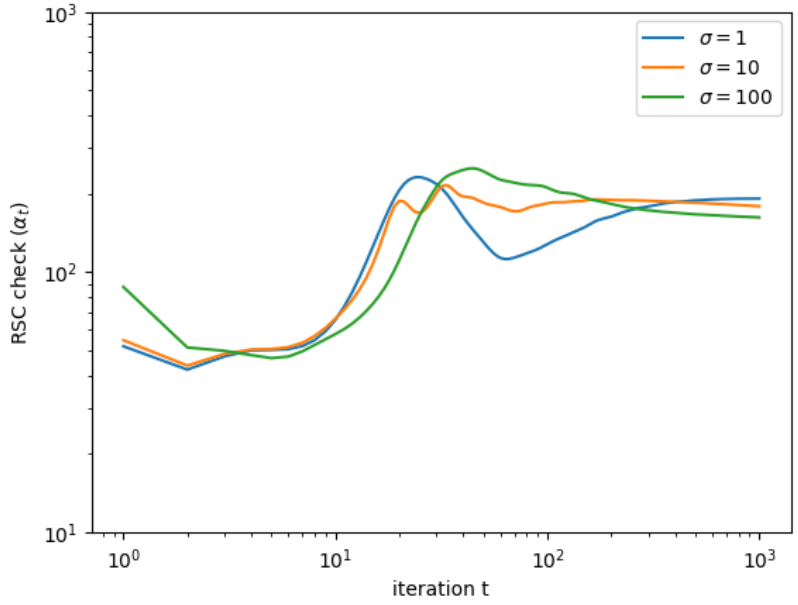

Figure 8: The figure shows the value of α_t (defined in (23)), which empirically validates the restricted strong convexity property for the CT imaging example. Both axes are on the log scale.

real CT data and simulation studies, e.g., in Schmidt et al. (2022); Rizzo et al. (2022); Schmidt et al. (2023); Rizzo et al. (2023).

Checking assumptions For the CT imaging example, it is not clear whether it is possible to establish the RSC property (6) theoretically. However, since we are in simulated setting where the target parameters $(\tilde{x}, \tilde{y})$ are known, we can nonetheless validate it empirically. For this example, since $f(x) \equiv 0$, it suffices to check that, for some $\alpha > 0$,

$$\langle y - \tilde{y}, \nabla g(y) - \nabla g(\tilde{y}) \rangle \geq \alpha \|\mathrm{vec}(y - \tilde{y})\|_2^2 - \frac{1}{2}\|\mathrm{vec}(Px - y)\|_\Sigma^2$$

holds for all x, y (here $\tilde{y} = P\tilde{x}$). If this is true, then the RSC property holds with $\alpha_f = 0$, $\alpha_g = \alpha$, $c_f = c_g = 1$, and $\varepsilon = 0$.

However, it is not feasible to verify this over all possible y, so we will instead verify that this holds for $y = y_t$ at each iteration t of the algorithm. (In fact, examining how the RSC assumption is used in the proof of Theorem 1, we see in (12) that the RSC assumption is only applied at values of x and y appearing along the iterations of the algorithm—specifically, at points (x, y) of the form (x_{t+1}, y_t) at each time t. In other words, for the proof of Theorem 1 to hold for the CT example, where we have $f(x) \equiv 0$, we only need to check that the inequality above holds at each iteration y_t, rather than at all values of y.)

To verify this, we calculate

$$\alpha_t := \frac{\langle y_t - \tilde{y}, \nabla g(y_t) - \nabla g(\tilde{y}) \rangle + \frac{1}{2}\|\mathrm{vec}(Px_{t+1} - y_t\|_\Sigma^2}{\|\mathrm{vec}(y_t - \tilde{y})\|_2^2}, \tag{23}$$

where x_{t+1} and y_t denote the iterates of the algorithm, while $\tilde{y} = P\tilde{x}$ is the projection of the true image. If the RSC property holds as above, then we should see $\alpha_t \geq \alpha$ for all t, for some constant $\alpha > 0$. Indeed, for the simulated example, Figure 8 shows that α_t remains bounded away from zero across all iterations of the algorithm. This validates Assumption 1.

Finally, we verify that approximate first-order optimality (9) holds in this setting. Choosing $\tilde{u} = 0$, we can see that (9) holds as long as $\|\nabla g(\tilde{y})\|_2$ is low. For our simulation, we compare $\|\nabla g(\tilde{y})\|_2$ to $\|\nabla g(0)\|_2$ (in order for our calculations to be on a meaningful scale), and we find that

$$\frac{\|\nabla g(\tilde{y})\|_2}{\|\nabla g(0)\|_2} = 0.000769,$$

verifying that approximate first-order optimality holds.

6. Discussion

The ADMM algorithm has long been known to perform well in a broad range of challenging scenarios, but existing theoretical analyses are largely restricted to a much more constrained range of settings. Our new theoretical results provide a novel understanding of the performance of ADMM in the presence of nonsmoothness and nonconvexity in the objective functions, through the lens of a restricted strong convexity property. A key nonconvex application of this algorithm is the CT image reconstruction problem, where many interesting open questions remain. In particular, for real CT scanner data, it is important to calibrate the beam intensity and detector sensitivity parameters that characterize the performance of the detector. In future work, we aim to extend the ADMM formulation of the image reconstruction problem to allow for simultaneous estimation of the calibration parameters (a preliminary study of the simultaneous estimation approach can be found in Ha et al. (2018)). Incorporating more complex aspects of the physical model, such as scatter, poses an additional challenge that we hope to address in future work to provide a more accurate reconstructed image.

From the theoretical perspective, a key remaining question is whether the RSC property can be further relaxed to allow for convergence guarantees in an even broader range of settings. On the other hand, the RSC property does not appear to be sufficient to ensure convergence of the iterates x_t, y_t (rather than the running averages $\bar{x}_t, \bar{y}_t$), as was seen in the quantile regression example. An important open question is whether a stronger form of the RSC property would allow for convergence guarantees without averaging. From the practical side, another important question is the issue of optimization with a stochastic, or mini-batch, approach—analogous to stochastic gradient descent, the ADMM algorithm can be run using stochastic approximations to gradients at each step (see, e.g., Zhong and Kwok (2014)), leading to computational speedup, and can be immensely helpful for allowing the method to be applied to large scale applications (including CT imaging, see, e.g., Nien and Fessler (2014)). Another important open question, therefore, is whether the theoretical results of this work for convergence in a nonconvex setting can be extended to the stochastic version of the ADMM algorithm. The empirical performance of the algorithm might also be improved by incorporating techniques such as adaptive restart (O'Donoghue and Candès, 2015; Kim and Fessler, 2018), to speed up convergence.

Acknowledgements

R.F.B. and E.Y.S. were both supported by the National Institutes of Health via grant NIH R01-023968. R.F.B. was also supported by the National Science Foundation via grant DMS–1654076, and by the Office of Naval Research via grant N00014-20-1-2337. E.Y.S. was also supported by National Institutes of Health via grant NIH R01-026282. The authors thank Michael Bian for helpful feedback.

References

Rina Foygel Barber and Emil Y Sidky. MOCCA: Mirrored convex/concave optimization for nonconvex composite functions. *The Journal of Machine Learning Research*, 17(1): 5006–5056, 2016.

Rina Foygel Barber, Emil Y Sidky, Taly Gilat Schmidt, and Xiaochuan Pan. An algorithm for constrained one-step inversion of spectral CT data. *Physics in Medicine & Biology*, 61(10):3784–3818, 2016.

Alexandre Belloni and Victor Chernozhukov. ℓ_1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130, 2011.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Nonconvex Lagrangian-based optimization: monitoring schemes and global convergence. *Mathematics of Operations Research*, 43(4):1210–1232, 2018.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Emmanuel J Candès, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 14:877–905, 2008.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40 (1):120–145, 2011.

Gong Chen and Marc Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathematical Programming*, 64(1-3):81–101, 1994.

Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B): 3539–3568, 2018.

Maryam Fazel, Haitham Hindi, and Stephen P Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 3, pages 2156–2162. IEEE, 2003.

Suriya Gunasekar, Arindam Banerjee, and Joydeep Ghosh. Unified view of matrix completion under general structural constraints. In *Advances in Neural Information Processing Systems*, pages 1180–1188, 2015.

Ke Guo, Deren Han, David ZW Wang, and Tingting Wu. Convergence of ADMM for multi-block nonconvex separable optimization models. *Frontiers of Mathematics in China*, 12 (5):1139–1162, 2017.

Wooseok Ha, Emil Y Sidky, Rina Foygel Barber, Taly Gilat Schmidt, and Xiaochuan Pan. Alternating minimization based framework for simultaneous spectral calibration and image reconstruction in spectral CT. In *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, pages 1–5. IEEE, 2018.

Bingsheng He and Xiaoming Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.

Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional M-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014.

Bo Jiang, Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. *Computational Optimization and Applications*, 72(1):115–157, 2019.

Donghwan Kim and Jeffrey A Fessler. Adaptive restart of the optimized gradient method for convex optimization. *Journal of Optimization Theory and Applications*, 178(1):240–263, 2018.

Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011.

Alessandro Lanza, Serena Morigi, Ivan Selesnick, and Fiorella Sgallari. Nonconvex nonsmooth optimization via convex–nonconvex majorization–minimization. *Numerische Mathematik*, 136(2):343–381, 2017.

Guoyin Li and Ting Kei Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

Qinghua Liu, Xinyue Shen, and Yuantao Gu. Linearized ADMM for nonconvex nonsmooth optimization with convergence analysis. *IEEE Access*, 7:76131–76144, 2019.

Po-Ling Loh and Martin J Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *The Journal of Machine Learning Research*, 16(1):559–616, 2015.

Sindri Magnússon, Pradeep Chathuranga Weeraddana, Michael G Rabbat, and Carlo Fischione. On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems. *IEEE Transactions on Control of Network Systems*, 3 (3):296–309, 2015.

Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Hung Nien and Jeffrey A Fessler. Fast X-ray CT image reconstruction using a linearized augmented Lagrangian method with ordered subsets. *IEEE transactions on medical imaging*, 34(2):388–399, 2014.

Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8(1):331–372, 2015.

Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15:715–732, 2015.

Thomas Pock and Antonin Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *2011 International Conference on Computer Vision*, pages 1762–1769. IEEE, 2011.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE transactions on information theory*, 57 (10):6976–6994, 2011.

Benjamin M Rizzo, Emil Y Sidky, and Taly Gilat Schmidt. Material decomposition from unregistered dual kV data using the cOSSCIR algorithm. In *7th International Conference on Image Formation in X-Ray Computed Tomography*, volume 12304, pages 539–544. SPIE, 2022.

Benjamin M Rizzo, Emil Y Sidky, and Taly Gilat Schmidt. Experimental dual-kV reconstructions of objects containing metal using the cOSSCIR algorithm. In *Medical Imaging 2023: Physics of Medical Imaging*, volume 12463, pages 907–912. SPIE, 2023.

Taly Gilat Schmidt, Rina Foygel Barber, and Emil Y Sidky. Spectral CT metal artifact reduction using weighted masking and a one step direct inversion reconstruction algorithm. In *Medical Imaging 2020: Physics of Medical Imaging*, volume 11312, page 113121F. International Society for Optics and Photonics, 2020.

Taly Gilat Schmidt, Barbara A Sammut, Rina Foygel Barber, Xiaochuan Pan, and Emil Y Sidky. Addressing ct metal artifacts using photon-counting detectors and one-step spectral CT image reconstruction. *Medical Physics*, 49(5):3021–3040, 2022.

Taly Gilat Schmidt, Emil Y Sidky, Xiaochuan Pan, Rina Foygel Barber, Fredrik Grönberg, Martin Sjölin, and Mats Danielsson. Constrained one-step material decomposition reconstruction of head CT data from a silicon photon-counting prototype. *Medical Physics*, 2023.

Emil Y Sidky, Rina Foygel Barber, Taly Gilat-Schmidt, and Xiaochuan Pan. Three material decomposition for spectral computed tomography enabled by block-diagonal step-preconditioning. *arXiv preprint arXiv:1801.06263*, 2018.

Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. Douglas-Rachford splitting and ADMM for nonconvex optimization: Accelerated and Newton-type algorithms. *arXiv preprint arXiv:2005.10230*, 2020.

Tuomo Valkonen. A primal–dual hybrid gradient method for nonlinear operators with applications to MRI. *Inverse Problems*, 30(5):055012, 2014.

Fenghui Wang, Zongben Xu, and Hong-Kun Xu. Convergence of Bregman alternating direction method with multipliers for nonconvex composite problems. *arXiv preprint arXiv:1410.8625*, 2014.

Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of multi-block Bregman ADMM for nonconvex composite problems. *Science China Information Sciences*, 61(12):122101, 2018.

Huahua Wang and Arindam Banerjee. Bregman alternating direction method of multipliers. In *Advances in Neural Information Processing Systems*, pages 2816–2824, 2014.

Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

Tianqi Zhao, Mladen Kolar, and Han Liu. A general framework for robust testing and confidence regions in high-dimensional quantile regression. *arXiv preprint arXiv:1412.8724*, 2014.

Wenliang Zhong and James Kwok. Fast stochastic alternating direction method of multipliers. In *International conference on machine learning*, pages 46–54. PMLR, 2014.

Appendix A. Additional details and proofs

A.1 A closer look at restricted strong convexity

To better understand this condition in the setting of the composite optimization problem (1) studied in this work, consider the augmented Lagrangian $\mathcal{L}_\Sigma$ defined in (3). Since the x and y update steps of ADMM are performing (approximate) alternating minimization on this augmented Lagrangian, it is intuitive that convexity of the map $(x, y) \mapsto \mathcal{L}_\Sigma(x, y, u)$ (at a fixed u) is generally needed for convergence to be possible.

On the other hand, if $(x, y) \mapsto \mathcal{L}_{\Sigma/2}(x, y, u)$ is strongly convex (note that we have replaced the penalty matrix Σ with a smaller penalty, $\Sigma/2$), this is sufficient to ensure the restricted strong convexity condition (6) holds (with $\varepsilon = 0$) at any feasible point $(\tilde{x}, \tilde{y})$. To see why, for any $\xi_x \in \partial f(x)$ and $\zeta_y \in \partial g(y)$, using the fact that $A\tilde{x} + B\tilde{y} = c$ by feasibility,

an elementary calculation shows that

$$\begin{aligned}\left\langle \begin{pmatrix} x-\tilde{x} \\ y-\tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle + \frac{1}{2}\|Ax+By-c\|_\Sigma^2 \\ = \left\langle \begin{pmatrix} x-\tilde{x} \\ y-\tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle + \frac{1}{2}\|Ax+By-A\tilde{x}-B\tilde{y}\|_\Sigma^2 \\ = \left\langle \begin{pmatrix} x-\tilde{x} \\ y-\tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x + \frac{1}{2}A^\top\Sigma(Ax+By) - \xi_{\tilde{x}} - \frac{1}{2}A^\top\Sigma(A\tilde{x}+B\tilde{y}) \\ \zeta_y + \frac{1}{2}B^\top\Sigma(Ax+By) - \zeta_{\tilde{y}} - \frac{1}{2}B^\top\Sigma(A\tilde{x}+B\tilde{y}) \end{pmatrix} \right\rangle. \quad (24)\end{aligned}$$

We can also calculate

$$\begin{pmatrix} \xi_x + \frac{1}{2}A^\top\Sigma(Ax+By) \\ \zeta_y + \frac{1}{2}B^\top\Sigma(Ax+By) \end{pmatrix} \in \partial_{(x,y)}\mathcal{L}_{\Sigma/2}(x,y,\tilde{u})$$

and similarly

$$\begin{pmatrix} \xi_{\tilde{x}} + \frac{1}{2}A^\top\Sigma(A\tilde{x}+B\tilde{y}) \\ \zeta_{\tilde{y}} + \frac{1}{2}B^\top\Sigma(A\tilde{x}+B\tilde{y}) \end{pmatrix} \in \partial_{(x,y)}\mathcal{L}_{\Sigma/2}(\tilde{x},\tilde{y},\tilde{u}).$$

Therefore, the final expression in (24) will be lower-bounded by strong convexity of $\mathcal{L}_{\Sigma/2}$. Thus, we can interpret the RSC condition (6) as only mildly stronger than requiring strong convexity of the augmented Lagrangian.

A.2 Completing the proof of Theorem 1

To complete the proof of Theorem 1, we only need to prove that the bound (11) holds under the assumption (10) on the step size matrices H_f, H_g, for any point (x,y,u) with $Ax+By=c$.

By definition of x_{t+2} (i.e., since x_{t+2} is a minimizer of the subproblem that defines its update step), we must have

$$\begin{aligned}0 &\in \partial f_c(x_{t+2}) + \nabla f_d(x_{t+1}) + A^\top u_{t+1} + A^\top\Sigma(Ax_{t+2}+By_{t+1}-c) + H_f(x_{t+2}-x_{t+1}) \\ &= \partial f_c(x_{t+2}) + \nabla f_d(x_{t+1}) + A^\top(2u_{t+1}-u_t) + A^\top\Sigma A(x_{t+2}-x_{t+1}) + H_f(x_{t+2}-x_{t+1}),\end{aligned}$$

since $u_{t+1} = u_t + \Sigma(Ax_{t+1}+By_{t+1}-c)$. Since $\partial f(x_{t+2}) = \partial f_c(x_{t+2}) + \nabla f_d(x_{t+2})$, this implies that there exists some $\xi_{x_{t+2}} \in \partial f(x_{t+2})$ such that

$$\xi_{x_{t+2}} = \nabla f_d(x_{t+2}) - \nabla f_d(x_{t+1}) - A^\top(2u_{t+1}-u_t) - A^\top\Sigma A(x_{t+2}-x_{t+1}) - H_f(x_{t+2}-x_{t+1})$$

and therefore

$$\begin{aligned}\langle x_{t+2}-x, \xi_{x_{t+2}} + A^\top u\rangle = \langle x_{t+2}-x, -A^\top(2u_{t+1}-u_t-u) - A^\top\Sigma A(x_{t+2}-x_{t+1})\rangle \\ + \langle x_{t+2}-x, \nabla f_d(x_{t+2}) - \nabla f_d(x_{t+1}) - H_f(x_{t+2}-x_{t+1})\rangle.\end{aligned}$$

We can similarly calculate

$$\begin{aligned}0 &\in \partial g_c(y_{t+1}) + \nabla g_d(y_t) + B^\top u_t + B^\top\Sigma(Ax_{t+1}+By_{t+1}-c) + H_g(y_{t+1}-y_t) \\ &= \partial g_c(y_{t+1}) + \nabla g_d(y_t) + B^\top u_{t+1} + H_g(y_{t+1}-y_t),\end{aligned}$$

and so there exists some $\zeta_{y_{t+1}} \in \partial g(y_{t+1})$ satisfying

$$\langle y_{t+1} - y, \zeta_{y_{t+1}} + B^\top u \rangle = \langle y_{t+1} - y, -B^\top (u_{t+1} - u) \rangle \\ + \langle y_{t+1} - y, \nabla g_d(y_{t+1}) - \nabla g_d(y_t) - H_g(y_{t+1} - y_t) \rangle.$$

We can further calculate

$$\begin{aligned}
&\langle x_{t+2} - x, -A^\top (2u_{t+1} - u_t - u) - A^\top \Sigma A (x_{t+2} - x_{t+1}) \rangle \\
&= \begin{pmatrix} x_{t+2} - x \\ u_{t+1} - u \end{pmatrix}^\top \begin{pmatrix} A^\top \Sigma A & A^\top \\ A & \Sigma^{-1} \end{pmatrix} \begin{pmatrix} x_{t+1} - x_{t+2} \\ u_t - u_{t+1} \end{pmatrix} - \langle \Sigma^{-1}(u_t - u_{t+1}) + A(x_{t+1} - x), u_{t+1} - u \rangle \\
&= \begin{pmatrix} x_{t+2} - x \\ u_{t+1} - u \end{pmatrix}^\top \begin{pmatrix} A^\top \Sigma A & A^\top \\ A & \Sigma^{-1} \end{pmatrix} \begin{pmatrix} x_{t+1} - x_{t+2} \\ u_t - u_{t+1} \end{pmatrix} + \langle B(y_{t+1} - y), u_{t+1} - u \rangle.
\end{aligned}$$

Combining our calculations so far, we have

$$\left\langle \begin{pmatrix} x_{t+2} - x \\ y_{t+1} - y \end{pmatrix}, \begin{pmatrix} \xi_{x_{t+2}} + A^\top u \\ \zeta_{y_{t+1}} + B^\top u \end{pmatrix} \right\rangle = (z_{t+1} - z)^\top M (z_t - z_{t+1}) \\ + \langle x_{t+2} - x, \nabla f_d(x_{t+2}) - \nabla f_d(x_{t+1}) \rangle + \langle y_{t+1} - y, \nabla g_d(y_{t+1}) - \nabla g_d(y_t) \rangle, \quad (25)$$

where we define $z = (x, y, u)$ and $z_t = (x_{t+1}, y_t, u_t)$ for each t, and let

$$M = \begin{pmatrix} H_f + A^\top \Sigma A & 0 & A^\top \\ 0 & H_g & 0 \\ A & 0 & \Sigma^{-1} \end{pmatrix} \succeq 0.$$

Next, defining $\|v\|_M = \sqrt{v^\top M v}$, we can use a telescoping sum to calculate

$$\begin{aligned}
\sum_{t=0}^{T-1} (z_{t+1} - z)^\top M (z_t - z_{t+1}) &= \sum_{t=0}^{T-1} \left(\frac{1}{2}\|z - z_t\|_M^2 - \frac{1}{2}\|z - z_{t+1}\|_M^2 - \frac{1}{2}\|z_t - z_{t+1}\|_M^2 \right) \\
&= \frac{1}{2}\|z - z_0\|_M^2 - \frac{1}{2}\|z - z_T\|_M^2 - \frac{1}{2}\sum_{t=0}^{T-1} \|z_t - z_{t+1}\|_M^2.
\end{aligned}$$

Furthermore,

$$\begin{aligned}
&\|z_t - z_{t+1}\|_M^2 - \|x_{t+1} - x_{t+2}\|_{H_f}^2 - \|y_t - y_{t+1}\|_{H_g}^2 \\
&\qquad = \begin{pmatrix} x_{t+1} - x_{t+2} \\ u_t - u_{t+1} \end{pmatrix}^\top \begin{pmatrix} A^\top \Sigma A & A^\top \\ A & \Sigma^{-1} \end{pmatrix} \begin{pmatrix} x_{t+1} - x_{t+2} \\ u_t - u_{t+1} \end{pmatrix} \\
&\qquad = \|\Sigma^{-1}(u_t - u_{t+1}) + A(x_{t+1} - x_{t+2})\|_\Sigma^2 = \|Ax_{t+2} + By_{t+1} - c\|_\Sigma^2,
\end{aligned}$$

where the last step plugs in the update step for u_{t+1}. Combining these calculations with (25), we obtain

$$\begin{aligned}
&\sum_{t=0}^{T-1} \left\langle \begin{pmatrix} x_{t+2} - x \\ y_{t+1} - y \end{pmatrix}, \begin{pmatrix} \xi_{x_{t+2}} + A^\top u \\ \zeta_{y_{t+1}} + B^\top u \end{pmatrix} \right\rangle \\
&\qquad = \frac{1}{2}\|z - z_0\|_M^2 - \frac{1}{2}\|z - z_T\|_M^2 - \frac{1}{2}\sum_{t=0}^{T-1}\|Ax_{t+2} + By_{t+1} - c\|_\Sigma^2 \\
&\qquad + \sum_{t=0}^{T-1}\left[\langle x_{t+2} - x, \nabla f_d(x_{t+2}) - \nabla f_d(x_{t+1})\rangle - \frac{1}{2}\|x_{t+1} - x_{t+2}\|_{H_f}^2\right] \\
&\qquad\qquad + \sum_{t=0}^{T-1}\left[\langle y_{t+1} - y, \nabla g_d(y_{t+1}) - \nabla g_d(y_t)\rangle - \frac{1}{2}\|y_t - y_{t+1}\|_{H_g}^2\right]. \quad (26)
\end{aligned}$$

Now, since $H_f \succeq \nabla^2 f_d(x)$ by the assumption (10), we can write

$$f_d(x_{t+2}) \le f_d(x_{t+1}) + \langle x_{t+2} - x_{t+1}, \nabla f_d(x_{t+1})\rangle + \frac{1}{2}\|x_{t+1} - x_{t+2}\|_{H_f}^2$$

for each t. Rearranging terms and taking a telescoping sum, this means that

$$\begin{aligned}
&\sum_{t=0}^{T-1}\left[\langle x_{t+2} - x, \nabla f_d(x_{t+2}) - \nabla f_d(x_{t+1})\rangle - \frac{1}{2}\|x_{t+1} - x_{t+2}\|_{H_f}^2\right] \\
&\qquad\qquad \le f_d(x_1) - f_d(x_{T+1}) + \langle x - x_1, \nabla f_d(x_1)\rangle - \langle x - x_{T+1}, \nabla f_d(x_{T+1})\rangle.
\end{aligned}$$

Again applying $H_f \succeq \nabla^2 f_d(x)$, we also have

$$f_d(x) \le f_d(x_{T+1}) + \langle x - x_{T+1}, \nabla f_d(x_{T+1})\rangle + \frac{1}{2}\|x - x_{T+1}\|_{H_f}^2$$

and

$$f_d(x_1) \le f_d(x) + \langle x_1 - x, \nabla f_d(x)\rangle + \frac{1}{2}\|x - x_1\|_{H_f}^2,$$

which combined with the above yields

$$\begin{aligned}
&\sum_{t=0}^{T-1}\left[\langle x_{t+2} - x, \nabla f_d(x_{t+2}) - \nabla f_d(x_{t+1})\rangle - \frac{1}{2}\|x_{t+1} - x_{t+2}\|_{H_f}^2\right] \\
&\qquad\qquad \le -\langle x - x_1, \nabla f_d(x) - \nabla f_d(x_1)\rangle + \frac{1}{2}\|x - x_{T+1}\|_{H_f}^2 + \frac{1}{2}\|x - x_1\|_{H_f}^2.
\end{aligned}$$

Performing an identical calculation for the y terms, and combining these calculations with (26) along with the fact that $\|z - z_T\|_M^2 \ge \|x - x_{T+1}\|_{H_f}^2 + \|y - y_T\|_{H_g}^2$, we obtain

$$\begin{aligned}
&\sum_{t=0}^{T-1} \left\langle \begin{pmatrix} x_{t+2} - x \\ y_{t+1} - y \end{pmatrix}, \begin{pmatrix} \xi_{x_{t+2}} + A^\top u \\ \zeta_{y_{t+1}} + B^\top u \end{pmatrix} \right\rangle \\
&\qquad\qquad \le C_1(x, y, u; x_0, y_0, u_0) - \frac{1}{2}\sum_{t=0}^{T-1}\|Ax_{t+2} + By_{t+1} - c\|_\Sigma^2,
\end{aligned}$$

where we define

$$C_1(x,y,u;x_0,y_0,u_0) = \frac{1}{2}\|z-z_0\|_M^2 - \langle x-x_1, \nabla f_d(x) - \nabla f_d(x_1)\rangle + \frac{1}{2}\|x-x_1\|_{H_f}^2 \\ - \langle y-y_0, \nabla g_d(y) - \nabla g_d(y_0)\rangle + \frac{1}{2}\|y-y_0\|_{H_g}^2. \quad (27)$$

(Note that x_1 is a deterministic function of (x_0,y_0,u_0), and therefore C_1 can depend implicitly on x_1.) This proves the desired bound (11).

A.3 Details for implementing ADMM for the CT application

To run Algorithm 1 for the CT image reconstruction problem (21), plugging in our choices of parameters H_f, H_g, Σ and the values of A, B, c and $f(x) \equiv 0$, our update steps can be calculated as follows. Note that in our notation below, the x, y, u variables are all treated as matrices, with $n_k \times n_m$ dimensional x variables and with $n_\ell \times n_m$ dimensional y and u variables.

- The x update step is given by

$$x_{t+1} = x_t + Q_f^{-1} P^\top \big(\tilde{\Sigma}(y_t - Px_t) - u_t\big).$$

  Since Q_f and $\tilde{\Sigma}$ are diagonal while P is sparse, this requires only inexpensive matrix-vector calculations.

- The y update step is given by solving the minimization problem

$$y_{t+1} = \arg\min_y \left\{ g_c(y) + \langle y, \nabla g_d(y_t) - (u_t + \tilde{\Sigma} P x_{t+1})\rangle + \frac{1}{2}\operatorname{vec}(y)^\top (\tilde{\Sigma} \otimes \mathbf{I}_{n_m}) \operatorname{vec}(y) \right\}.$$

  We recall from the definition of g_c (22) that this function separates over the n_ℓ many rays—that is, we can write $g_c(y) = \sum_\ell g_{c,\ell}(y_\ell)$, where $y_\ell \in \mathbb{R}^{n_m}$ is the portion of y corresponding to the ℓ-th ray, and where

$$g_{c,\ell}(y_\ell) = \sum_w \sum_i S_{w\ell i}\, \mathrm{qexp}\left\{ -\sum_m \mu_{mi} y_{\ell m} \right\}.$$

  Therefore, equivalently, the y update step is given by solving

$$(y_{t+1})_\ell = \arg\min_{y_\ell \in \mathbb{R}^{n_m}} \left\{ g_{c,\ell}(y) + \langle y_\ell, (\nabla g_d(y_t))_\ell - (u_t)_\ell - \tilde{\Sigma}_{\ell\ell}(Px_{t+1})_\ell\rangle + \frac{\tilde{\Sigma}_{\ell\ell}}{2}\|y_\ell\|_2^2 \right\}$$

  for each $\ell = 1, \ldots, n_\ell$. Since we typically work with a small number of materials (e.g., $n_m = 3$ or $n_m = 5$), solving each one of these convex minimization problems is computationally very inexpensive. We will use the Newton–Raphson method to solve the minimization subproblem approximately, in parallel for each ℓ: setting $y_{t+1}^{(0)} = y_t$, we define

$$(y_{t+1}^{(i+1)})_\ell = (y_{t+1}^{(i)})_\ell - (\nabla^2 g_{c,\ell}((y_{t+1}^{(i)})_\ell) + \tilde{\Sigma}_{\ell\ell}\mathbf{I}_{n_m})^{-1} \cdot \\ \Big(\nabla g_{c,\ell}((y_{t+1}^{(i)})_\ell) + (\nabla g_d(y_t))_\ell + \tilde{\Sigma}_{\ell\ell}(y_{t+1}^{(i)} - Px_{t+1})_\ell - (u_t)_\ell\Big),$$

for each $i = 0, 1, 2, \ldots, N-1$, and then set $y_{t+1} = y_{t+1}^{(N)}$. In our implementation, at each iteration t we run $N = 10$ steps of the Newton–Raphson method to compute the y update, which is sufficient to obtain a near-exact solution.

- The u update step is given by

$$u_{t+1} = u_t + \tilde{\Sigma}(Px_{t+1} - y_{t+1}).$$

Since $\tilde{\Sigma}$ is diagonal while P is sparse, this again requires only inexpensive matrix-vector calculations.

A.4 Details for implementing ADMM for the sparse quantile regression example

We now compute the steps of Algorithm 1 for the sparse quantile regression example, i.e., for the problem of minimizing (14). Plugging in our choices of the parameters H_f, H_g, Σ and of A, B, c, the steps of Algorithm 1 are given by

$$\begin{aligned}
x_{t+1} &= \operatorname*{arg\,min}_{x\in\mathbb{R}^d} \left\{ f_c(x) + \langle x, \nabla f_d(x_t) + \Phi^\top u_t\rangle + \frac{\sigma}{2}\|\Phi x - y_t\|_2^2 + \frac{\sigma}{2}\|x - x_t\|^2_{\gamma\mathbf{I}_d - \Phi^\top\Phi} \right\}, \\
y_{t+1} &= \operatorname*{arg\,min}_{y\in\mathbb{R}^n} \left\{ g_c(y) + \langle y, \nabla g_d(y_t) - u_t\rangle + \frac{\sigma}{2}\|\Phi x_{t+1} - y\|_2^2 \right\}, \\
u_{t+1} &= u_t + \sigma(\Phi x_{t+1} - y_{t+1}).
\end{aligned}$$

Now we compute the x and y update steps explicitly. First, for x, recall that $f_c(x) = \lambda\|x\|_1 + \delta_{\|x\|_2\le R}$ and

$$f_d(x) = \lambda \sum_{j=1}^{d} \left(\beta \log(1 + |x_j|/\beta) - |x_j|\right).$$

We can calculate the gradient as

$$\left[\nabla f_d(x)\right]_j = -\frac{\lambda x_j}{\beta + |x_j|}.$$

Therefore,

$$\begin{aligned}
&f_c(x) + \langle x, \nabla f_d(x_t) + \Phi^\top u_t\rangle + \frac{\sigma}{2}\|\Phi x - y_t\|_2^2 + \frac{\sigma}{2}\|x - x_t\|^2_{\gamma\mathbf{I}_d - \Phi^\top\Phi} \\
&= \sum_{j=1}^{d} \left(\frac{\sigma\gamma}{2} x_j^2 - x_j \cdot \left[\frac{\lambda (x_t)_j}{\beta + |(x_t)_j|} + \sigma(\gamma x_t - \Phi^\top(\Phi x_t - y_t + u_t/\sigma))_j \right] \right) + \lambda\|x\|_1 + \delta_{\|x\|_2\le R} \\
&= \frac{\sigma\gamma}{2} \sum_{j=1}^{d} \left(x_j^2 - 2x_j \cdot (\tilde{x}_{t+1})_j \right) + \lambda\|x\|_1 + \delta_{\|x\|_2\le R},
\end{aligned}$$

where we define a vector $\tilde{x}_{t+1}$ with entries

$$(\tilde{x}_{t+1})_j = (x_t)_j - \frac{\left(\Phi^\top(\Phi x_t - y_t + u_t/\sigma)\right)_j}{\gamma} + \frac{\lambda}{\sigma\gamma} \cdot \frac{(x_t)_j}{\beta + |(x_t)_j|}.$$

Then we can verify that the objective function above is minimized by defining

$$x_{t+1} = \text{SoftThresh}_{\frac{\lambda}{\sigma\gamma}}(\tilde{x}_{t+1}) \cdot \min\left\{1, \frac{R}{\|\text{SoftThresh}_{\frac{\lambda}{\sigma\gamma}}(\tilde{x}_{t+1})\|_2}\right\},$$

where the soft thresholding function, $\text{SoftThresh}_\lambda : \mathbb{R}^d \to \mathbb{R}^d$, is defined elementwise as

$$\left[\text{SoftThresh}_\lambda(x)\right]_j = \begin{cases} x_j - \lambda, & \text{if } x_j > \lambda, \\ 0, & \text{if } |x_j| \le \lambda, \\ x_j + \lambda, & \text{if } x_j < -\lambda. \end{cases}$$

Next, for the y update step, recall $g_d(y) \equiv 0$ and

$$g_c(y) = \frac{1}{n}\sum_{i=1}^{n} q\max\{w_i - y_i, 0\} + (1-q)\max\{y_i - w_i, 0\}.$$

Then the optimization problem for the y update step separates over the n entries of y:

$$\begin{aligned} &g_c(y) + \langle y, \nabla g_d(y_t) - u_t\rangle + \frac{\sigma}{2}\|\Phi x_{t+1} - y\|_2^2 \\ &= \sum_{i=1}^{n}\left(\frac{1}{n}\left[q\max\{w_i - y_i, 0\} + (1-q)\max\{y_i - w_i, 0\}\right] + \frac{\sigma}{2}y_i^2 - y_i\cdot(\sigma(\Phi x_{t+1})_i + (u_t)_i)\right). \end{aligned}$$

This is minimized by setting y_{t+1} to have entries

$$(y_{t+1})_i = \begin{cases} (\Phi x_{t+1})_i + \frac{(u_t)_i}{\sigma} + \frac{q}{n\sigma}, & \text{if } (\Phi x_{t+1})_i + \frac{(u_t)_i}{\sigma} + \frac{q}{n\sigma} < w_i, \\ (\Phi x_{t+1})_i + \frac{(u_t)_i}{\sigma} - \frac{1-q}{n\sigma}, & \text{if } (\Phi x_{t+1})_i + \frac{(u_t)_i}{\sigma} - \frac{1-q}{n\sigma} > w_i, \\ w_i, & \text{if } (\Phi x_{t+1})_i + \frac{(u_t)_i}{\sigma} - \frac{1-q}{n\sigma} \le w_i \le (\Phi x_{t+1})_i + \frac{(u_t)_i}{\sigma} + \frac{q}{n\sigma}. \end{cases}$$

A.5 Proof of Proposition 2 (verifying assumptions for the sparse quantile regression example)

To prove the result, we need to check that, with probability at least $1 - (nd)^{-1}$, the RSC bound (6) and the approximate first-order optimality condition (9) both at the point $(\tilde{x}, \tilde{y}, \tilde{u})$, with parameters defined as in the statement of the proposition. Concretely, let $\tilde{u} \in \mathbb{R}^n$ have entries

$$\tilde{u}_i = \frac{1}{n}(-q \cdot \mathbb{1}\{z_i > 0\} + (1-q)\cdot \mathbb{1}\{z_i < 0\}).$$

Then we can verify $\zeta_{\tilde{y}} = \tilde{u} \in \partial g(\tilde{y})$. Define also $\xi_{\tilde{x}}$ to have entries

$$(\xi_{\tilde{x}})_j = \begin{cases} \frac{\lambda\beta\,\text{sign}(\tilde{x}_j)}{\beta + |\tilde{x}_j|}, & \tilde{x}_j \neq 0, \\ (-\Phi^\top\tilde{u})_j, & \tilde{x}_j = 0. \end{cases}$$

We will show that, with the desired probability,

$$\left\langle \begin{pmatrix} x-\tilde{x} \\ y-\tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq C_3 \min\left\{\|x-\tilde{x}\|_2^2, \|x-\tilde{x}\|_2\right\} - \frac{\sigma}{2}\|y-\Phi x\|_2^2 - C_4 \max\{1,\sigma^{-1}\} \cdot \frac{s_* \log(nd)}{n} \quad (28)$$

for all $x \in \mathrm{dom}(f)$ $y \in \mathrm{dom}(g)$, $\xi_x \in \partial f(x)$, $\zeta_y \in \partial g(y)$, and that

$$\xi_{\tilde{x}} \in \partial f(\tilde{x}) \text{ and } \|-\Phi^\top \tilde{u} - \xi_{\tilde{x}}\|_2 \leq \min\left\{\frac{C_3}{2}, \sqrt{C_3 C_5}\sqrt{\frac{s_* \log(nd)}{n}}\right\}, \quad (29)$$

where $C_3, C_4, C_5 > 0$ are constants that depend only on $c_z, t_z, a_\phi, b_\phi, B_\phi$ and on C_λ. These bounds are sufficient to verify Assumptions 1 and 2, as desired.

A.5.1 Verifying approximate first-order optimality

First we check that $\xi_{\tilde{x}} \in \partial f(\tilde{x})$. Recall that we can write $f(x) = f_c(x) + f_d(x)$ where, for any $x \in \mathrm{dom}(f)$ (i.e., $\|x\|_2 \leq R$), we have

$$f(x) = \lambda \sum_{j=1}^{d} \beta \log(1 + |x_j|/\beta).$$

Now fix any $x \in \mathrm{dom}(f)$. Then we can calculate that a subgradient $\xi_x \in \partial f(x)$ must have entries satisfying

$$\begin{cases} (\xi_x)_j = \frac{\lambda\beta\,\mathrm{sign}(x_j)}{\beta+|x_j|}, & x_j \neq 0, \\ (\xi_x)_j \in [-\lambda, \lambda], & x_j = 0. \end{cases} \quad (30)$$

From this calculation, we can see that to verify $\xi_{\tilde{x}} \in \partial f(\tilde{x})$, we only need to check that $|(\xi_{\tilde{x}})_j| \leq \lambda$ for all j with $\tilde{x}_j = 0$. Since $\|\Phi\|_\infty \leq B_\phi$ with probability 1, while $\tilde{u}$ is a $\frac{1}{n}$-bounded zero-mean vector, Hoeffding's inequality shows that

$$\mathbb{P}\left\{\|\Phi^\top \tilde{u}\|_\infty \leq 2B_\phi\sqrt{\frac{\log(nd)}{n}}\right\} \geq 1 - (2nd)^{-1}. \quad (31)$$

From this point on, we will assume that this event holds. Since $\lambda = C_\lambda\sqrt{\frac{\log(nd)}{n}} \geq C_1\sqrt{\frac{\log(nd)}{n}} \geq 2B_\phi\sqrt{\frac{\log(nd)}{n}}$ (as long as we take $C_1 \geq 2B_\phi$, as we will do below), this verifies that $|(\xi_{\tilde{x}})_j| \leq \lambda$ for j such that $\tilde{x}_j = 0$, and thus $\xi_{\tilde{x}} \in \partial f(\tilde{x})$, as desired.

Next we check that (29) holds, to complete our verification of the approximate first-order optimality assumption. Writing $S_* \subseteq \{1, \ldots, d\}$ to denote the support of $\tilde{x}$, we have

$$\|(\xi_{\tilde{x}})_{S_*}\|_2 = \sqrt{\sum_{j:\tilde{x}_j \neq 0} \left(\frac{\lambda\beta\,\mathrm{sign}(\tilde{x}_j)}{\beta+|\tilde{x}_j|}\right)^2} \leq \sqrt{\sum_{j:\tilde{x}_j \neq 0} \lambda^2} \leq \sqrt{s_*}\lambda,$$

and also,

$$\|(\Phi^\top \tilde{u})_{S_*}\|_2 \leq \sqrt{s_*}\|(\Phi^\top \tilde{u})_{S_*}\|_\infty \leq \sqrt{s_*} \cdot 2B_\phi\sqrt{\frac{\log(nd)}{n}}.$$

Then

$$\|-\Phi^\top \tilde{u} - \xi_{\tilde{x}}\|_2 \leq \|(-\Phi^\top \tilde{u} - \xi_{\tilde{x}})_{S_*}\|_2 + \|(-\Phi^\top \tilde{u} - \xi_{\tilde{x}})_{S_*^c}\|_2 \\ \leq \|(\Phi^\top \tilde{u})_{S_*}\|_2 + \|(\xi_{\tilde{x}})_{S_*}\|_2 + \|(-\Phi^\top \tilde{u} - \xi_{\tilde{x}})_{S_*^c}\|_2.$$

Since $\|(-\Phi^\top \tilde{u} - \xi_{\tilde{x}})_{S_*^c}\|_2 = 0$ by definition, this establishes that

$$\|-\Phi^\top \tilde{u} - \xi_{\tilde{x}}\|_2 \leq \sqrt{s_*}\left(\lambda + 2B_\phi\sqrt{\frac{\log(nd)}{n}}\right) = (C_\lambda + 2B_\phi)\sqrt{\frac{s_* \log(nd)}{n}}.$$

Recall $\frac{s_* \log(nd)}{n} \leq C_0$ by assumption, and furthermore,

$$\lambda\sqrt{s_*} = C_\lambda\sqrt{\frac{\log(nd)}{n}} \cdot s_*^{1/2} \leq C_1\sqrt{\frac{C_0 \frac{n}{\log(nd)}}{s_*}}\sqrt{\frac{\log(nd)}{n}} \cdot s_*^{1/2} = C_1\sqrt{C_0}. \tag{32}$$

We therefore have

$$\|-\Phi^\top \tilde{u} - \xi_{\tilde{x}}\|_2 \leq \min\left\{\sqrt{C_0}(C_1 + 2B_\phi), (C_\lambda + 2B_\phi)\sqrt{\frac{s_* \log(nd)}{n}}\right\}.$$

Finally, choosing the constants as $C_3 = 2\sqrt{C_0}(C_1 + 2B_\phi)$ and $C_5 = (C_\lambda + 2B_\phi)^2/C_3$, we have proved (29).

A.5.2 Verifying restricted strong convexity

Next we will verify that (28) holds, to validate the restricted strong convexity property.

Bounding the x term Recall our earlier calculation (30) of the subgradient $\partial f(x)$. Writing $S_* \subseteq \{1, \ldots, d\}$ to denote the support of $\tilde{x}$ as before, for each $j \in S_*^c$ we have

$$(x - \tilde{x})_j \cdot (\xi_x)_j = x_j \cdot (\xi_x)_j = \frac{\lambda\beta|x_j|}{\beta + |x_j|} \geq \lambda|x_j| - \lambda\beta^{-1}x_j^2$$

if $x_j \neq 0$, or if $x_j = 0$ then $(x - \tilde{x})_j \cdot (\xi_x)_j = 0 = \lambda|x_j| - \lambda\beta^{-1}x_j^2$ holds trivially. Thus

$$\langle (x - \tilde{x})_{S_*^c}, (\xi_x)_{S_*^c}\rangle \geq \lambda\|x_{S_*^c}\|_1 - \lambda\beta^{-1}\|x_{S_*^c}\|_2^2 = \lambda\|(x - \tilde{x})_{S_*^c}\|_1 - \lambda\beta^{-1}\|(x - \tilde{x})_{S_*^c}\|_2^2.$$

Next, since $(\xi_{\tilde{x}})_{S_*^c} = (-\Phi^\top \tilde{u})_{S_*^c}$ and we know that $\|\Phi^\top \tilde{u}\|_\infty \leq 2B_\phi\sqrt{\frac{\log(nd)}{n}}$ by (31), we have

$$\langle (x - \tilde{x})_{S_*^c}, (\xi_x - \xi_{\tilde{x}})_{S_*^c}\rangle \geq \left(\lambda - 2B_\phi\sqrt{\frac{\log(nd)}{n}}\right)\|x_{S_*^c}\|_1 - \lambda\beta^{-1}\|x_{S_*^c}\|_2^2.$$

Next, the function $t \mapsto \beta\log(1 + |t|/\beta)$ can be decomposed as

$$\beta\log(1 + |t|/\beta) = |t| + (\beta\log(1 + |t|/\beta) - |t|),$$

where the first term is convex while the second term is concave and twice differentiable with second derivative $\geq -\beta^{-1}$, which proves that

$$\langle (x-\tilde{x})_{S_*}, (\xi_x - \xi_{\tilde{x}})_{S_*} \rangle \geq -\lambda\beta^{-1}\|(x-\tilde{x})_{S_*}\|_2^2.$$

Putting all our calculations together, we have established that

$$\begin{aligned}
\langle x-\tilde{x}, \xi_x - \xi_{\tilde{x}} \rangle &\geq \left(\lambda - 2B_\phi\sqrt{\frac{\log(nd)}{n}}\right)\|x_{S_*^c}\|_1 - \lambda\beta^{-1}\|x-\tilde{x}\|_2^2 \\
&\geq \left(\lambda - 2B_\phi\sqrt{\frac{\log(nd)}{n}}\right)\|x-\tilde{x}\|_1 - \lambda\beta^{-1}\|x-\tilde{x}\|_2^2 - \lambda\|(x-\tilde{x})_{S_*}\|_1 \\
&\geq (C_1 - 2B_\phi)\sqrt{\frac{\log(nd)}{n}}\|x-\tilde{x}\|_1 - \lambda\beta^{-1}\|x-\tilde{x}\|_2^2 - \lambda s_*^{1/2}\|x-\tilde{x}\|_2,
\end{aligned}$$

where the last step holds since $|S_*| \leq s_*$, and by definition of λ. Finally, if $\|x-\tilde{x}\|_2 \leq 1$, then we have

$$\begin{aligned}
\lambda\beta^{-1}\|x-\tilde{x}\|_2^2 + \lambda s_*^{1/2}\|x-\tilde{x}\|_2 &\leq C_2^{-1}\|x-\tilde{x}\|_2^2 + C_\lambda\sqrt{\frac{s_*\log(nd)}{n}}\|x-\tilde{x}\|_2 \\
&\leq 2C_2^{-1}\|x-\tilde{x}\|_2^2 + \frac{C_2 C_\lambda^2}{4}\cdot\frac{s_*\log(nd)}{n},
\end{aligned}$$

by our bound on β along with the fact that $ab \leq ca^2/2 + b^2/2c$ for all $a, b, c > 0$. If instead $\|x-\tilde{x}\|_2 > 1$, then $\|x-\tilde{x}\|_2 \leq 2R$ since $x, \tilde{x} \in \mathrm{dom}(f)$, and so

$$\begin{aligned}
\lambda\beta^{-1}\|x-\tilde{x}\|_2^2 + \lambda s_*^{1/2}\|x-\tilde{x}\|_2 &\leq \left(2\lambda\beta^{-1}R + \lambda s_*^{1/2}\right)\|x-\tilde{x}\|_2 \\
&\leq \left(2C_2^{-1} + C_1\sqrt{C_0}\right)\|x-\tilde{x}\|_2
\end{aligned}$$

since $2\lambda\beta^{-1}R \leq 2C_2^{-1}$ by our bound on β, and since $\lambda s_*^{1/2} \leq C_1\sqrt{C_0}$ as calculated in (32) above. Therefore, combining everything,

$$\begin{aligned}
\langle x-\tilde{x}, \xi_x - \xi_{\tilde{x}} \rangle &\geq (C_1 - 2B_\phi)\sqrt{\frac{\log(nd)}{n}}\cdot\|x-\tilde{x}\|_1 \\
&\quad - \left(2C_2^{-1} + C_1\sqrt{C_0}\right)\cdot\min\{\|x-\tilde{x}\|_2^2, \|x-\tilde{x}\|_2\} - \frac{C_2 C_\lambda^2}{4}\cdot\frac{s_*\log(nd)}{n}.
\end{aligned}$$

Bounding the y term First, we compute the subgradient of $t \mapsto \ell_q(s-t)$:

$$\partial_t \ell_q(s-t) = \begin{cases} \{-q\}, & t < s, \\ [-q, 1-q], & t = s, \\ \{1-q\}, & t > s. \end{cases}$$

Therefore any $\zeta_y \in \partial g(y)$ must have entries satisfying

$$\begin{cases} n(\zeta_y)_i = -q, & y_i < \tilde{y}_i + z_i, \\ n(\zeta_y)_i \in [-q, 1-q], & y_i = \tilde{y}_i + z_i, \\ n(\zeta_y)_i = 1-q, & y_i > \tilde{y}_i + z_i. \end{cases}$$

By definition of $\zeta_{\tilde{y}}$ from above, we can therefore calculate

$$\begin{cases} n\big((\zeta_y)_i - (\zeta_{\tilde{y}})_i\big) = 0, & \text{if } z_i > 0 \text{ and } z_i > y_i - \tilde{y}_i, \text{ or } z_i < 0 \text{ and } z_i < y_i - \tilde{y}_i, \\ n\big((\zeta_y)_i - (\zeta_{\tilde{y}})_i\big) \in [0,1], & \text{if } z_i > 0 \text{ and } z_i = y_i - \tilde{y}_i, \\ n\big((\zeta_y)_i - (\zeta_{\tilde{y}})_i\big) \in [-1,0], & \text{if } z_i < 0 \text{ and } z_i = y_i - \tilde{y}_i, \\ n\big((\zeta_y)_i - (\zeta_{\tilde{y}})_i\big) = -1, & \text{if } z_i < 0 \text{ and } z_i > y_i - \tilde{y}_i, \\ n\big((\zeta_y)_i - (\zeta_{\tilde{y}})_i\big) = 1, & \text{if } z_i > 0 \text{ and } z_i < y_i - \tilde{y}_i. \end{cases}$$

(Note that $z_i \neq 0$ almost surely, so we can ignore the case $z_i = 0$.) We can therefore calculate

$$\langle y - \tilde{y}, \zeta_y - \zeta_{\tilde{y}} \rangle \geq \frac{1}{n} \cdot \overbrace{\sum_i (y_i - \tilde{y}_i) \cdot \mathbb{1}\left\{y_i - \tilde{y}_i > z_i > 0\right\}}^{\text{Term 1}} + \frac{1}{n} \cdot \underbrace{\sum_i (\tilde{y}_i - y_i) \cdot \mathbb{1}\left\{\tilde{y}_i - y_i > -z_i > 0\right\}}_{\text{Term 2}}.$$

Writing $(t)_+ = \max\{t, 0\}$ for any $t \in \mathbb{R}$, we then have

$$\begin{aligned} \text{Term 1} &= \sum_i (y_i - \tilde{y}_i) \cdot \mathbb{1}\left\{y_i - \tilde{y}_i > z_i > 0\right\} \\ &\geq \sum_i (y_i - \tilde{y}_i - z_i)_+ \cdot \mathbb{1}\left\{z_i > 0\right\} \\ &\geq \sum_i (\phi_i^\top (x - \tilde{x}) - 2/(\sigma n) - z_i)_+ \cdot \mathbb{1}\left\{z_i > 0\right\} \\ &\qquad - \sum_i |y_i - \phi_i^\top x| \cdot \mathbb{1}\left\{z_i > 0\right\} \cdot \mathbb{1}\left\{|y_i - \phi_i^\top x| > 2/(\sigma n)\right\} \\ &\geq \sum_i (\phi_i^\top (x - \tilde{x}) - z_i)_+ \cdot \mathbb{1}\left\{z_i > 0\right\} - \frac{2}{\sigma n} \sum_i \mathbb{1}\left\{z_i > 0\right\} - \frac{\sigma n}{2} \sum_i (y_i - \phi_i^\top x)^2 \cdot \mathbb{1}\left\{z_i > 0\right\} \end{aligned}$$

and similarly,

$$\text{Term 2} \geq \sum_i (-\phi_i^\top (x - \tilde{x}) + z_i)_+ \cdot \mathbb{1}\left\{z_i < 0\right\} - \frac{2}{\sigma n} \sum_i \mathbb{1}\left\{z_i < 0\right\} - \frac{\sigma n}{2} \sum_i (y_i - \phi_i^\top x)^2 \cdot \mathbb{1}\left\{z_i < 0\right\}.$$

Therefore, defining

$$H(x) = \frac{1}{n} \sum_i \Big[(\phi_i^\top x - z_i)_+ \cdot \mathbb{1}\left\{z_i > 0\right\} + (-\phi_i^\top x + z_i)_+ \cdot \mathbb{1}\left\{z_i < 0\right\} \Big], \tag{33}$$

and simplifying, we have

$$\langle y - \tilde{y}, \zeta_y - \zeta_{\tilde{y}} \rangle \geq H(x - \tilde{x}) - \frac{\sigma}{2} \|y - \Phi x\|_2^2 - \frac{2}{\sigma n}.$$

We will now use the following lemma (proved in Appendix A.5.3):

Lemma 3 *Suppose that $n \geq 4$, and that $\phi_1, \ldots, \phi_n \in \mathbb{R}^d$ and $z_1, \ldots, z_n \in \mathbb{R}$ satisfy the assumptions of Proposition 2. Then, with probability at least $1 - (2nd)^{-1}$,*

$$H(x) \geq C_1^* \cdot \min\{\|x\|_2^2, \|x\|_2\} - C_2^* \cdot \sqrt{\frac{\log(nd)}{n}} \cdot \|x\|_1 - C_3^* \cdot \frac{\sqrt{\log(nd)}}{n} \text{ for all } x \in \mathbb{R}^d,$$

where $H(x)$ is defined as in (33), and where C_1^*, C_2^*, C_3^* are positive and finite, and depend only on the constants $a_\phi, b_\phi, B_\phi, c_z, t_z$ appearing in the assumptions of Proposition 2.

Returning to our work above we therefore see that, with probability at least $1 - (2nd)^{-1}$,

$$\begin{aligned}\langle y - \tilde{y}, \zeta_y - \zeta_{\tilde{y}} \rangle \geq\ & C_1^* \min\{\|x - \tilde{x}\|_2^2, \|x - \tilde{x}\|_2\} \\ & - C_2^* \sqrt{\frac{\log(nd)}{n}} \cdot \|x - \tilde{x}\|_1 - \frac{\sigma}{2}\|y - \Phi x\|_2^2 - \left(\frac{C_3^* \sqrt{\log(nd)}}{n} + \frac{2}{\sigma n}\right)\end{aligned}$$

for all $x \in \mathrm{dom}(f)$, $y \in \mathbb{R}^n$, and $\zeta_y \in \partial g(y)$.

Combining the x and y terms Combining our bounds for the x and y terms, we have shown that, with probability at least $1 - (nd)^{-1}$, for all $x \in \mathrm{dom}(f)$, $y \in \mathbb{R}^n$, $\xi_x \in \partial f(x)$, and $\zeta_y \in \partial g(y)$,

$$\begin{aligned}\left\langle \begin{pmatrix} x - \tilde{x} \\ y - \tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq & \Bigg[(C_1 - 2B_\phi) \sqrt{\frac{\log(nd)}{n}} \|x - \tilde{x}\|_1 \\ & - \left(2C_2^{-1} + C_1\sqrt{C_0}\right) \cdot \min\{\|x - \tilde{x}\|_2^2, \|x - \tilde{x}\|_2\} - \frac{C_2 C_\lambda^2}{4} \cdot \frac{s_* \log(nd)}{n} \Bigg] \\ & + \Bigg[C_1^* \min\{\|x - \tilde{x}\|_2^2, \|x - \tilde{x}\|_2\} \\ & - C_2^* \sqrt{\frac{\log(nd)}{n}} \cdot \|x - \tilde{x}\|_1 - \frac{\sigma}{2}\|y - \Phi x\|_2^2 - \left(\frac{C_3^* \sqrt{\log(nd)}}{n} + \frac{2}{\sigma n}\right) \Bigg].\end{aligned}$$

We can simplify this to

$$\begin{aligned}\left\langle \begin{pmatrix} x - \tilde{x} \\ y - \tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq & (C_1 - 2B_\phi - C_2^*) \sqrt{\frac{\log(nd)}{n}} \|x - \tilde{x}\|_1 \\ & + (C_1^* - 2C_2^{-1} - C_1\sqrt{C_0}) \min\{\|x - \tilde{x}\|_2^2, \|x - \tilde{x}\|_2\} \\ & - \frac{\sigma}{2}\|y - \Phi x\|_2^2 - \left(\frac{C_3^* \sqrt{\log(nd)}}{n} + \frac{2}{\sigma n} + \frac{C_2 C_\lambda^2}{4} \cdot \frac{s_* \log(nd)}{n}\right).\end{aligned}$$

Choosing $C_1 = 2B_\phi + C_2^*$, , $C_2 = 8/C_1^*$, and $C_4 = C_3^* + 2 + \frac{C_2 C_\lambda^2}{4}$, and choosing C_0 to satisfy $C_0 \leq (C_1^*/4C_1)^2$, this simplifies to

$$\begin{aligned}\left\langle \begin{pmatrix} x - \tilde{x} \\ y - \tilde{y} \end{pmatrix}, \begin{pmatrix} \xi_x - \xi_{\tilde{x}} \\ \zeta_y - \zeta_{\tilde{y}} \end{pmatrix} \right\rangle \geq & \frac{C_1^*}{2} \min\{\|x - \tilde{x}\|_2^2, \|x - \tilde{x}\|_2\} \\ & - \frac{\sigma}{2}\|y - \Phi x\|_2^2 - C_4 \max\{1, \sigma^{-1}\} \cdot \frac{s_* \log(nd)}{n}.\end{aligned}$$

To complete our proof that (28) holds, we only need to verify that $C_1^*/2 \geq C_3$. Recall that we have defined this constant as $C_3 = 2\sqrt{C_0}(C_1 + 2B_\phi)$. Therefore, by taking $C_0 = (C_1^*/4(C_1 + 2B_\phi))^2$, all the necessary bounds are verified and we have completed the proof.

A.5.3 Proof of Lemma 3

For any fixed x, define

$$\tilde{H}(x) = \mathbb{E}\left[H(x)\right] = \mathbb{E}\left[(\phi^\top x - z)_+ \cdot \mathbb{1}\left\{z > 0\right\} + (-\phi^\top x + z)_+ \cdot \mathbb{1}\left\{z < 0\right\}\right],$$

where the expectation is taken with respect to $\phi \sim \mathcal{D}_\phi$ and $z \sim h_z$, with $\phi \perp\!\!\!\perp z$. We can calculate

$$\begin{aligned}
\mathbb{E}\left[(\phi^\top x - z)_+ \cdot \mathbb{1}\left\{z > 0\right\} \;\middle|\; \phi\right] &= \int_{t=0}^{(\phi^\top x)_+} \left[(\phi^\top x)_+ - t\right] h_z(t)\, \mathsf{d}t \\
&\geq \int_{t=0}^{\min\{t_z, (\phi^\top x)_+\}} \left[(\phi^\top x)_+ - t\right] c_z\, \mathsf{d}t \\
&= \left[c_z \min\{(\phi^\top x)_+^2, t_z(\phi^\top x)_+\} - \frac{c_z}{2}\min\{(\phi^\top x)_+, t_z\}^2\right] \\
&\geq \frac{c_z}{2}\min\{(\phi^\top x)_+^2, t_z(\phi^\top x)_+\}.
\end{aligned}$$

Similarly,

$$\mathbb{E}\left[(-\phi^\top x + z)_+ \cdot \mathbb{1}\left\{z < 0\right\} \;\middle|\; \phi\right] \geq \frac{c_z}{2}\min\{(-\phi^\top x)_+^2, t_z(-\phi^\top x)_+\}.$$

Therefore, for a fixed x,

$$\begin{aligned}
\tilde{H}(x) &= \mathbb{E}\left[(\phi^\top x - z)_+ \cdot \mathbb{1}\left\{z > 0\right\} + (-\phi^\top x + z)_+ \cdot \mathbb{1}\left\{z < 0\right\}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[(\phi^\top x - z)_+ \cdot \mathbb{1}\left\{z > 0\right\} + (-\phi^\top x + z)_+ \cdot \mathbb{1}\left\{z < 0\right\} \;\middle|\; \phi\right]\right] \\
&\geq \mathbb{E}\left[\frac{c_z}{2}\min\{(\phi^\top x)^2, t_z|\phi^\top x|\}\right].
\end{aligned}$$

Next, for any unit vector u, we can calculate

$$\begin{aligned}
\mathbb{E}\left[|\phi^\top u|^2 \cdot \mathbb{1}\left\{|\phi^\top u| \leq \frac{2b_\phi}{a_\phi}\right\}\right] &= \mathbb{E}\left[|\phi^\top u|^2\right] - \mathbb{E}\left[|\phi^\top u|^2 \cdot \mathbb{1}\left\{|\phi^\top u| > \frac{2b_\phi}{a_\phi}\right\}\right] \\
&\geq \mathbb{E}\left[|\phi^\top u|^2\right] - \frac{a_\phi}{2b_\phi}\mathbb{E}\left[|\phi^\top u|^3\right] \geq \frac{a_\phi}{2},
\end{aligned}$$

by our assumptions on $\mathcal{D}_\phi$. For any $x \neq 0$, writing $u = \frac{x}{\|x\|_2}$,

$$\begin{aligned}
&\min\left\{(\phi^\top x)^2, t_z|\phi^\top x|\right\} \\
&\quad= \min\left\{\|x\|_2^2 \cdot (\phi^\top u)^2, t_z\|x\|_2 \cdot |\phi^\top u|\right\} \\
&\quad\geq \min\left\{\|x\|_2^2, \frac{t_z a_\phi}{2b_\phi}\|x\|_2\right\} \cdot \min\left\{(\phi^\top u)^2, \frac{2b_\phi}{a_\phi}|\phi^\top u|\right\} \\
&\quad\geq \min\left\{\|x\|_2^2, \frac{t_z a_\phi}{2b_\phi}\|x\|_2\right\} \cdot (\phi^\top u)^2 \cdot \mathbb{1}\left\{|\phi^\top u| \leq \frac{2b_\phi}{a_\phi}\right\},
\end{aligned}$$

and therefore for all x,

$$\begin{aligned}\mathbb{E}\left[\min\left\{(\phi^\top x)^2, t_z|\phi^\top x|\right\}\right] &\geq \mathbb{E}\left[\min\left\{\|x\|_2^2, \frac{a_\phi t_z}{2b_\phi}\|x\|_2\right\}\cdot(\phi^\top u)^2\cdot\mathbb{1}\left\{|\phi^\top u|\leq\frac{2b_\phi}{a_\phi}\right\}\right] \\ &\geq \frac{a_\phi}{2}\min\left\{\|x\|_2^2, \frac{t_z a_\phi}{2b_\phi}\|x\|_2\right\}.\end{aligned}$$

Combining this with the work above,

$$\tilde{H}(x)\geq\frac{a_\phi c_z}{4}\min\left\{\|x\|_2^2, \frac{t_z a_\phi}{2b_\phi}\|x\|_2\right\}$$

for all $x\in\mathbb{R}^d$.

Next, we will use a peeling argument to bound $\left|H(x)-\tilde{H}(x)\right|$. First, fixing any $B>0$,

$$\begin{aligned}&\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|H(x)-\tilde{H}(x)\right|\right] \\ &\quad\leq 2\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i\xi_i\left[(\phi_i^\top x-z_i)_+\cdot\mathbb{1}\{z_i>0\}+(-\phi_i^\top x+z_i)_+\cdot\mathbb{1}\{z_i<0\}\right]\right|\right]\end{aligned}$$

by symmetrization (Koltchinskii, 2011, Theorem 2.1), where $\xi_1,\ldots,\xi_n\overset{\text{iid}}{\sim}\text{Unif}\{\pm1\}$. Next, fixing the z_i's, define $\varphi_i(t)=(t-z_i)_+\cdot\mathbb{1}\{z_i>0\}+(-t+z_i)_+\cdot\mathbb{1}\{z_i<0\}$. Then φ_i is 1-Lipschitz for all i, and so

$$\begin{aligned}&\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i\xi_i\left[(\phi_i^\top x-z_i)_+\cdot\mathbb{1}\{z_i>0\}+(-\phi_i^\top x+z_i)_+\cdot\mathbb{1}\{z_i<0\}\right]\right|\,\middle|\, z_{1:n}\right] \\ &\qquad\leq 2\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i\xi_i\cdot\phi_i^\top x\right|\,\middle|\, z_{1:n}\right]=2\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i\xi_i\cdot\phi_i^\top x\right|\right]\end{aligned}$$

by the Rademacher comparison inequality (Koltchinskii, 2011, Theorem 2.2). Finally,

$$\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i\xi_i\cdot\phi_i^\top x\right|\right]\leq\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\|\frac{1}{n}\Phi^\top\xi\|_\infty\|x\|_1\right]=B\mathbb{E}\left[\|\frac{1}{n}\Phi^\top\xi\|_\infty\right].$$

And, we know that $\|\frac{1}{n}\Phi^\top\xi\|_\infty\leq B_\phi$ deterministically (since $\|\xi\|_\infty\leq 1$ and $\|\Phi\|_\infty\leq B_\phi$), and so applying (31), we have

$$\mathbb{E}\left[\|\frac{1}{n}\Phi^\top\xi\|_\infty\right]\leq 2B_\phi\sqrt{\frac{\log(nd)}{n}}+\frac{B_\phi}{2nd}\leq 3B_\phi\sqrt{\frac{\log(nd)}{n}},$$

where the last step holds since $\frac{1}{2nd}\leq\sqrt{\frac{\log(nd)}{n}}$ for all $n\geq 4, d\geq 1$. So, we have

$$\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i\xi_i\cdot\phi_i^\top x\right|\right]\leq 3BB_\phi\sqrt{\frac{\log(nd)}{n}}.$$

Combining our work so far, we have shown that

$$\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|H(x)-\tilde{H}(x)\right|\right]\leq 4\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|\frac{1}{n}\sum_i \xi_i\cdot\phi_i^\top x\right|\right]\leq 12BB_\phi\sqrt{\frac{\log(nd)}{n}}.$$

Next, if we alter one data point ϕ_i, z_i, we can see that the value of $H(x)$ changes by at most $\frac{1}{n}B_\phi\|x\|_1$. Therefore, by McDiarmid's inequality, for any $t>0$,

$$\mathbb{P}\left\{\sup_{x:\|x\|_1\leq B}\left|H(x)-\tilde{H}(x)\right|>\mathbb{E}\left[\sup_{x:\|x\|_1\leq B}\left|H(x)-\tilde{H}(x)\right|\right]+t\right\}\leq\exp\left\{-\frac{2t^2}{n^{-1}B^2B_\phi^2}\right\}.$$

Setting $t=BB_\phi\sqrt{\log(nd)/n}$ and plugging in our calculation for the expected value,

$$\mathbb{P}\left\{\sup_{x:\|x\|_1\leq B}\left|H(x)-\tilde{H}(x)\right|>13BB_\phi\sqrt{\frac{\log(nd)}{n}}\right\}\leq(nd)^{-2}.$$

Therefore, applying this result with $B=\sqrt{d},2^{-1}\sqrt{d},\ldots,2^{-(K-1)}\sqrt{d}$ for $K=1+\lceil\frac{1}{2}\log_2(nd)\rceil$ (i.e., a peeling argument, with K chosen so that the smallest value of B is $\leq n^{-1/2}$), we see that $K\leq nd/2$ (this holds for any $n\geq 4, d\geq 1$), and so with probability at least $1-(2nd)^{-1}$,

$$\left|H(x)-\tilde{H}(x)\right|\leq\max\{\|x\|_1,n^{-1/2}\}\cdot 26B_\phi\sqrt{\frac{\log(nd)}{n}}$$

for all x with $\|x\|_1\leq\sqrt{d}$—and therefore, for all x with $\|x\|_2\leq 1$. Combining everything so far, we have shown that with probability at least $1-(2nd)^{-1}$,

$$H(x)\geq\frac{a_\phi c_z}{4}\min\left\{1,\frac{t_z a_\phi}{2b_\phi}\right\}\cdot\|x\|_2^2-\max\{\|x\|_1,n^{-1/2}\}\cdot 26B_\phi\sqrt{\frac{\log(nd)}{n}}\qquad(34)$$

for all $x\in\mathbb{R}^d$ with $\|x\|_2\leq 1$.

Now we consider x with $\|x\|_2\geq 1$. Let $x'=\frac{x}{\|x\|_2}$. Since $H(x)$ is convex, and $1=\|x'\|_2\leq\|x'\|_1$, if the bound (34) holds (at $x=x'$) then we have

$$\frac{1}{\|x\|_2}H(x)+\left(1-\frac{1}{\|x\|_2}\right)H(0)\geq H(x')\geq\frac{a_\phi c_z}{4}\min\left\{1,\frac{t_z a_\phi}{2b_\phi}\right\}-\|x'\|_1\cdot 26B_\phi\sqrt{\frac{\log(nd)}{n}}.$$

Clearly $H(0)=0$, and since $\|x\|_1=\|x'\|_1\cdot\|x\|_2$, we can simplify this to

$$H(x)\geq\frac{a_\phi c_z}{4}\min\left\{1,\frac{t_z a_\phi}{2b_\phi}\right\}\|x\|_2-\|x\|_1\cdot 26B_\phi\sqrt{\frac{\log(nd)}{n}}.$$

Combining both cases (i.e., $\|x\|_2\leq 1$ or $\|x\|_2>1$), we have therefore proved that, with probability at least $1-(2nd)^{-1}$,

$$H(x)\geq\frac{a_\phi c_z}{4}\min\left\{1,\frac{t_z a_\phi}{2b_\phi}\right\}\cdot\min\{\|x\|_2^2,\|x\|_2\}-\max\{\|x\|_1,n^{-1/2}\}\cdot 26B_\phi\sqrt{\frac{\log(nd)}{n}},$$

for all $x\in\mathbb{R}^d$, which completes the proof.