

**ISE**



Industrial and  
Systems Engineering

# Inexact Proximal-Gradient Methods with Support Identification

YUTONG DAI

Department of Industrial and Systems Engineering, Lehigh University

DANIEL P. ROBINSON

Department of Industrial and Systems Engineering, Lehigh University

COR@L Technical Report 22T-014



**LEHIGH**  
UNIVERSITY.

**COR@L**  
COMPUTATIONAL OPTIMIZATION  
RESEARCH AT LEHIGH 

# Inexact Proximal-Gradient Methods with Support Identification

YUTONG DAI<sup>\*1</sup> AND DANIEL P. ROBINSON<sup>†2</sup>

<sup>1</sup>Department of Industrial and Systems Engineering, Lehigh University

<sup>2</sup>Department of Industrial and Systems Engineering, Lehigh University

November 3, 2022

## Abstract

We consider the proximal-gradient method for minimizing an objective function that is the sum of a smooth function and a non-smooth convex function. A feature that distinguishes our work from most in the literature is that we assume that the associated proximal operator does not admit a closed-form solution. To address this challenge, we study two *adaptive* and *implementable* termination conditions that dictate how accurately the proximal-gradient subproblem is solved. We prove that the number of iterations required for the inexact proximal-gradient method to reach a  $\tau > 0$  approximate first-order stationary point is  $\mathcal{O}(\tau^{-2})$ , which matches the similar result that holds when exact subproblem solutions are computed. Also, by focusing on the overlapping group  $\ell_1$  regularizer, we propose an algorithm for approximately solving the proximal-gradient subproblem, and then prove that its iterates identify (asymptotically) the support of an optimal solution. If one imposes additional control over the accuracy to which each subproblem is solved, we give an upper bound on the maximum number of iterations before the support of an optimal solution is obtained.

## 1 Introduction

We consider the minimization of a function written as the sum of a smooth function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and a non-smooth convex function  $r : \mathbb{R}^n \rightarrow \mathbb{R}$ , which can be written as

$$\min_{x \in \mathbb{R}^n} f(x) + r(x). \quad (1)$$

Many algorithms exist for solving such problems (e.g., see [2, 4]), which cover a wide range of machine learning problems. A common choice for the regularizer  $r$  is the  $\ell_1$  norm [28], which is used to attain solutions to problem (1) that have few nonzero entries, i.e., a sparse solution. Computing such a solution is critical in machine learning applications that use model prediction since it is only the nonzero entries in the solution that define the model, and therefore sparse solutions define simpler and easier to interpret and understand models. Despite the successes of  $\ell_1$ -norm regularization, its inadequacy for some modern applications has been observed. For example, since covariates often come in groups (e.g., genes that regulate hormone levels), one may wish to select them jointly. Also, incorporating group information into the modeling process can improve both the interpretability and accuracy of the resulting model [34]. Therefore, new regularizers like the elastic-net [36] and group lasso [33] have been proposed to address these issues, and have been successful in practice for genome-wide SNP selection, predicting Parkinson's disease, and gene microarray selection [1, 8, 26, 35].

---

\*E-mail: yud319@lehigh.edu

†E-mail: daniel.p.robinson@lehigh.edu

The proximal-gradient (PG) method and its accelerated variants (APG) (e.g., see [4]) form an important class of algorithms for solving problems of the form given by (1). The core computation in a PG method is the evaluation of the proximal operator associated with the regularizer at a given point. In particular, given a proximal parameter  $\alpha > 0$ , the proximal operator associated with  $r(\cdot)$  evaluated at  $u \in \mathbb{R}^n$  is defined as the unique solution to the following optimization problem:

$$\text{prox}_{\alpha r}(u) := \underset{x \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2\alpha} \|x - u\|_2^2 + r(x). \quad (2)$$

The most basic PG method sets  $u \leftarrow x - \alpha \nabla f(x)$ , where  $x$  is the current estimate of a solution to problem (1), so that the PG method update takes the form  $x \leftarrow \text{prox}_{\alpha r}(x - \alpha \nabla f(x))$ . For relatively simple regularizers, such as the  $\ell_1$  norm, the solution to (2) can be computed in closed-form. However, for more complicated regularizers, such as the overlapping group  $\ell_1$  norm [32], the solution to (2) does not admit a closed-form solution, and therefore an iterative algorithm (the iterations of which are often called the *inner iterations*) is required to *approximately* solve problem (2) each iteration. As mentioned in [25], many numerical experiments have shown that PG methods often work well even when the proximal operator is computed inexactly by approximately solving (2). Consequently, research has been devoted to understanding how inexact evaluations of the proximal operator impact the convergence properties of the PG method; we discuss these efforts next.

## 1.1 Related Work

Since the PG method solves problem (1) by solving a sequence of subproblems of the form (2), to guarantee convergence of an *inexact* PG method, the errors when approximately solving (2) must be carefully managed. There are two basic types of strategies for controlling errors, an *absolute criterion* and an *adaptive criterion*.

Following the convention in [15], an *absolute criterion* defines *in advance* the error tolerance for subproblem (2) that is acceptable during each iteration. For example, a framework is analyzed in [25] that shows that if the error in solving (2) is decreased at the rate of  $\mathcal{O}(1/k^\delta)$ , where  $k$  is the iteration counter and  $\delta > 2$  (respectively,  $\delta > 4$ ) for the PG (respectively, APG) method, then the inexact PG (respectively, APG) method shares the same convergence rate as its exact counterpart. In that work, for  $\epsilon > 0$ , the authors define  $\hat{x}$  as an  $\epsilon$ -approximate solution to subproblem (2) if and only if it satisfies  $\frac{1}{2\alpha} \|\hat{x} - u\|_2^2 + r(\hat{x}) \leq \min_{x \in \mathbb{R}^n} (\frac{1}{2\alpha} \|x - u\|_2^2 + r(x)) + \epsilon$ . Other papers have considered different ways of characterizing an inexact solution to subproblem (2). For example, [30, Definition 2.1] defines  $\hat{x}$  to be an inexact solution if and only if  $(u - \hat{x})/\alpha \in \partial_\epsilon r(\hat{x})$  for an appropriate choice of  $u$  (see Definition 3.1 for the definition of the  $\epsilon$ -subdifferential  $\partial_\epsilon$ ), and then use this notion to analyze the convergence rate of the APG method. In [22, Definition 2.5],  $\hat{x}$  is considered an inexact solution to (2) if and only if  $\hat{x} = \text{prox}_{\alpha r}(u + e)$  for an appropriate choice of  $u$  and where  $\|e\| \leq \sqrt{2\alpha\epsilon}$ , and then analyze the convergence rate of primal-dual algorithms for solving a saddle-point problem. These latter two notions are more stringent than the notion in [25], and all three are different in how the error is decomposed among the  $\epsilon$ -subdifferential of  $r(x)$  and  $\frac{1}{2\alpha} \|x - u\|_2^2$  (for more details, see the discussion in [22, Lemma 2.6]).

The motivation for an *adaptive criterion* is to produce, given the current iterate  $x_k$ , an approximate solution  $x_{k+1}$  to subproblem (2) such that  $f(x_{k+1}) + r(x_{k+1})$  is “significantly” less than  $f(x_k) + r(x_k)$ . A criterion of this type, by its nature, cannot be define in advance (as is the case with an absolute criterion) because it must take information about  $f + r$  at  $x_k$  into account, thus making it adaptive. Moreover, as argued in [10], an absolute criterion may not be ideal for overall efficiency since the actual goal is not to find  $x_{k+1} \approx \text{prox}_{\alpha_k r}(u)$ , but rather to find an  $x_{k+1}$  that significantly reduces the objective function. The authors in [10] consider problem (1) for the special case when  $f$  is ill-conditioned and  $r \equiv 0$ , in which case the method is equivalent to an inexact proximal-point method. In particular, given  $x_k$  and  $\alpha_k > 0$ , they approximately solve  $\min_{x \in \mathbb{R}^n} (\frac{1}{2\alpha_k} \|x - x_k\|_2^2 + f(x))$ , where  $x_{k+1}$  is acceptable as an inexact solution if  $f(x_k) - f(x_{k+1}) \geq \eta \|\nabla f(x_{k+1})\|_2 \|x_k - x_{k+1}\|_2$  with  $\eta \in (0, 1)$ . This work is extended in [15] to allow for nontrivial  $r$  and acceleration by approximately solving  $\min_{x \in \mathbb{R}^n} (\frac{1}{2\alpha_k} \|x - x_k\|_2^2 + f(x) + r(x))$ , where  $x_{k+1}$  is an acceptable inexact solution if  $\frac{1}{2\alpha_k} \|x_{k+1} - x_k\|_2^2 + f(x_{k+1}) + r(x_{k+1}) - \min_{x \in \mathbb{R}^n} (\frac{1}{2\alpha_k} \|x - x_k\|_2^2 + f(x) +$

$r(x) \leq \kappa \|x_{k+1} - x_k\|_2^2$  for some  $\kappa > 0$ . Relating a measure of error in the subproblem to  $\mathcal{O}(\|x_{k+1} - x_k\|_2)$  dates back to [23] and has been extensively studied for the proximal-point method. To integrate second-derivative information, [14] proposed to approximately solve, for a symmetric matrix  $H$ , the subproblem

$$\min_{x \in \mathbb{R}^n} \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T H(x - x_k) + r(x) - r(x_k). \quad (3)$$

When  $H = (1/\alpha_k)I$  and  $u = x_k - \alpha_k \nabla f(x_k)$ , subproblem (3) recovers subproblem (2). The vector  $x_{k+1}$  is considered an acceptable approximate solution to subproblem (3) if it achieves a decrease in its objective function that is at least some fraction of that achieved by its exact solution. Verifying this adaptive criterion requires either estimating a tight lower bound of the minimal value for subproblem (3) or devising (when problem (3) is strongly convex) an algorithm with global linear convergence for (3). Recently, [22] designed two adaptive criteria specifically tailored to FISTA [3].

When  $r$  is a sparsity-inducing regularizer, one is interested in a support identification property (i.e., the iterates of an algorithm correctly identify the support of an optimal solution to problem (1) in finitely many iterations). It is shown in [20, 27] that when problem (2) is solved *exactly*, the PG and APG methods have the support identification property under non-degeneracy assumptions.<sup>1</sup> While support identification results are proved in [13] for a framework built upon subproblem (3), they either require the explicit computation of the exact solution to (2) or require its implicit computation by making an assumption on the inexact subproblem solution obtained. To the best of our knowledge, no research has explored the support identification property when problem (2) must be solved *inexactly*, which is a unique aspect of our paper.

## 1.2 Contributions

This paper makes the following contributions in the area of PG methodology.

- We propose an *inexact* PG framework that allows for two practical *adaptive criteria* for determining acceptable approximate solutions to the PG subproblem. The adaptive criteria are easier to verify compared to most in the literature. For example, compared with [15], our criteria are easier to verify for regularizers such as the overlapping group- $\ell_1$ ; to verify the adaptive criterion in [15], one must either evaluate the proximal operator of  $r$  exactly, which excludes the overlapping group- $\ell_1$ , or estimate a tight lower-bound of  $\min_{x \in \mathbb{R}^n} \frac{1}{2\alpha_k} \|x - u\|^2 + f(x) + r(x)$ , which can be as difficult as solving problem (1) itself.
- We provide a unified worst-case complexity bound analysis of our PG framework noted in the previous bullet point. Interestingly, the complexity result for our *inexact* PG framework matches the result that holds for the *exact* PG method.
- For the special case of the overlapping group- $\ell_1$  regularizer, whose proximal operator does not admit a closed-form solution, we propose an enhanced PG subproblem solver that is designed with support identification in mind. When our PG framework uses our PG subproblem solver, we prove (under common assumptions) a worst-case complexity result for the number of iterations until optimal support identification occurs. To the best of our knowledge, this is the first work that establishes such a result for a framework built upon implementable (verifiable) conditions when the regularizer does not have a closed-form solution.

## 1.3 Notation and assumptions

Let  $\mathbb{R}$  denote the set of real numbers,  $\mathbb{R}^n$  denote the set of  $n$ -dimensional real vectors, and  $\mathbb{R}^{m \times n}$  denote the set of  $m$ -by- $n$ -dimensional real matrices. Let  $\|\cdot\|$  denote the  $\ell_2$  norm. The set of nonnegative integers is denoted as  $\mathbb{N} := \{0, 1, 2, \dots\}$  and for any positive integer  $n$ , we define  $[n] := \{1, 2, \dots, n\}$ . For a matrix  $A \in \mathbb{R}^{m \times n}$  and index sets  $(I, I') \subseteq [m] \times [n]$ , we let  $A_{[I, \cdot]}$  denote the sub-matrix of  $A$  formed by taking

<sup>1</sup>Support identification is called active-set identification in [20] and manifold identification in [27].

rows from the index set  $I$  and all columns, and  $A_{[I,I']}$  denote the sub-matrix of  $A$  formed by taking rows in the index set  $I$  and columns in the index set  $I'$ . For a convex function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$ , denote its Fenchel conjugate as  $h^*$ . Given  $x \in \mathbb{R}^n$  and  $d \in \mathbb{R}^n$ , the directional derivative of  $h$  at  $x$  in the direction  $d$  is denoted by  $D_h(x; d)$ . Given a set  $\mathcal{C} \subseteq \mathbb{R}^n$  that is closed and convex, define the projection operator  $\text{Proj}_{\mathcal{C}}(\cdot) : \mathbb{R}^n \rightarrow \mathcal{C}$  as

$$\text{Proj}_{\mathcal{C}}(\bar{y}) := \underset{y \in \mathcal{C}}{\text{argmin}} \|y - \bar{y}\|_2,$$

which then also allows us to define the distance of a vector  $\bar{y} \in \mathbb{R}^n$  to the set  $\mathcal{C}$  as

$$\text{dist}(\bar{y}, \mathcal{C}) := \|\bar{y} - \text{Proj}_{\mathcal{C}}(\bar{y})\|.$$

For any  $\bar{x} \in \mathbb{R}^n$  and  $\bar{\alpha} > 0$ , we define the PG update as

$$T(\bar{x}, \bar{\alpha}) := \underset{x \in \mathbb{R}^n}{\text{argmin}} \phi(x; \bar{x}, \bar{\alpha}) \quad \text{with} \quad \phi(x; \bar{x}, \bar{\alpha}) := \frac{1}{2\bar{\alpha}} \|x - (\bar{x} - \bar{\alpha}\nabla f(\bar{x}))\|^2 + r(x). \quad (4)$$

Using the notion of  $\epsilon$ -inexactness from [25, equation (4)], we use  $\mathcal{T}_{\epsilon}(\bar{x}, \bar{\alpha})$  to denote the *set* of  $\epsilon$ -PG updates, which is defined by

$$\mathcal{T}_{\epsilon}(\bar{x}, \bar{\alpha}) := \{\hat{x} \in \mathbb{R}^n \mid \phi(\hat{x}; \bar{x}, \bar{\alpha}) \leq \phi(T(\bar{x}, \bar{\alpha}); \bar{x}, \bar{\alpha}) + \epsilon\}. \quad (5)$$

Note that it follows from these definitions that  $\mathcal{T}_0(\bar{x}, \bar{\alpha}) = \{T(\bar{x}, \bar{\alpha})\}$ .

Finally, we make the following assumption throughout the paper.

**Assumption 1.1.** Define  $\mathcal{L} := \{x \in \mathbb{R}^n : f(x) + r(x) \leq f(x_0) + r(x_0)\}$  where  $x_0$  is an initial solution estimate to (1). The function  $f$  is continuously differentiable and  $\nabla f$  is Lipschitz continuous with constant  $L_g$  on an open neighborhood containing  $\mathcal{L}$ . The function  $r$  is closed, proper, and convex. The function  $f + r$  is bounded below.

## 1.4 Organization

In Section 2, we describe the inexact PG method that uses adaptive termination criteria for solving problem (1). The convergence analysis and support identification result is presented in Section 3 and Section 4, respectively. Numerical results are given in Section 5, and concluding remarks are presented in Section 6.

## 2 Algorithm

Our proposed inexact PG method is stated as Algorithm 1. Given the  $k$ th iterate  $x_k$  and proximal parameter  $\alpha_k$ , the algorithm computes an  $\epsilon_k$ -PG update  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  and associated step  $s_k = \hat{x}_{k+1} - x_k$ , where the value for  $\epsilon_k$  depends on the value of the input parameter `option`  $\in \{\text{option\_1}, \text{option\_2}\}$ . In particular, if `option` = `option\_1`, then  $\epsilon_k = c_k \|s_k\|^2$  with  $c_k$  chosen in Step 5, while if `option` = `option\_2`, then  $\epsilon_k = \gamma_2 (\phi(x_k; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k))$  with  $\gamma_2 \in (0, 1/2]$ . Next,  $\Delta_k$  is set to a value that depends on the value of `option` (see Steps 7 and 10). The choices for  $\Delta_k$  and  $c_k$  and the restrictions on  $\gamma_1$  and  $\gamma_2$  in Algorithm 1 ensure that  $s_k$  is a descent direction for  $f + r$  at  $x_k$  (see Lemma 3.4), and that the line search performed in Steps 13–15 gives a decrease in the objective function proportional to  $\alpha_k \chi_k^2$  when the unit step is accepted (see Lemma 3.5(i)) where

$$\chi_k := \|T(x_k, \alpha_k) - x_k\| / \alpha_k \quad (6)$$

is a measure of first-order optimality for problem (1) at  $x_k$  (see [3, Theorem 10.7] for more details on  $\chi_k$ ). Finally, in Steps 17–21 the proximal parameter for the next iteration is set to a fraction of its current value anytime  $j > 0$  (i.e., anytime the unit step size is not accepted during the linesearch).

To successfully implement Algorithm 1, for each  $k \in \mathbb{N}$ , one must be able to compute the triple  $(\hat{x}_{k+1}, s_k, \epsilon_k)$  needed in Steps 6 and 9. As long as  $x_k$  is not optimal (i.e., as long as  $\chi_k > 0$ ), the existence of such points is guaranteed, although computing one is challenging for nontrivial regularizers. We consider this practical question in Section 4.

---

**Algorithm 1** Inexact PG Method for solving problem (1).

---

1: **Input:** Initial estimate  $x_0 \in \mathbb{R}^n$  and  $\text{option} \in \{\text{option\_1}, \text{option\_2}\}$ .  
2: **Constants:**  $\{\xi, \eta, \zeta\} \subset (0, 1)$ ,  $\alpha_0 \in (0, \infty)$ ,  $\gamma_1 \in (0, 2)$ , and  $\gamma_2 \in (0, 1/2]$   
3: **for**  $k = 0, 1, 2, \dots$  **do**  
4:   **if** ( $\text{option} = \text{option\_1}$ ) **then**  
5:     Choose  $c_k \in \left(0, \frac{1}{4} \left(\sqrt{\frac{6}{(1+\gamma_1)\alpha_k}} - \sqrt{\frac{2}{\alpha_k}}\right)^2\right]$ .  
6:     Compute a triple  $(\hat{x}_{k+1}, s_k, \epsilon_k)$  satisfying the conditions  

$$\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k), \quad s_k = \hat{x}_{k+1} - x_k, \quad \text{and} \quad \epsilon_k = c_k \|s_k\|^2.$$
  
7:     Set  $\Delta_k \leftarrow -\frac{1}{\alpha_k} \|s_k\|^2 + \sqrt{(2/\alpha_k)\epsilon_k} \|s_k\| + \epsilon_k$ .  
8:   **else** ( $\text{option} = \text{option\_2}$ )  
9:     Compute a triple  $(\hat{x}_{k+1}, s_k, \epsilon_k)$  satisfying the conditions  

$$\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k), \quad s_k = \hat{x}_{k+1} - x_k, \quad \text{and} \quad \epsilon_k = \gamma_2 (\phi(x_k; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k)).$$
  
10:     Set  $\Delta_k \leftarrow r(x_k + s_k) - r(x_k) + \nabla f(x_k)^T s_k$ .  
11:   **end if**  
12:   Set  $j \leftarrow 0$  and  $x_{k+1,0}^{\text{trial}} \leftarrow x_k + s_k$ .  
13:   **while**  $f(x_{k+1,j}^{\text{trial}}) + r(x_{k+1,j}^{\text{trial}}) > f(x_k) + r(x_k) + \eta \xi^j \Delta_k$  **do**  
14:     Set  $j \leftarrow j + 1$  and  $x_{k+1,j}^{\text{trial}} \leftarrow x_k + \xi^j s_k$ .  
15:   **end while**  
16:    $x_{k+1} \leftarrow x_{k+1,j}^{\text{trial}}$   
17:   **if**  $j = 0$  **then**  
18:      $\alpha_{k+1} \leftarrow \alpha_k$  and  $\text{flag}_k \leftarrow \text{same\_}\alpha$ .  
19:   **else**  
20:      $\alpha_{k+1} \leftarrow \zeta \alpha_k$  and  $\text{flag}_k \leftarrow \text{decrease\_}\alpha$ .  
21:   **end if**  
22: **end for**

---

### 3 Convergence Analysis

In this section we analyze Algorithm 1. In particular, in Section 3.1, we give preliminary results related to  $\epsilon$ -PG updates, and in Section 3.2 we present a complexity analysis.

#### 3.1 Preliminary results

In this section, we discuss the  $\epsilon$ -subdifferential of a function, and characterize the properties of the  $\epsilon$ -PG update. We begin with the definition of the  $\epsilon$ -subdifferential.

**Definition 3.1** ( $\epsilon$ -subdifferential). [7, Section 4.3] For a given convex function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$ , vector  $x \in \mathbb{R}^n$ , and scalar  $\epsilon > 0$ , the  $\epsilon$ -subdifferential of  $h$  at  $x \in \mathbb{R}^n$  is

$$\partial_\epsilon h(x) := \{d \in \mathbb{R}^n \mid h(z) \geq h(x) + d^T(z - x) - \epsilon \text{ for all } z \in \mathbb{R}^n\}.$$

We now bound the difference between an  $\epsilon$ -PG update and the exact PG update.

**Lemma 3.2** ([24]). *Let  $\bar{x} \in \mathbb{R}^n$ ,  $\bar{\alpha} > 0$ , and  $\epsilon > 0$ . It holds that*

$$\|T(\bar{x}, \bar{\alpha}) - \hat{x}\|^2 \leq 2\bar{\alpha}\epsilon \text{ for all } \hat{x} \in \mathcal{T}_\epsilon(\bar{x}, \bar{\alpha}).$$

*Proof.* Note that  $\phi(x; \bar{x}, \bar{\alpha})$  (see (4)) is strongly convex as a function of  $x$  with strong convexity parameter greater than or equal to  $1/\bar{\alpha}$ . Therefore, we may conclude that

$$\epsilon \geq \phi(\hat{x}; \bar{x}, \bar{\alpha}) - \phi(T(\bar{x}, \bar{\alpha}); \bar{x}, \bar{\alpha}) \geq \frac{1}{2\bar{\alpha}} \|T(\bar{x}, \bar{\alpha}) - \hat{x}\|^2 \quad \text{for all } \hat{x} \in \mathcal{T}_\epsilon(\bar{x}, \bar{\alpha})$$

where the last inequality follows from  $0 \in \partial\phi(T(\bar{x}, \bar{\alpha}); \bar{x}, \bar{\alpha})$  (i.e., it follows from the optimality conditions for problem (4)).  $\square$

Part (i) of the next result constructs two upper-bounds on the directional derivative of the objective function  $f + r$  at  $\bar{x}$  along the direction associated with the  $\epsilon$ -PG update. This result suggests that by choosing  $\epsilon$  sufficiently small, the directional derivative will be negative. For additional insight, the reader should compare them to the definition of  $\Delta_k$  in lines 7 and 10 of Algorithm 1. The second part of the next result bounds the change in  $f + r$  that results from taking the step along the direction.

**Lemma 3.3.** *Let  $\bar{x} \in \mathbb{R}^n$ ,  $\bar{\alpha} > 0$ , and  $\epsilon > 0$ . If  $\hat{x} \in \mathcal{T}_\epsilon(\bar{x}, \bar{\alpha})$ , then  $s = \hat{x} - \bar{x}$  satisfies the following results.*

(i) *The directional derivative of  $f + r$  at  $\bar{x}$  in the direction  $s$  satisfies*

$$D_{f+r}(\bar{x}; s) \leq -\frac{1}{\bar{\alpha}} \|s\|^2 + \sqrt{\frac{2\epsilon}{\bar{\alpha}}} \|s\| + \epsilon \quad \text{and} \quad D_{f+r}(\bar{x}; s) \leq r(\bar{x} + s) - r(\bar{x}) + \nabla f(\bar{x})^T s.$$

(ii) *A bound on the change in the objective function after taking the step  $s$  is*

$$f(\bar{x} + s) + r(\bar{x} + s) \leq f(\bar{x}) + r(\bar{x}) - \left(\frac{1}{\bar{\alpha}} - \frac{L_g}{2}\right) \|s\|^2 + \sqrt{\frac{2\epsilon}{\bar{\alpha}}} \|s\| + \epsilon.$$

*Proof.* Let  $\hat{x} \in \mathcal{T}_\epsilon(\bar{x}, \bar{\alpha})$ . It follows from [25, Lemma 2] that there is a  $w \in \mathbb{R}^n$  so that

$$\|w\| \leq \sqrt{2\bar{\alpha}\epsilon} \quad \text{and} \quad g_\epsilon := \frac{1}{\bar{\alpha}} [\bar{x} - \bar{\alpha}\nabla f(\bar{x}) + w - \hat{x}] \in \partial_\epsilon r(\hat{x}). \quad (7)$$

From  $g_\epsilon \in \partial_\epsilon r(\hat{x})$  and convexity of  $r(x)$ , it follows that

$$r(\bar{x}) \geq r(\hat{x}) + g_\epsilon^T (\bar{x} - \hat{x}) - \epsilon \quad \text{and} \quad r(\hat{x}) \geq r(\bar{x}) + g_r^T (\hat{x} - \bar{x}) \quad \text{for all } g_r \in \partial r(\bar{x}). \quad (8)$$

Let us now proceed to prove part (i). Adding the two inequalities in (8) together gives  $(g_r - g_\epsilon)^T (\hat{x} - \bar{x}) \leq \epsilon$  for all  $g_r \in \partial r(\bar{x})$ . This inequality may then be combined with  $s = \hat{x} - \bar{x}$ , the definition of  $g_\epsilon$ , the Cauchy-Schwarz inequality, and (7) to obtain

$$\begin{aligned} s^T (\nabla f(\bar{x}) + g_r) &= (\hat{x} - \bar{x})^T (\nabla f(\bar{x}) + g_r) = (\hat{x} - \bar{x})^T \left( \frac{1}{\bar{\alpha}} (\bar{x} - \hat{x} + w) + g_r - g_\epsilon \right) \\ &= -\frac{1}{\bar{\alpha}} \|\hat{x} - \bar{x}\|^2 + \frac{1}{\bar{\alpha}} (\hat{x} - \bar{x})^T w + (\hat{x} - \bar{x})^T (g_r - g_\epsilon) \\ &\leq -\frac{1}{\bar{\alpha}} \|s\|^2 + \frac{1}{\bar{\alpha}} \sqrt{2\bar{\alpha}\epsilon} \|s\| + \epsilon \quad \text{for all } g_r \in \partial r(\bar{x}). \end{aligned} \quad (9)$$

By the definition of the directional derivative, the fact that  $f(x)$  is differentiable, convexity of  $r$ , the result in [18, Theorem 2.87], and (9), it follows that

$$D_{f+r}(\bar{x}; s) = s^T \nabla f(\bar{x}) + \sup_{g_r \in \partial r(\bar{x})} s^T g_r \leq -\frac{1}{\bar{\alpha}} \|s\|^2 + \sqrt{\frac{2\epsilon}{\bar{\alpha}}} \|s\| + \epsilon,$$

which completes the proof of the first inequality in part (i). The proof of the second inequality in part (i) follows directly from the proof in [14, Lemma 1].

To prove part (ii), we can use the Lipschitz continuity of  $\nabla f$  with Lipschitz constant  $L_g$  (see Assumption 1.1),  $\hat{x} = \bar{x} + s$ , and the first inequality in (8) to obtain

$$f(\bar{x} + s) + r(\bar{x} + s) \leq f(\bar{x}) + \nabla f(\bar{x})^T s + \frac{L_g}{2} \|s\|^2 + r(\bar{x}) + g_\epsilon^T s + \epsilon$$

$$= f(\bar{x}) + r(\bar{x}) + (\nabla f(\bar{x}) + g_\epsilon)^T s + \frac{L_g}{2} \|s\|^2 + \epsilon.$$

Combining this inequality with (7) and the Cauchy-Schwarz inequality gives

$$\begin{aligned} f(\bar{x} + s) + r(\bar{x} + s) &\leq f(\bar{x}) + r(\bar{x}) - \frac{1}{\alpha}(s - w)^T s + \frac{L_g}{2} \|s\|^2 + \epsilon \\ &\leq f(\bar{x}) + r(\bar{x}) - \left(\frac{1}{\alpha} - \frac{L_g}{2}\right) \|s\|^2 + \frac{1}{\alpha} \|w\| \|s\| + \epsilon \\ &\leq f(\bar{x}) + r(\bar{x}) - \left(\frac{1}{\alpha} - \frac{L_g}{2}\right) \|s\|^2 + \frac{1}{\alpha} \sqrt{2\alpha\epsilon} \|s\| + \epsilon, \end{aligned}$$

which completes the proof.  $\square$

### 3.2 Global Complexity

In this section, we analyze the worst-case iteration complexity of Algorithm 1 for identifying an approximate stationary point of problem (1). Specifically, given a tolerance  $\tau \in (0, 1]$ , we derive an upper bound on the number of iterations until  $\chi_k \leq \tau$ . We will assume throughout, for each  $k \in \mathbb{N}$ , that  $\chi_k > 0$  since if  $\chi_k = 0$  then  $x_k$  would be an *exact* first-order solution. The next result gives an upper bound on  $\Delta_k$ , which in turn gives an upper bound on the directional derivative of  $f + r$  at  $x_k$  in the direction  $s_k$ .

**Lemma 3.4.** *For each  $k \in \mathbb{N}$ , the directional derivative of  $f + r$  satisfies*

$$D_{f+r}(x_k; s_k) \leq \Delta_k \leq -\beta \alpha_k \chi_k^2 < 0 \quad \text{where} \quad \beta := \begin{cases} \frac{\gamma_1}{2} & \text{if option} = \text{option}_1, \\ \frac{1}{4} & \text{if option} = \text{option}_2. \end{cases} \quad (10)$$

*Proof.* We start by discussing properties of  $\hat{x}_{k+1}$  and  $\epsilon_k$  that hold regardless of the value of `option`. It follows from (6) and the triangle inequality that  $\hat{x}_{k+1}$  satisfies

$$\begin{aligned} \chi_k^2 &= \frac{\|T(x_k, \alpha_k) - \hat{x}_{k+1} + \hat{x}_{k+1} - x_k\|^2}{\alpha_k^2} \\ &\leq \frac{\|T(x_k, \alpha_k) - \hat{x}_{k+1}\|^2}{\alpha_k^2} + \frac{2\|T(x_k, \alpha_k) - \hat{x}_{k+1}\| \|\hat{x}_{k+1} - x_k\|}{\alpha_k^2} + \frac{\|\hat{x}_{k+1} - x_k\|^2}{\alpha_k^2}. \end{aligned} \quad (11)$$

Also, since  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$ , we can apply Lemma 3.2 with  $\hat{x} = \hat{x}_{k+1}$ ,  $\bar{x} = x_k$ ,  $\bar{\alpha} = \alpha_k$ , and  $\epsilon = \epsilon_k$  to get

$$\|T(x_k, \alpha_k) - \hat{x}_{k+1}\|^2 \leq 2\alpha_k \epsilon_k. \quad (12)$$

Now, let us consider the two cases determined by the value of the parameter `option`.

**Case 1 (option = option\_1).** It follows from (11), (12), and  $s_k = \hat{x}_{k+1} - x_k$  that

$$\alpha_k \chi_k^2 \leq 2\epsilon_k + \frac{2\sqrt{2\alpha_k \epsilon_k} \|s_k\|}{\alpha_k} + \frac{\|s_k\|^2}{\alpha_k}. \quad (13)$$

Next, it follows from Step 6 and how  $c_k$  is chosen in Step 5 that

$$\epsilon_k = c_k \|s_k\|^2 \in \left(0, \frac{1}{4} \left( \sqrt{\frac{6}{(1+\gamma_1)\alpha_k}} - \sqrt{\frac{2}{\alpha_k}} \right)^2 \|s_k\|^2 \right].$$

Using the fact that  $\epsilon_k$  lies in this range, it can be verified that

$$\frac{\gamma_1}{2\alpha_k} \left( 2\alpha_k \epsilon_k + 2\sqrt{2\alpha_k \epsilon_k} \|s_k\| + \|s_k\|^2 \right) \leq \frac{1}{\alpha_k} \|s_k\|^2 - \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| - \epsilon_k. \quad (14)$$

Next, it follows from Step 7, (14), and (13) that

$$\Delta_k = - \left( \frac{1}{\alpha_k} \|s_k\|^2 - \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| - \epsilon_k \right) \leq - \frac{\gamma_1}{2\alpha_k} \left( 2\alpha_k \epsilon_k + 2\sqrt{2\alpha_k \epsilon_k} \|s_k\| + \|s_k\|^2 \right)$$



$$= -\frac{\gamma_1}{2} \left( 2\epsilon_k + \frac{2\sqrt{2\alpha_k\epsilon_k}\|s_k\|}{\alpha_k} + \frac{\|s_k\|^2}{\alpha_k} \right) \leq -\frac{\gamma_1}{2} \alpha_k \chi_k^2. \quad (15)$$

Finally, we can combine the equality in (15) with Lemma 3.3(i) to obtain

$$D_{f+r}(x_k; s_k) \leq -\left( \frac{1}{\alpha_k} \|s_k\|^2 - \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| - \epsilon_k \right) = \Delta_k.$$

This result together with (15) completes the proof for this case.

**Case 2 (option = option\_2).** From  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  and the definition of  $\epsilon_k$  in Step 9, we have

$$\phi(\hat{x}_{k+1}; x_k, \alpha_k) \leq \phi(T(x_k, \alpha_k); x_k, \alpha_k) + \gamma_2(\phi(x_k; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k)).$$

By adding the term  $-\phi(x_k; x_k, \alpha_k)$  to both sides of the equation, rearranging terms, and then using the definition of  $\epsilon_k$  in Step 9, we obtain

$$\begin{aligned} \phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi(x_k; x_k, \alpha_k) &\leq (\gamma_2 - 1)(\phi(x_k; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k)) \\ &= \left( \frac{\gamma_2 - 1}{\gamma_2} \right) \epsilon_k. \end{aligned} \quad (16)$$

Using algebraic simplification and the definition of  $\Delta_k$  in Step 10, we obtain

$$\begin{aligned} \phi(x_k; x_k, \alpha_k) - \phi(\hat{x}_{k+1}; x_k, \alpha_k) &= -\nabla f(x_k)^T s_k - \frac{1}{2\alpha_k} \|s_k\|^2 + r(x_k) - r(\hat{x}_{k+1}) \\ &= -\Delta_k - \frac{1}{2\alpha_k} \|s_k\|^2. \end{aligned}$$

By combining this equality with (16) and recalling that  $\gamma_2 \in (0, 1/2]$ , we get

$$\begin{aligned} -\Delta_k &= \phi(x_k; x_k, \alpha_k) - \phi(\hat{x}_{k+1}; x_k, \alpha_k) + \frac{1}{2\alpha_k} \|s_k\|^2 \\ &\geq \left( \frac{1 - \gamma_2}{\gamma_2} \right) \epsilon_k + \frac{1}{2\alpha_k} \|s_k\|^2 \geq \epsilon_k + \frac{1}{2\alpha_k} \|s_k\|^2. \end{aligned} \quad (17)$$

Applying Young's inequality to the product of norms in (11) and using (12), we obtain

$$\chi_k^2 \leq \frac{2\|T(x_k, \alpha_k) - \hat{x}_{k+1}\|^2}{\alpha_k^2} + \frac{2\|s_k\|^2}{\alpha_k^2} \leq \frac{4\epsilon_k}{\alpha_k} + \frac{2\|s_k\|^2}{\alpha_k^2}. \quad (18)$$

Multiplying both sides of (18) by  $\alpha_k/4$ , and then applying (17), using the definition of  $\Delta_k$  in Step 10, and Lemma 3.3(i) with  $\hat{x} = \hat{x}_{k+1}$ ,  $\bar{x} = x_k$ ,  $s = s_k$ , and  $\epsilon = \epsilon_k$  gives

$$\begin{aligned} \frac{\alpha_k}{4} \chi_k^2 &\leq \epsilon_k + \frac{1}{2\alpha_k} \|s_k\|^2 \leq -\Delta_k \\ &= -(r(x_k + s_k) - r(x_k) + \nabla f(x_k)^T s_k) \leq -D_{f+r}(x_k; s_k). \end{aligned} \quad (19)$$

Multiplying this inequality by a negative one completes the proof for this case.

It follows from  $\alpha_k \in (0, \infty)$  and  $\chi_k > 0$  that  $-\beta\alpha_k\chi_k^2 < 0$ , as claimed in (10).  $\square$

It is convenient to define the following partition of iterations performed by Algorithm 1 that depend on the status of the PG parameter:

$$\begin{aligned} \mathcal{K}_{\rightarrow} &:= \{k \in \mathbb{N} : \text{flag}_k = \text{same\_}\alpha \text{ in Line 18}\} \quad \text{and} \\ \mathcal{K}_{\downarrow} &:= \{k \in \mathbb{N} : \text{flag}_k = \text{decrease\_}\alpha \text{ in Line 20}\}. \end{aligned}$$

Note that  $\mathcal{K}_{\rightarrow} \cap \mathcal{K}_{\downarrow} = \emptyset$  and that every iteration of the algorithm is in  $\mathcal{K}_{\rightarrow} \cup \mathcal{K}_{\downarrow}$ . Using these sets, the next result shows that the **while** loop in Algorithm 1 always terminates and that the new iterate produces a useful decrease in the objective function  $f + r$ .

**Lemma 3.5.** *For each  $k \in \mathbb{N}$ , the **while** loop in Step 13 terminates finitely. Also, the following hold:*

(i) If  $k \in \mathcal{K}_\rightarrow$ , then  $\alpha_{k+1} = \alpha_k$  and, with  $\beta$  defined in (10), it holds that

$$f(x_{k+1}) + r(x_{k+1}) \leq f(x_k) + r(x_k) + \eta\Delta_k \leq f(x_k) + r(x_k) - \eta\beta\alpha_k\chi_k^2.$$

(ii) If  $k \in \mathcal{K}_\downarrow$ , then  $\alpha_{k+1} = \zeta\alpha_k$  and  $f(x_{k+1}) + r(x_{k+1}) < f(x_k) + r(x_k)$ .

Thus, the objective function  $f + r$  is monotonically strictly decreasing over  $\{x_k\}$ .

*Proof.* From Lemma 3.4 we have  $D_{f+r}(x_k; s_k) < 0$  so that standard results for a backtracking Armijo linesearch ensure that the **while** loop starting in Step 13 will terminate finitely, thus proving the first claim.

To prove part (i), suppose that  $k \in \mathcal{K}_\rightarrow$ . From the definition of  $\mathcal{K}_\rightarrow$  we know that  $\text{flag}_k = \text{same}_\alpha$ , and thus from Step 18 that  $\alpha_{k+1} = \alpha_k$  and that  $j = 0$  when Step 17 is reached. This latter fact means that the inequality in Step 13 does not hold for  $j = 0$ , meaning that the computed  $x_{k+1}$  satisfies  $x_{k+1} = x_{k+1,0}^{\text{trial}} = x_k + s_k$  and  $f(x_{k+1}) + r(x_{k+1}) \leq f(x_k) + r(x_k) + \eta\Delta_k$ . Combining this inequality with Lemma 3.4 gives

$$f(x_{k+1}) + r(x_{k+1}) \leq f(x_k) + r(x_k) + \eta\Delta_k \leq f(x_k) + r(x_k) - \eta\beta\alpha_k\chi_k^2$$

as claimed, thus completing the proof of part (i).

To prove part (ii), suppose that  $k \in \mathcal{K}_\downarrow$ . From the definition of  $\mathcal{K}_\downarrow$  we know that  $\text{flag}_k = \text{decrease}_\alpha$ , and therefore from Step 20 we have that  $\alpha_{k+1} = \zeta\alpha_k$  and that  $j > 0$  when Step 17 is reached. Moreover, the linesearch in Steps 13–15 produces a vector  $x_{k+1,j}^{\text{trial}} \leftarrow x_k + \xi^j s_k$  that, with Lemma 3.4, satisfies

$$f(x_{k+1,j}^{\text{trial}}) + r(x_{k+1,j}^{\text{trial}}) \leq f(x_k) + r(x_k) + \eta\xi^j\Delta_k \leq f(x_k) + r(x_k) - \eta\xi^j\beta\alpha_k\chi_k^2. \quad (20)$$

Combining this with  $x_{k+1} \leftarrow x_{k+1,j}^{\text{trial}}$ ,  $\alpha_k > 0$ , and  $\chi_k > 0$  establishes that

$$f(x_{k+1}) + r(x_{k+1}) = f(x_{k+1,j}^{\text{trial}}) + r(x_{k+1,j}^{\text{trial}}) \leq f(x_k) + r(x_k) - \eta\xi^j\beta\alpha_k\chi_k^2 < f(x_k) + r(x_k),$$

which completes the proof of part (ii).

Finally, the objective function  $f + r$  is monotonically strictly decreasing over  $\{x_k\}$  as a consequence of parts (i) and (ii),  $\eta \in (0, 1)$ ,  $\beta > 0$ ,  $\alpha_k > 0$ , and  $\chi_k > 0$ .  $\square$

We now show that the PG parameters are bounded away from zero, thus implying that the unit step size is accepted for all iterations with sufficiently large index.

**Lemma 3.6.** *For all  $k \in \mathbb{N}$ , the PG parameter  $\alpha_k$  satisfies*

$$\alpha_k \geq \alpha_{\min} := \begin{cases} \min \left\{ \alpha_0, \frac{3\gamma_1\zeta(1-\eta)}{L_g(1+\gamma_1)} \right\} & \text{if option} = \text{option}_1, \\ \min \left\{ \alpha_0, \frac{\zeta(1-\eta)}{L_g} \right\} & \text{if option} = \text{option}_2. \end{cases} \quad (21)$$

Moreover, a bound on the number of times the PG parameter is decreased is given by

$$|\mathcal{K}_\downarrow| \leq c_\downarrow^\alpha := \frac{\log(\alpha_{\min}/\alpha_0)}{\log(\zeta)}. \quad (22)$$

*Proof.* We consider the two cases depending on the value of **option**.

**Case 1 (option = option<sub>1</sub>).** To prove (21), we first establish the following:

$$\text{if } \alpha_k \leq \frac{3\gamma_1(1-\eta)}{L_g(1+\gamma_1)} \text{ then } \eta\left(\frac{1}{\alpha_k} \|s_k\|^2 - \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| - \epsilon_k\right) \leq \left(\frac{1}{\alpha_k} - \frac{L_g}{2}\right) \|s_k\|^2 - \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| - \epsilon_k. \quad (23)$$

Using basic algebra, it may be shown that (23) is equivalent to establishing that

$$\text{if } \alpha_k \leq \frac{3\gamma_1(1-\eta)}{L_g(1+\gamma_1)} \text{ then } \left(\frac{1}{\alpha_k} - \frac{L_g}{2(1-\eta)}\right) \|s_k\|^2 \geq \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| + \epsilon_k. \quad (24)$$

Note that by rearranging (14), one can obtain

$$\frac{2-\gamma_1}{2(1+\gamma_1)\alpha_k} \|s_k\|^2 \geq \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| + \epsilon_k. \quad (25)$$

Next, notice that if  $\alpha_k \leq \frac{3\gamma_1(1-\eta)}{L_g(1+\gamma_1)}$  then  $(\frac{1}{\alpha_k} - \frac{L_g}{2(1-\eta)}) \|s_k\|^2 \geq \frac{2-\gamma_1}{2(1+\gamma_1)\alpha_k} \|s_k\|^2$ , which together with (25) proves that (24) holds (equivalently, that (23) holds). Combining (23) with  $x_{k+1,0}^{\text{trial}} = x_k + s_k = \hat{x}_{k+1}$ , Lemma 3.3(ii) with  $\hat{x} = \hat{x}_{k+1}$ ,  $\bar{x} = x_k$ ,  $\epsilon = \epsilon_k$ , and  $\bar{\alpha} = \alpha_k$ , and Step 7 yields

$$\begin{aligned} \text{if } \alpha_k \leq \frac{3\gamma_1(1-\eta)}{L_g(1+\gamma_1)} \text{ then } f(x_{k+1,0}^{\text{trial}}) + r(x_{k+1,0}^{\text{trial}}) & \\ & \leq f(x_k) + r(x_k) - \left(\frac{1}{\alpha_k} - \frac{L_g}{2}\right) \|s_k\|^2 + \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| + \epsilon_k \\ & \leq f(x_k) + r(x_k) - \eta \left(\frac{1}{\alpha_k} \|s_k\|^2 - \sqrt{\frac{2\epsilon_k}{\alpha_k}} \|s_k\| - \epsilon_k\right) \\ & = f(x_k) + r(x_k) + \eta \Delta_k. \end{aligned}$$

This inequality implies that the condition checked in Line 13 for  $j = 0$  will not hold so that  $j = 0$  when Line 17 is reached, meaning that the update  $\alpha_{k+1} \leftarrow \alpha_k$  will take place. Summarizing, we have shown that, for any  $k \in \mathbb{N}$  satisfying  $\alpha_k \leq \frac{3\gamma_1(1-\eta)}{L_g(1+\gamma_1)}$ , it holds that  $\alpha_{k+1} = \alpha_k$ . Combining this observation with the fact that the  $\alpha_0$  input to the algorithm can be any positive number, and that anytime the PG parameter is decreased it is by a factor  $\zeta$  (see Step 20), allows us to conclude that (21) holds.

**Case 2 (option = option\_2).** Note that the inequality in (17) still holds, and thus  $-\Delta_k \geq \frac{1}{2\alpha_k} \|s_k\|^2$  with  $\Delta_k$  defined in Step 10. Combining this inequality with the Lipschitz continuity assumption on  $\nabla f$  (see Assumption 1.1),  $x_{k+1,0}^{\text{trial}} = x_k + s_k$ , and the definition of  $\Delta_k$  in Step 10, one obtains

$$\begin{aligned} f(x_{k+1,0}^{\text{trial}}) + r(x_{k+1,0}^{\text{trial}}) & \\ & \leq f(x_k) + \nabla f(x_k)^T s_k + \frac{L_g}{2} \|s_k\|^2 + r(x_k + s_k) \\ & = f(x_k) + r(x_k) + \nabla f(x_k)^T s_k + \frac{L_g}{2} \|s_k\|^2 + r(x_k + s_k) - r(x_k) \\ & = f(x_k) + r(x_k) + \Delta_k + \frac{L_g}{2} \|s_k\|^2 \leq f(x_k) + r(x_k) + \Delta_k(1 - \alpha_k L_g). \end{aligned} \quad (26)$$

From this inequality and  $\Delta_k < 0$  (see Lemma 3.4), we can observe that if  $\alpha_k \leq (1-\eta)/L_g$ , then  $f(x_{k+1,0}^{\text{trial}}) + r(x_{k+1,0}^{\text{trial}}) \leq f(x_k) + r(x_k) + \eta \Delta_k$ , which means that the condition checked in Line 13 for  $j = 0$  will not hold, which in turn means that  $j = 0$  when Line 17 is reached so that the update  $\alpha_{k+1} \leftarrow \alpha_k$  would take place. Combining this observation with the fact that the  $\alpha_0$  input to the algorithm can be any positive number, and that anytime the PG parameter is decreased it is done so by a factor of  $\zeta$  (see Step 20), allows us to conclude that (21) holds.

As for (22), an upper bound on  $|\mathcal{K}_\downarrow|$  is the smallest integer  $\ell$  such that  $\alpha_0 \zeta^\ell \leq \alpha_{\min}$ . Solving this inequality for  $\ell$  shows that the result in (22) holds.  $\square$

The main theorem is now stated. It gives an upper bound on the number of iterations performed by Algorithm 1 before an approximate first-order solution is found.

**Theorem 3.7.** *For any  $\tau \in (0, 1]$ , the size of the set  $\mathcal{K}_\tau := \{k \in \mathbb{N} : \chi_k > \tau\}$  satisfies*

$$|\mathcal{K}_\rightarrow \cap \mathcal{K}_\tau| \leq c_{pg} \tau^{-2} \quad \text{with} \quad c_{pg} := \frac{f(x_0) + r(x_0) - \inf_{x \in \mathbb{R}^n} (f(x) + r(x))}{\eta \beta \alpha_{\min}} \quad (27)$$

where  $\beta$  is defined in (10) and  $\alpha_{\min}$  is defined in Lemma 3.6. Moreover, combining this result with Lemma 3.6 and the definition of  $c_\downarrow^\alpha$  in (22) shows that

$$|\mathcal{K}_\tau| \leq c_\downarrow^\alpha + c_{pg} \tau^{-2}. \quad (28)$$

*Proof.* Let us define  $\nu_k = f(x_k) + r(x_k) - (f(x_{k+1}) + r(x_{k+1}))$ . This definition, Lemma 3.5, Lemma 3.6, and the definition of  $\mathcal{K}_\tau$ , it holds for arbitrary  $\bar{k} \in \mathbb{N}$  that

$$\begin{aligned} & f(x_0) + r(x_0) - (f(x_{\bar{k}+1}) + r(x_{\bar{k}+1})) \\ &= \sum_{0 \leq k \leq \bar{k}} \nu_k \geq \sum_{\substack{k \in \mathcal{K}_\rightarrow \cap \mathcal{K}_\tau \\ 0 \leq k \leq \bar{k}}} \nu_k \geq \sum_{\substack{k \in \mathcal{K}_\rightarrow \cap \mathcal{K}_\tau \\ 0 \leq k \leq \bar{k}}} \eta\beta\alpha_k\chi_k^2 \geq \sum_{\substack{k \in \mathcal{K}_\rightarrow \cap \mathcal{K}_\tau \\ 0 \leq k \leq \bar{k}}} \eta\beta\alpha_{\min}\tau^2. \end{aligned}$$

Taking  $\bar{k} \rightarrow \infty$  and using the monotonicity of the objective values in Lemma 3.5 gives

$$f(x_0) + r(x_0) - \inf_{x \in \mathbb{R}^n} (f(x) + r(x)) \geq |\mathcal{K}_\rightarrow \cap \mathcal{K}_\tau| \eta\beta\alpha_{\min}\tau^2, \quad (29)$$

which proves (27). Finally, (27),  $|\mathcal{K}_\tau| = |\mathcal{K}_\rightarrow \cap \mathcal{K}_\tau| + |\mathcal{K}_\downarrow \cap \mathcal{K}_\tau|$ , and (22) gives (28).  $\square$

Theorem 3.7 shows that the worst-case iteration complexity for our *inexact* PG method is  $\mathcal{O}(\tau^{-2})$ , which is the same result that holds for the *exact* PG method.

## 4 Sparse Regularizers and Finite Support Identification

In this section we focus our attention on the case when  $r$  is chosen as the overlapping group- $\ell_1$  regularizer, whose associated proximal operator does not admit a closed-form solution. This regularizer, which is studied in [12], [21], and [32] is defined as

$$r(x) = \sum_{i=1}^{n_g} [\lambda]_i \| [x]_{g_i} \| \quad (30)$$

where  $n_g \in \mathbb{N} \setminus \{0\}$  denotes the number of groups,  $\lambda \in \mathbb{R}^{n_g}$  is a vector of strictly positive weights for the groups, and, for each  $i \in [n_g]$ , we use  $g_i \subseteq [n]$  to denote the index set of the variables for the  $i$ th group and  $[x]_{g_i} \in \mathbb{R}^{|g_i|}$  to denote the subvector of  $x$  that corresponds to the elements in group  $g_i$ . Next, we denote the  $j$ th entry of the  $i$ th group by  $g_{i,j}$  for every  $i \in [n_g]$  and  $j \in [|g_i|]$ . Finally, we assume that every variable is included in at least one group, i.e., we assume that  $\cup_{i=1}^{n_g} g_i \equiv [n]$ . This regularizer imposes structured sparsity on the set of solutions to problem (1) (also see [21]).

Since this choice of  $r$  is a special case of the more general regularizer considered in the previous section, we know that the worse-case iteration complexity of Theorem 3.7 holds. Although this result is comforting, it does not shed light on whether the iterates generated by our inexact PG method can identify the support of an optimal solution. In this section, we shall see that consideration of this topic is somewhat delicate. Indeed, no matter how accurately problem (4) is approximately solved, it is not guaranteed that the approximate solution returned by Algorithm 1 is sparse. Rather, one needs to exploit the geometric structure of  $r(x)$  and design a specialized algorithm for approximately solving subproblem (4).

Let us formally define the support and the support identification.

**Definition 4.1** (support and support identification property). The *support* of a point  $x \in \mathbb{R}^n$  with respect to  $r$  in (30) is

$$\mathcal{S}(x) := \{i \subseteq [n_g] \mid [x]_{g_i} \neq 0\}.$$

We say that an algorithm has the *support identification property* if and only if it generates iterates  $\{x_k\}$  satisfying  $\mathcal{S}(x_k) = \mathcal{S}(x^*)$  for some solution  $x^* \in \mathbb{R}^n$  to (1) and all sufficiently large  $k$ .

It is desirable that an algorithm has the support identification property. First, an algorithm with the support identification property is appropriate for identifying the most “important” variables in a regression problem; an algorithm without such a property needs to perform ad-hoc post processing to obtain a sparse estimate. Second, a solver that possesses the support identification property is an appropriate choice to be used within second-order subspace acceleration methods (e.g., see [9]), which are known to be extremely efficient when the PG problem can be solved exactly.

Our method for approximately solving the PG subproblem, which will be proved to have the support identification property, is based on sufficiently reducing a certain primal-dual gap and using a special projection. We describe the dual formulation next.

#### 4.1 A dual formulation of the PG subproblem

The PG subproblem (4) with regularizer  $r$  given by (30) that is approximately solved during the  $k$ th iteration of Algorithm 1 (see Line 6, Line 9, and (5)) is given by

$$x_k^* := \operatorname{argmin}_{x \in \mathbb{R}^n} \phi(x; x_k, \alpha_k) \quad \text{with} \quad \phi(x; x_k, \alpha_k) := \frac{1}{2\alpha_k} \|x - u_k\|^2 + \sum_{i=1}^{n_g} [\lambda]_i \| [x]_{g_i} \| \quad (31)$$

with  $u_k = x_k - \alpha_k \nabla f(x_k)$ . By comparing the definition of  $x_k^*$  with (4), we have

$$x_k^* = T(x_k, \alpha_k). \quad (32)$$

Introducing a vector of auxiliary variables  $z \in \mathbb{R}^{n_g}$ , subproblem (31) is equivalent to

$$\min_{x, z} \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z \quad \text{s.t.} \quad \begin{bmatrix} [x]_{g_i} \\ [z]_i \end{bmatrix} \in \mathcal{K}_i \text{ for all } i \in [n_g]$$

with

$$\mathcal{K}_i := \left\{ \begin{bmatrix} v \\ \theta \end{bmatrix} \mid v \in \mathbb{R}^{|g_i|}, \theta \in \mathbb{R}, \text{ and } \|v\| \leq \theta \right\} \text{ for all } i \in [n_g]$$

so that  $\mathcal{K}_i$  is a second-order cone for each  $i \in [n_g]$ . Denoting the characteristic function by  $\delta_{\mathcal{K}_i} : \mathbb{R}^{|g_i|+1} \rightarrow \{0, \infty\}$ , which is defined as  $\delta_{\mathcal{K}_i}(w) = 0$  if  $w \in \mathcal{K}_i$  and is equal to  $\infty$  otherwise, allows us to write this problem as

$$\min_{x, z} \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z + \sum_{i=1}^{n_g} \delta_{\mathcal{K}_i} \left( \begin{bmatrix} [x]_{g_i} \\ [z]_i \end{bmatrix} \right). \quad (33)$$

Introducing a set of auxiliary vectors  $\{p_i\}_{i=1}^{n_g}$  with  $p_i \in \mathbb{R}^{|g_i|}$  for each  $i \in [n_g]$  and an auxiliary vector  $q \in \mathbb{R}^{n_g}$ , we may now rewrite (33) in the equivalent form

$$\min_{x, z, \{p_i\}, q} \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z + \sum_{i=1}^{n_g} \delta_{\mathcal{K}_i} \left( \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} \right) \quad \text{s.t.} \quad q = z \text{ and } p_i = [x]_{g_i} \text{ for all } i \in [n_g]. \quad (34)$$

If we define dual vectors  $\pi \in \mathbb{R}^{n_g}$  and  $\{y_i\}_{i=1}^{n_g}$  with  $y_i \in \mathbb{R}^{|g_i|}$  for each  $i \in [n_g]$ , then the Lagrange function associated with the optimization problem (34) takes the form

$$\begin{aligned} \mathcal{L}_k(x, z, \{p_i\}, q, \pi, \{y_i\}) & \\ & := \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z + \sum_{i=1}^{n_g} \delta_{\mathcal{K}_i} \left( \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} \right) + (q - z)^T \pi + \sum_{i=1}^{n_g} (p_i - [x]_{g_i})^T y_i. \end{aligned} \quad (35)$$

The dual function is then given by

$$\begin{aligned} \mathcal{D}_k(\pi, \{y_i\}) & := \inf_{x, z, \{p_i\}, q} \mathcal{L}_k(x, z, \{p_i\}, q, \pi, \{y_i\}) \\ & = \inf_{x, z} \left\{ \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z - \sum_{i=1}^{n_g} y_i^T [x]_{g_i} - z^T \pi \right\} - \sup_{\{p_i\}, q} \sum_{i=1}^{n_g} \left( - \begin{bmatrix} y_i \\ [\pi]_i \end{bmatrix}^T \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} - \delta_{\mathcal{K}_i} \left( \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} \right) \right) \\ & = \inf_{x, z} \left\{ \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z - \sum_{i=1}^{n_g} y_i^T [x]_{g_i} - z^T \pi \right\} - \sum_{i=1}^{n_g} \sup_{p_i, [q]_i} \left( - \begin{bmatrix} y_i \\ [\pi]_i \end{bmatrix}^T \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} - \delta_{\mathcal{K}_i} \left( \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} \right) \right) \\ & = \inf_{x, z} \left\{ \frac{1}{2\alpha_k} \|x - u_k\|^2 + \lambda^T z - \sum_{i=1}^{n_g} y_i^T [x]_{g_i} - z^T \pi \right\} - \sum_{i=1}^{n_g} \sup_{p_i, [q]_i} \left\{ - \begin{bmatrix} y_i \\ [\pi]_i \end{bmatrix}^T \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} \mid \begin{bmatrix} p_i \\ [q]_i \end{bmatrix} \in \mathcal{K}_i \right\}. \end{aligned}$$

If, for each  $i \in [n_g]$ , we define  $\bar{y}_i \in \mathbb{R}^n$  so that  $[\bar{y}_i]_{g_i} = y_i$  and all other elements equal to zero, then it follows from the optimality conditions for the infimum problem above and the structure of the second-order cone  $\mathcal{K}_i$  appearing in the supremum that

$$\mathcal{D}_k(\pi, \{y_i\}) = \begin{cases} -\infty & \text{if } \pi \neq \lambda \text{ or } \begin{bmatrix} y_i \\ [\pi]_i \end{bmatrix} \notin \mathcal{K}_i \text{ for any } i \in [n_g], \\ -\frac{\alpha_k}{2} \left\| \sum_{i=1}^{n_g} \bar{y}_i \right\|^2 - u_k^T \left( \sum_{i=1}^{n_g} \bar{y}_i \right) & \text{if } \pi = \lambda \text{ and } \begin{bmatrix} y_i \\ [\pi]_i \end{bmatrix} \in \mathcal{K}_i \text{ for all } i \in [n_g]. \end{cases}$$

Thus, the dual problem obtained by maximizing the function  $\mathcal{D}_k$  can be written as

$$\max_{\{y_i\}} -\frac{\alpha_k}{2} \left\| \sum_{i=1}^{n_g} \bar{y}_i \right\|^2 - u_k^T \left( \sum_{i=1}^{n_g} \bar{y}_i \right) \quad \text{s.t.} \quad \begin{bmatrix} y_i \\ [\lambda]_i \end{bmatrix} \in \mathcal{K}_i \text{ for each } i \in [n_g]. \quad (36)$$

Next, note that if we define the matrix  $A \in \mathbb{R}^{n \times \sum_{i=1}^{n_g} |g_i|}$  such that

$$[A]_{(g_i, j), j + \sum_{\ell=1}^{i-1} |g_\ell|} = 1 \quad \text{for all } i \in [n_g] \text{ and } j \in [|g_i|] \quad (37)$$

and all other entries of  $A$  are set equal to zero, then it follows that

$$A\hat{y} = \sum_{i=1}^{n_g} \bar{y}_i \quad \text{where } \hat{y} := (y_1^T, y_2^T, \dots, y_{n_g}^T)^T \in \mathbb{R}^{\sum_{i=1}^{n_g} |g_i|}. \quad (38)$$

Introducing the set-valued mapping  $\mathcal{M} : \{1, 2, \dots, n_g\} \rightrightarrows \{1, 2, \dots, \sum_{i=1}^{n_g} |g_i|\}$  so that

$$\mathcal{M}(i) = \left\{ \sum_{l=1}^{i-1} |g_l| + 1, \sum_{l=1}^{i-1} |g_l| + 2, \dots, \sum_{l=1}^{i-1} |g_l| + |g_i| \right\}, \quad (39)$$

it follows that  $y_i = [\hat{y}]_{\mathcal{M}(i)}$  for all  $i \in [n_g]$ . Using (38) and (36) yields the dual problem

$$\hat{\mathcal{Y}}(x_k, \alpha_k) := \underset{\hat{y} \in \mathcal{F}_d}{\text{Argmax}} \phi_d(\hat{y}; x_k, \alpha_k) \quad \text{with} \quad \phi_d(\hat{y}; x_k, \alpha_k) := -\frac{\alpha_k}{2} \|A\hat{y}\|^2 - u_k^T A\hat{y} \quad (40)$$

where we recall that  $u_k = x_k - \alpha_k \nabla f(x_k)$  and we define the dual feasibility set

$$\mathcal{F}_d := \{\hat{y} \in \mathbb{R}^{\sum_{i=1}^{n_g} |g_i|} \mid \|[\hat{y}]_{\mathcal{M}(i)}\| \leq [\lambda]_i \text{ for each } i \in [n_g]\}. \quad (41)$$

We use capital ‘‘Argmax’’ in (40) to emphasize that  $\hat{\mathcal{Y}}(x_k, \alpha_k)$  is a set. Note that strong duality holds for the primal problem (31) and dual problem (40) since Slater’s condition holds for problem (40) (since the components of  $\lambda$  are all *strictly* positive).

We now establish results related to the dual solution set  $\hat{\mathcal{Y}}(x_k, \alpha_k)$  of problem (40). The first lemma establishes an important equation (the linking equation) that connects a dual solution to problem (40) and the primal solution  $x_k^*$  to problem (31).

**Lemma 4.2.** *The following results hold for the set of dual solutions.*

(i) *The set of dual solutions  $\hat{\mathcal{Y}}(x_k, \alpha_k)$  for problem (40) satisfies*

$$\hat{\mathcal{Y}}(x_k, \alpha_k) = \{\hat{y} \in \mathcal{F}_d \mid A\hat{y} = (x_k^* - u_k)/\alpha_k\}, \quad (42)$$

*where  $A$  is defined in (37) and  $x_k^*$  is the primal solution defined in (32).*

(ii) *The solution  $x_k^*$  to problem (32) satisfies  $x_k^* = u_k + \alpha_k A\hat{y}_k^*$  for all  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ .*

(iii) If  $i \in [n_g]$  and  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$  satisfying  $\|[\hat{y}_k^*]_{\mathcal{M}(i)}\| < [\lambda]_i$ , then  $[x_k^*]_{g_i} = 0$ .

*Proof.* We begin with part (i). To prove (42) we first show that  $\hat{\mathcal{Y}}(x_k, \alpha_k) \subseteq \{\hat{y} \in \mathcal{F}_d \mid A\hat{y} = (x_k^* - u_k)/\alpha_k\}$ . To this end, let  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ , and thus  $\hat{y}_k^* \in \mathcal{F}_d$  (see (40)). Since  $x_k^*$  is the unique solution to the primal problem, we know that  $(x_k^*, \hat{y}_k^*)$  is a primal-dual solution pair. Therefore, it follows from first-order necessary optimality conditions obtained by setting the derivative (with respect to  $x$ ) of the Lagrangian (35) to zero that  $x_k^* = u_k + \alpha_k A\hat{y}_k^*$ , where  $A$  is defined in (37). Since this equality and  $\hat{y}_k^* \in \mathcal{F}_d$  imply that  $\hat{y}_k^*$  belongs to the set in the right-hand side of (42), we have proved the first inclusion. Now, to establish the inclusion  $\hat{\mathcal{Y}}(x_k, \alpha_k) \supseteq \{\hat{y} \in \mathcal{F}_d \mid A\hat{y} = (x_k^* - u_k)/\alpha_k\}$ , let  $\hat{y}_k^*$  satisfies  $\hat{y}_k^* \in \mathcal{F}_d$  and  $A\hat{y}_k^* = (x_k^* - u_k)/\alpha_k$ . Since  $x_k^*$  is the unique solution to the primal problem and strong duality holds for the primal-dual pair, we know that there must exist  $\hat{y}_{\text{sol}} \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ . Combining this with the inclusion that we already proved shows that  $\hat{y}_{\text{sol}}$  must satisfy  $A\hat{y}_{\text{sol}} = (x_k^* - u_k)/\alpha_k$ . Combining this with the definition of the dual function  $\phi_d(\cdot; x_k, \alpha_k)$  and  $A\hat{y}_k^* = (x_k^* - u_k)/\alpha_k$  shows that

$$\begin{aligned} \phi_d(\hat{y}_k^*; x_k, \alpha_k) &= -\frac{\alpha_k}{2} \|A\hat{y}_k^*\|^2 - u_k^T A\hat{y}_k^* \\ &= -\frac{\alpha_k}{2} \|(x_k^* - u_k)/\alpha_k\|^2 - u_k^T (x_k^* - u_k)/\alpha_k \\ &= -\frac{\alpha_k}{2} \|A\hat{y}_{\text{sol}}\|^2 - u_k^T A\hat{y}_{\text{sol}} = \phi_d(\hat{y}_{\text{sol}}; x_k, \alpha_k). \end{aligned}$$

This equation and  $\hat{y}_k^* \in \mathcal{F}_d$  imply that  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ , thus completing the proof of this inclusion. Since we have established both inclusions, we know that (42) holds.

For (ii), let  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ . Then, (i) implies  $A\hat{y}_k^* = (x_k^* - u_k)/\alpha_k$  so that (ii) holds.

For (iii), let  $i \in [n_g]$  and  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$  satisfy  $\|[\hat{y}_k^*]_{\mathcal{M}(i)}\| < [\lambda]_i$ . From  $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ , first-order optimality conditions for (40), and  $\|[\hat{y}_k^*]_{\mathcal{M}(i)}\| < [\lambda]_i$  we have

$$\nabla_{[\hat{y}]_{\mathcal{M}(i)}} \phi_d(\hat{y}_k^*; x_k, \alpha_k) = 0 \quad \Rightarrow \quad [A^T(\alpha_k A\hat{y}_k^* + u_k)]_{\mathcal{M}(i)} = 0. \quad (43)$$

The definition of  $A$  gives  $A_{g_i, \mathcal{M}(i)} = I$  and  $A_{g_i^c, \mathcal{M}(i)} = 0$ , where  $I$  is the identify matrix of size  $|g_i|$  and  $g_i^c$  is the complement of  $g_i$ . Combining these facts with (43) gives

$$0 = [A^T(\alpha_k A\hat{y}_k^* + u_k)]_{\mathcal{M}(i)} = [A^T]_{\mathcal{M}(i), g_i} [\alpha_k A\hat{y}_k^* + u_k]_{g_i} = [\alpha_k A\hat{y}_k^* + u_k]_{g_i}.$$

Combining this equation with  $x_k^* = u_k + \alpha_k A\hat{y}_k^*$  implies that  $[x_k^*]_{g_i} = 0$ , as claimed.  $\square$

We now bound the distance between feasible points and the solution set  $\hat{\mathcal{Y}}(x_k, \alpha_k)$ .

**Lemma 4.3.** *The set  $\hat{\mathcal{Y}}(x_k, \alpha_k)$  is compact and convex, so that in particular the projection operator  $\text{Proj}_{\hat{\mathcal{Y}}(x_k, \alpha_k)}(\cdot)$  and  $\text{dist}(\cdot, \hat{\mathcal{Y}}(x_k, \alpha_k))$  are well defined. Moreover, there exists  $\nu_k > 0$  and  $\rho_k > 0$  such that*

$$\text{dist}(\hat{y}, \hat{\mathcal{Y}}(x_k, \alpha_k)) \leq \nu_k \|A\hat{y} - Ay\|^{\rho_k} \quad \text{for all } (\hat{y}, y) \in (\mathcal{F}_d, \hat{\mathcal{Y}}(x_k, \alpha_k)).$$

*Proof.* From the definition of  $\hat{\mathcal{Y}}(x_k, \alpha_k)$  in (42) it is clear that  $\hat{\mathcal{Y}}(x_k, \alpha_k)$  is compact and convex, which proves the first part of the lemma. Next, by applying [16, Theorem 2.2] with  $X = \mathbb{R}^{\sum_{i=1}^{n_g} |g_i|}$ ,  $\bar{X} = \mathcal{F}_d$ ,  $f_i(\hat{y}) = \|[\hat{y}]_{\mathcal{M}(i)}\|^2 - [\lambda]_i^2$  for all  $i \in [n_g]$ ,  $g(\hat{y}) = A\hat{y} - (x_k^* - u_k)/\alpha_k$ , it follows that there exists  $\nu_k > 0$  and  $\rho_k > 0$  such that

$$\text{dist}(\hat{y}, S_k) \leq \nu_k \|A\hat{y} - (x_k^* - u_k)/\alpha_k\|^{\rho_k} \quad \text{for all } \hat{y} \in \mathcal{F}_d,$$

where  $S_k = \{\hat{y} \in \mathbb{R}^{\sum_{i=1}^{n_g} |g_i|} \mid f_i(\hat{y}) \leq 0 \text{ for all } i \in [n_g] \text{ and } g(\hat{y}) = 0\}$ . Since  $S_k \equiv \hat{\mathcal{Y}}(x_k, \alpha_k)$  (see (42)), one can use the previous equality and Lemma 4.2(ii) to conclude

$$\begin{aligned} \text{dist}(\hat{y}, \hat{\mathcal{Y}}(x_k, \alpha_k)) &\leq \nu_k \|A\hat{y} - (x_k^* - u_k)/\alpha_k\|^{\rho_k} \\ &= \nu_k \|A\hat{y} - A\hat{y}_k^*\|^{\rho_k} \quad \text{for all } (\hat{y}, \hat{y}_k^*) \in (\mathcal{F}_d, \hat{\mathcal{Y}}(x_k, \alpha_k)), \end{aligned}$$

thus completing the proof.  $\square$

## 4.2 A new algorithm for approximately solving the PG subproblem

During the  $k$ th iteration of Algorithm 1, it is necessary to compute  $\hat{x}_{k+1}$  satisfying  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  for a particular value of  $\epsilon_k$ . One way to perform this task is to apply an iterative solver to the dual subproblem (40) while monitoring the primal-dual gap. For example, (40) can be approximately solved by the projected gradient-ascent algorithm with arc search [6, Section 2.3] armed with an appropriate early termination test. Unfortunately, the straightforward application of such a method to the *dual* problem is insufficient for discovering the support of an optimal solution to the *primal* problem. Therefore, in this section we present Algorithm 2, which is an enhanced projected gradient-ascent algorithm that we prove terminates with an  $x_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  and (asymptotically) to correctly identify the support of an optimal solution to problem (1) under a nondegeneracy assumption.

Since we propose to use Algorithm 2 as the subproblem solver to be used during the  $k$ th iteration of Algorithm 1, we denote its primal and dual iterate sequences by  $\{\hat{x}_{k,t}, \hat{y}_{k,t}\}_{t \geq 0}$  where  $t$  is the iteration counter of the subproblem solver. Given the  $t$ th dual iterate  $\hat{y}_{k,t}$ , motivated by Lemma 4.2 we compute a group index set  $\mathcal{P}_{k,t}$ , a primal trial iterate  $x_{k,t}$ , and a projected primal trial iterate  $\hat{x}_{k,t}$  in Line 6 and Line 7. The group index set  $\mathcal{P}_{k,t}$  holds the groups *predicted* to be zero at a primal solution, and  $\hat{x}_{k,t}$  is constructed by zeroing out all groups in  $\mathcal{P}_{k,t}$  so that  $\hat{x}_{k,t}$  is at least as sparse as  $x_{k,t}$ . We draw your attention to the usage of  $\epsilon_{k-1}^l$  appearing in the definition of  $\mathcal{P}_{k,t}$  in Line 6, which is critical to ensuring that Algorithm 2 is well posed and that our ultimate complexity result for support identification holds. Lines 8–16 check for termination of the subproblem solver, the conditions of which are chosen to ensure that  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  anytime termination occurs (see Lemma 4.4 below and Lines 6 and 9 in Algorithm 1). If termination does not occur during the  $t$ th iteration, a standard projected gradient-ascent search is performed from Lines 17–25 to compute  $\hat{y}_{k,t+1}$ . (There is nothing special about us using the projected gradient-ascent search; we simply need the sequence  $\{\hat{y}_{k,t}\}_{t \geq 0}$  to converge to the solution set  $\hat{\mathcal{Y}}(x_k, \alpha_k)$ .)

We conclude this section by proving that Algorithm 2 is well posed.

**Lemma 4.4.** *For each  $k \in \mathbb{N}$ , if  $x_k$  does not satisfy the first-order optimality conditions for problem (1) with regularizer given by (30), then Algorithm 2 terminates finitely with  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  with  $\epsilon_k$  defined in Line 6 of Algorithm 1 if `option=option_1` or with  $\epsilon_k$  defined in Line 9 of Algorithm 1 if `option=option_2`.*

*Proof.* For a proof by contradiction, assume that Algorithm 2 never terminates, and thus it generates an infinite subsequence of iterates  $\{\hat{y}_{k,t}\}_{t \geq 0}$  that is equivalent to the sequence generated by the projected gradient-ascent with arc search algorithm by [6, Section 2.3]. It is well known (e.g., see [11]) that such an algorithm when applied to a concave optimization problem will satisfy  $\lim_{t \rightarrow \infty} \phi_d(\hat{y}_{k,t}; x_k, \alpha_k) = \phi_d(\hat{y}_k^*; x_k, \alpha_k)$ , for any  $\hat{y}_k^*$  that is an optimal solution to the dual problem (40). Combining this with the linking equation (see Lemma 4.2(iii)) and Line 7 of Algorithm 2, it follows that  $\lim_{t \rightarrow \infty} \phi(x_{k,t}; x_k, \alpha_k) = \phi(x_k^*; x_k, \alpha_k)$  where  $\hat{x}_k^*$  is the optimal solution to the primal problem (31). Therefore, since strong duality holds for our primal-dual problems (Slater's condition holds for problem (40)), we can conclude that

$$\lim_{t \rightarrow \infty} (\phi(x_{k,t}; x_k, \alpha_k) - \phi_d(\hat{y}_{k,t}; x_k, \alpha_k)) = 0. \quad (44)$$

Next, we claim that there exists a subsequence  $\mathcal{T}$  such that  $\lim_{t \in \mathcal{T}} \hat{x}_{k,t} = x_k^*$ . For a proof by contradiction, suppose that no such subsequence  $\mathcal{T}$  exists. Combining this supposition with  $\lim_{t \rightarrow \infty} x_{k,t} = x_k^*$  (this follows from (44) and the fact that the primal proximal subproblem has a unique solution), we can conclude that there exists an  $i_* \in [n_g]$ ,  $t_* \in \mathbb{N}$ , and constants  $\epsilon$  and  $\mathcal{G}$  such that

$$\|[\hat{x}_{k,t} - x_{k,t}]_{g_{i_*}}\| \geq \epsilon > 0 \quad \text{and} \quad \|[x_{k,t} - x_k^*]_{g_{i_*}}\| < \epsilon. \quad \text{for all } t \geq t_*. \quad (45)$$

The first inequality in (45) together with the definition of  $\hat{x}_{k,t}$  in Line 7 shows that  $i_* \in \mathcal{P}_{k,t}$  for all  $t \geq t_*$ . Using this fact, the definition of  $\mathcal{P}_{k,t}$ , and (45) gives

$$i_* \in \mathcal{P}_{k,t} \quad \text{and} \quad \|[\hat{y}_{k,t}]_{\mathcal{M}(i_*)}\| < [\lambda]_{i_*} - \epsilon_{k-1}^l \quad \text{for all } t \geq t_*. \quad (46)$$

Using  $\lim_{t \rightarrow \infty} \text{dist}(\hat{y}_{k,t}, \hat{\mathcal{Y}}(x_k, \alpha_k)) = 0$ ,  $\epsilon_{k-1}^l > 0$ , and (46), we know that there exists a  $y_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$  satisfying  $\|[y_k^*]_{\mathcal{M}(i_*)}\| < [\lambda]_{i_*}$ . This fact and Lemma 4.2(iii) it follows that  $[x_k^*]_{g_{i_*}} = 0$ . Using this equality



---

**Algorithm 2** An enhanced projected gradient-ascent method for solving problem (40).

---

```

1: Input: An initial dual solution estimate  $\hat{y}_{k,0} \in \mathbb{R}^{\sum_{i=1}^n |g_i|}$ .
2: Constants:  $\{\eta_2, \xi_2\} \subset (0, 1)$  and  $\{\sigma, \iota\} \subset (0, \infty)$ 
3: Values from Algorithm 1:  $x_k, \alpha_k, c_k, \epsilon_{k-1}$ , option, and  $\gamma_2$ .
4: Set  $t \leftarrow 0$ , define  $A$  as in (37), and set  $u_k \leftarrow x_k - \alpha_k \nabla f(x_k)$ .
5: for  $t = 1, 2, \dots$  do
6:   Set  $\mathcal{P}_{k,t} \leftarrow \{i \in [n_g] \mid \|\hat{y}_{k,t}\|_{\mathcal{M}(i)} < [\lambda]_i - \epsilon_{k-1}^\iota\}$  with  $\mathcal{M}$  defined in (39).
7:   Define the trial primal iterate and trial projected primal iterate as
      
$$x_{k,t} \leftarrow u_k + \alpha_k A \hat{y}_{k,t} \quad \text{and} \quad [\hat{x}_{k,t}]_{g_i} \leftarrow \begin{cases} 0, & \text{if } i \in \mathcal{P}_{k,t}, \\ [x_{k,t}]_{g_i}, & \text{if } i \notin \mathcal{P}_{k,t}, \end{cases} \quad \text{for each } i \in [n_g].$$

8:   if (option = option_1) then
9:     if  $\phi(\hat{x}_{k,t}; x_k, \alpha_k) - \phi_d(\hat{y}_{k,t}; x_k, \alpha_k) \leq c_k \|\hat{x}_{k,t} - x_k\|^2$  then
10:      return  $\hat{y}_{k+1} = \hat{y}_{k,t}$  and  $\hat{x}_{k+1} = \hat{x}_{k,t}$ 
11:    end if
12:   else (option = option_2)
13:     if  $\phi(\hat{x}_{k,t}; x_k, \alpha_k) - \phi_d(\hat{y}_{k,t}; x_k, \alpha_k) \leq \gamma_2(\phi(x_k; x_k, \alpha_k) - \phi_d(\hat{y}_{k,t}; x_k, \alpha_k))$  then
14:      return  $\hat{y}_{k+1} = \hat{y}_{k,t}$  and  $\hat{x}_{k+1} = \hat{x}_{k,t}$ 
15:    end if
16:   end if
17:    $j \leftarrow 0$ .
18:   loop
19:     Set the trial step length  $\sigma_{k,t} \leftarrow \sigma \xi_2^j$ .
20:     Set the trial iterate  $\hat{y}_{k,t+1} \leftarrow \text{Proj}_{\mathcal{F}_d}(\hat{y}_{k,t} + \sigma_{k,t} \nabla \phi_d(\hat{y}_{k,t}; x_k, \alpha_k))$ .
21:     if  $\phi_d(\hat{y}_{k,t+1}; x_k, \alpha_k) \geq \phi_d(\hat{y}_{k,t}; x_k, \alpha_k) + \eta_2 \nabla \phi_d(\hat{y}_{k,t}; x_k, \alpha_k)^T (\hat{y}_{k,t+1} - \hat{y}_{k,t})$  then
22:       break the inner loop
23:     end if
24:      $j \leftarrow j + 1$ .
25:   end loop
26: end for

```

---

with the triangle inequality,  $i_* \in \mathcal{P}_{k,t}$  for all  $t \geq t_*$  (see (46)) meaning that  $[\hat{x}_{k,t}]_{g_{i_*}} = [x_k^*]_{g_{i_*}}$ , and the second inequality in (45) yields

$$\|[\hat{x}_{k,t} - x_{k,t}]_{g_{i_*}}\| \leq \|[\hat{x}_{k,t} - x_k^*]_{g_{i_*}}\| + \|[x_k^* - x_{k,t}]_{g_{i_*}}\| = \|[x_k^* - x_{k,t}]_{g_{i_*}}\| < \epsilon \quad \text{for all } t \geq t_*,$$

which contradicts the first inequality in (45). Therefore, our claim must be true, namely that there exists a subsequence  $\mathcal{T}$  such that  $\lim_{t \in \mathcal{T}} \hat{x}_{k,t} = x_k^*$ .

Using  $\lim_{t \in \mathcal{T}} \hat{x}_{k,t} = x_k^*$ , we find that

$$\lim_{t \in \mathcal{T}} (\phi(\hat{x}_{k,t}; x_k, \alpha_k) - \phi_d(\hat{y}_{k,t+1}; x_k, \alpha_k)) = 0. \quad (47)$$

Moreover, since by assumption  $x_k$  does not satisfy the first-order optimality conditions for problem (1), we know that there exists a constant  $\delta > 0$  such that  $\min\{c_k \|\hat{x}_{k,t} - x_k\|^2, \gamma_2(\phi(x_k; x_k, \alpha_k) - \phi_d(\hat{y}_{k,t+1}; x_k, \alpha_k))\} \geq \delta > 0$  for all sufficiently large  $t \in \mathcal{T}$ . This fact may be combined with (47) to conclude that the conditions in Lines 9 and 13 will both hold for all sufficiently large  $t \in \mathcal{T}$ , therefore proving that Algorithm 2 will finitely terminate (regardless of the value of **option**), which contradicts our original supposition, and thus Algorithm 2 finitely terminates.

Next, suppose that **option** = **option\_1**. In this case, Algorithm 2 will finitely terminate in Line 9 and

return the vector  $\hat{x}_{k+1}$  satisfying

$$\phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k) \leq \phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k) \leq c_k \|\hat{x}_{k+1} - x_k\|^2.$$

This inequality and (5) show that  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  with  $\epsilon_k = c_k \|\hat{x}_{k+1} - x_k\|^2$  as defined in Line 6 of Algorithm 1, which completes the proof for this case.

Finally, suppose that `option` = `option.2`. In this case, Algorithm 2 will finitely terminate in Line 13 and return the vector  $\hat{x}_{k+1}$  satisfying

$$\phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k) \leq \gamma_2(\phi(x_k; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k)).$$

Rearranging terms and using  $\gamma_2 \in (0, 1/2]$  and weak duality gives

$$\phi(\hat{x}_{k+1}; x_k, \alpha_k) - \gamma_2\phi(x_k; x_k, \alpha_k) \leq (1 - \gamma_2)\phi_d(\hat{y}_{k+1}; x_k, \alpha_k) \leq (1 - \gamma_2)\phi(T(x_k, \alpha_k); x_k, \alpha_k).$$

Thus,  $\phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k) \leq \gamma_2(\phi(x_k; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k))$ , so that  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  with  $\epsilon_k = \gamma_2(\phi(x_k; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k))$  as defined in Line 9 of Algorithm 1, which completes the proof for this case.  $\square$

### 4.3 Support Identification

In this section, we prove that Algorithm 1, when paired with Algorithm 2 as its subproblem solver, has a finite support identification property. This result is proved under the following two assumptions.

**Assumption 4.5.** The function  $f$  is strongly convex with strong convexity parameter  $\mu_f > 0$ . Thus, problem (1) with regularizer defined by (30) has a unique minimizer  $x^*$ .

**Assumption 4.6.** The parameter  $\gamma_1 \in (0, 2)$  from Algorithm 1 is chosen to satisfy  $3\gamma_1/(1 + \gamma_1) \leq 1$ , and the initial PG parameter is then chosen to satisfy

$$\alpha_0 \in \begin{cases} \left(0, \frac{3\gamma_1(1-\eta)}{L_g(1+\gamma_1)}\right] & \text{if option} = \text{option.1,} \\ \left(0, \frac{1-\eta}{L_g}\right] & \text{if option} = \text{option.2.} \end{cases}$$

Assumption 4.6 is made to simplify the presentation, since it ensures that the sequence  $\{\alpha_k\}$  is constant and smaller than  $1/L_g$  (regardless of the value of `option`) as shown in Lemma 4.7. Otherwise, the analysis below holds for all iterations beyond some fixed iteration, as a consequence of Lemma 3.6.

The following limits hold as a consequence of Assumption 4.5 and Assumption 4.6.

**Lemma 4.7.** *Under Assumption 4.5 and Assumption 4.6, with  $\alpha^* := \alpha_0$ , it holds that*

$$\lim_{k \rightarrow \infty} \epsilon_k = 0, \quad \lim_{k \rightarrow \infty} \|s_k\| = 0, \quad \lim_{k \rightarrow \infty} x_k = x^*, \quad \lim_{k \rightarrow \infty} x_k^* = x^*, \quad \text{and} \\ \alpha_k = \alpha^* = \alpha_0 \in (0, (1 - \eta)/L_g] \text{ for all } k \in \mathbb{N}.$$

*Proof.* The fact that  $\alpha_k = \alpha^* = \alpha_0$  for all  $k \in \mathbb{N}$  follows from Assumption 4.6, the definition of  $\alpha^*$ , and the proof of Lemma 3.6. The inclusion  $\alpha_0 \in (0, (1 - \eta)/L_g]$  follows from Assumption 4.6. Next, Lemma 3.5(i) and Lemma 3.6 imply that  $f(x_{k+1}) + r(x_{k+1}) \leq f(x_k) + r(x_k) + \eta\Delta_k \leq f(x_k) + r(x_k) - \eta\beta\alpha^*\chi_k^2$  for all  $k \in \mathbb{N}$ . These inequalities and the fact that  $\{f(x_k) + r(x_k)\}$  is bounded below because of Assumption 4.5, show that  $\{\chi_k\} \rightarrow 0$  and  $\{\Delta_k\} \rightarrow 0$ . Using these limits and Lemma 3.6, it follows from (15) when `option` = `option.1` and from (19) when `option` = `option.2` that  $0 = \lim_{k \rightarrow \infty} \epsilon_k = \lim_{k \rightarrow \infty} \|s_k\|$ , which are the first two limits we aimed to prove. Next, it follows from  $\{\chi_k\} \rightarrow 0$ , the fact that  $\chi_k$  is a first-order optimality measure for (1) (see [3, Theorem 10.7]), and Assumption 4.5 that  $\lim_{k \rightarrow \infty} x_k = x^*$ . Next, combining  $\{\epsilon_k\} \rightarrow 0$ , Lemma 3.2, and Lemma 3.6 proves that  $\{\|x_k^* - x_k\|\} \rightarrow 0$ . Combining this limit with  $\{x_k\} \rightarrow x^*$  and  $\|x_k^* - x^*\| \leq \|x_k^* - x_k\| + \|x_k - x^*\|$  gives  $\lim_{k \rightarrow \infty} x_k^* = x^*$ .  $\square$

We now show that the dual solutions to problem (1) are given by  $\hat{\mathcal{Y}}(x^*, \alpha^*)$ .

**Lemma 4.8.** *Under Assumption 4.5, the solution set to the dual of problem (1) is*

$$\hat{\mathcal{Y}}^{\text{main}} := \max_{\hat{y} \in \mathcal{F}_d} \phi_d^{\text{main}}(\hat{y}) \quad \text{with} \quad \phi_d^{\text{main}}(\hat{y}) := \inf_{x \in \mathbb{R}^n} (f(x) - x^T A \hat{y}). \quad (48)$$

Also, if Assumption 4.6 holds, then the dual solutions satisfy  $\hat{\mathcal{Y}}^{\text{main}} \equiv \hat{\mathcal{Y}}(x^*, \alpha^*)$ .

*Proof.* Following a similar derivation as that which led to the dual (40) for the PG subproblem (31), one can see that the dual problem for (1) can be written as (48).

Before proving  $\hat{\mathcal{Y}}^{\text{main}} \equiv \hat{\mathcal{Y}}(x^*, \alpha^*)$ , we consider a shifted version of the dual (40) with  $(x_k, \alpha_k) = (x^*, \alpha^*)$  and  $u^* = x^* - \alpha^* \nabla f(x^*)$  where  $\alpha^*$  is defined in Lemma 4.7:

$$\max_{\hat{y} \in \mathcal{F}_d} \phi_d^{\text{shift}}(\hat{y}) \quad (49)$$

where

$$\begin{aligned} \phi_d^{\text{shift}}(\hat{y}) &:= -\frac{\alpha^*}{2} \|A\hat{y}\|^2 - (u^*)^T A\hat{y} + f(x^*) - \frac{\alpha^*}{2} \|\nabla f(x^*)\|^2 \\ &= \inf_{x \in \mathbb{R}^n} \left( \frac{1}{2\alpha^*} \|x - u^*\|^2 - x^T A\hat{y} + f(x^*) - \frac{\alpha^*}{2} \|\nabla f(x^*)\|^2 \right). \end{aligned}$$

This shifted dual problem serves as a bridge between problems (48) and (40), and has two important properties. First,  $\hat{\mathcal{Y}}(x^*, \alpha^*)$  is the solution set of problem (49) since it is also the solution set to problem (40) (see Lemma 4.2(i)). Second, the following hold:

$$\phi_d^{\text{shift}}(\pi) = f(x^*) - (x^*)^T \nabla f(x^*) \quad \text{and} \quad A\pi = \nabla f(x^*) \quad \text{for all } \pi \in \hat{\mathcal{Y}}(x^*, \alpha^*), \quad (50)$$

which can be seen to hold from the following argument. Let  $\pi \in \hat{\mathcal{Y}}(x^*, \alpha^*)$ . Since  $x^*$  is a solution to problem (1), it is easy to show that  $x^*$  is also a solution to problem (31). Therefore,  $(x^*, \pi)$  is a primal-dual solution pair for the proximal problem and its dual with  $(x_k, \alpha_k) = (x^*, \alpha^*)$ . We can now apply Lemma 4.2(ii) to conclude that  $x^* = u^* + \alpha^* A\pi = x^* - \alpha^* \nabla f(x^*) + \alpha^* A\pi$ , which implies the second equality in (50). It now follows from the second equality in (50) and the definition of  $u^*$  that  $\phi_d^{\text{shift}}(\pi) = -\frac{\alpha^*}{2} \|A\pi\|^2 - (u^*)^T A\pi + f(x^*) - \frac{\alpha^*}{2} \|\nabla f(x^*)\|^2 = -\alpha^* \|\nabla f(x^*)\|^2 - (x^* - \alpha^* \nabla f(x^*))^T \nabla f(x^*) + f(x^*) = f(x^*) - (x^*)^T \nabla f(x^*)$ , which establishes the first equality in (50).

We first prove that  $\hat{\mathcal{Y}}(x^*, \alpha^*) \subseteq \hat{\mathcal{Y}}^{\text{main}}$ . To that end, suppose that  $\hat{y}^* \in \hat{\mathcal{Y}}(x^*, \alpha^*)$  and define (see the function inside the infimum in the definition of  $\phi_d^{\text{shift}}$ ) the vector

$$\bar{x} := \operatorname{argmin}_{x \in \mathbb{R}^n} \frac{1}{2\alpha^*} \|x - u^*\|^2 - x^T A\hat{y}^* + f(x^*) - \frac{\alpha^*}{2} \|\nabla f(x^*)\|^2.$$

First-order optimality conditions for this problem show that  $\frac{1}{\alpha^*}(\bar{x} - x^*) + \nabla f(x^*) = A\hat{y}^*$ , which combined with  $A\hat{y}^* = \nabla f(x^*)$  (see (50)) proves that  $\bar{x} = x^*$ . It is also straightforward to verify using first-order optimality conditions and  $A\hat{y}^* = \nabla f(x^*)$  that  $x^* = \operatorname{argmin}_{x \in \mathbb{R}^n} (f(x) - x^T A\hat{y}^*)$ , so that

$$\phi_d^{\text{main}}(\hat{y}^*) = f(x^*) - (x^*)^T A\hat{y}^* = f(x^*) - (x^*)^T \nabla f(x^*). \quad (51)$$

Next, it follows from the definition of  $u^*$ , Assumption 1.1,  $\alpha^* \in (0, (1 - \eta)/L_g]$  (see Lemma 4.7), and  $\eta \in (0, 1)$  that

$$\begin{aligned} &\frac{1}{2\alpha^*} \|x - u^*\|^2 + f(x^*) - \frac{\alpha^*}{2} \|\nabla f(x^*)\|^2 \\ &= f(x^*) + (x - x^*)^T \nabla f(x^*) + \frac{1}{2\alpha^*} \|x - x^*\|^2 \geq f(x) \quad \text{for all } x \in \mathbb{R}^n. \end{aligned}$$

Subtracting  $x^T A \hat{y}$  from both sides of the previous inequality and then taking the infimum over  $x \in \mathbb{R}^n$ , we find that  $\phi_d^{\text{shift}}(\hat{y}) \geq \phi_d^{\text{main}}(\hat{y})$  for all  $\hat{y} \in \mathcal{F}_d$ . Using this inequality, (51), (50), and  $\hat{y}^* \in \hat{\mathcal{Y}}(x^*, \alpha^*)$  (recall that we earlier commented that  $\hat{\mathcal{Y}}(x^*, \alpha^*)$  is also the solution set to problem (49)) that

$$\phi_d^{\text{main}}(\hat{y}^*) = \phi_d^{\text{shift}}(\hat{y}^*) = f(x^*) - (x^*)^T \nabla f(x^*) \geq \phi_d^{\text{shift}}(\hat{y}) \geq \phi_d^{\text{main}}(\hat{y}) \text{ for all } \hat{y} \in \mathcal{F}_d,$$

which shows  $\hat{y}^* \in \hat{\mathcal{Y}}^{\text{main}}$ , and completes the proof for this inclusion.

Next, we prove the inclusion  $\hat{\mathcal{Y}}^{\text{main}} \subseteq \hat{\mathcal{Y}}(x^*, \alpha^*)$ . Let  $\hat{y}^* \in \hat{\mathcal{Y}}^{\text{main}}$ . If we denote  $x^{\text{main}} := \operatorname{argmin}_{x \in \mathbb{R}^n} (f(x) - x^T A \hat{y}^*)$ , then first-order optimality conditions give

$$\nabla f(x^{\text{main}}) = A \hat{y}^*. \quad (52)$$

Since  $(x^*, \hat{y}^*)$  is an optimal primal-dual solution pair for problem (1) and its dual (48), it follows that  $A \hat{y}^* = \nabla f(x^*)$ . Together with (52) shows that  $\nabla f(x^*) = \nabla f(x^{\text{main}})$ , which combined with  $f$  being strongly convex with parameter  $\mu_f > 0$  (see Assumption 4.5) to obtain  $0 = \|\nabla f(x^*) - \nabla f(x^{\text{main}})\| \geq \mu \|x^* - x^{\text{main}}\|$ , so that  $x^{\text{main}} = x^*$ . Using the definition of  $x^{\text{main}}$ ,  $x^{\text{main}} = x^*$ , and  $A \hat{y}^* = \nabla f(x^*)$ , we have that

$$\phi_d^{\text{main}}(\hat{y}^*) = f(x^{\text{main}}) - (x^{\text{main}})^T A \hat{y}^* = f(x^*) - (x^*)^T \nabla f(x^*). \quad (53)$$

Next, using  $A \hat{y}^* = \nabla f(x^*)$  and the definition of  $u^*$ , it follows that

$$\begin{aligned} \phi_d^{\text{shift}}(\hat{y}^*) &= -\frac{\alpha^*}{2} \|A \hat{y}^*\|^2 - (u^*)^T A \hat{y}^* + f(x^*) - \frac{\alpha^*}{2} \|\nabla f(x^*)\|^2 \\ &= -\alpha^* \|\nabla f(x^*)\|^2 - (x^* - \alpha^* \nabla f(x^*))^T \nabla f(x^*) + f(x^*) \\ &= f(x^*) - (x^*)^T \nabla f(x^*). \end{aligned} \quad (54)$$

Taking any  $\hat{y}' \in \hat{\mathcal{Y}}(x^*, \alpha^*)$ , it follows from (50) that  $\phi_d^{\text{shift}}(\hat{y}') = f(x^*) - (x^*)^T \nabla f(x^*)$ , which together with (53) and (54) implies

$$\phi_d^{\text{shift}}(\hat{y}^*) = \phi_d^{\text{main}}(\hat{y}^*) = \phi_d^{\text{shift}}(\hat{y}') \geq \phi_d^{\text{shift}}(\hat{y}) \text{ for all } \hat{y} \in \mathcal{F}_d.$$

Since this inequality means  $\hat{y}^* \in \hat{\mathcal{Y}}(x^*, \alpha^*)$ , we conclude that  $\hat{\mathcal{Y}}^{\text{main}} \subseteq \hat{\mathcal{Y}}(x^*, \alpha^*)$ .  $\square$

Since we established in Lemma 4.8 that  $\hat{\mathcal{Y}}(x^*, \alpha^*)$  is the set of dual solutions to problem (1), we may now present our non-degeneracy assumption, which uses the support of  $x^*$ . Note that each  $i \in [n_g]$  and  $\hat{y} \in \hat{\mathcal{Y}}(x^*, \alpha^*)$  together satisfy  $\|[\hat{y}]_{\mathcal{M}(i)}\| \leq [\lambda]_i$  by the definition of  $\hat{\mathcal{Y}}(x^*, \alpha^*)$ . Therefore, the following non-degeneracy assumption is a strengthening of this inequality for all groups not in  $\mathcal{S}(x^*)$ .

**Assumption 4.9.** The quantity

$$\delta_{\text{nd}} := \begin{cases} \min_{\hat{y} \in \hat{\mathcal{Y}}(x^*, \alpha^*), i \notin \mathcal{S}(x^*)} ([\lambda]_i - \|[\hat{y}]_{\mathcal{M}(i)}\|) & \text{if } \mathcal{S}(x^*) \subsetneq [n_g], \\ 1 & \text{if } \mathcal{S}(x^*) = [n_g], \end{cases}$$

satisfies  $\delta_{\text{nd}} > 0$ . It follows that  $\delta^* := \min\{1, \delta_{\text{nd}}\} \in (0, 1]$ .

The essence of the previous assumption is that, for each  $i \notin \mathcal{S}(x^*)$ , the set  $\{[\hat{y}]_{\mathcal{M}(i)}\}_{\hat{y} \in \hat{\mathcal{Y}}(x^*, \alpha^*)}$  is bounded away from the boundary of the two-norm ball centered at zero of radius  $[\lambda]_i$ . This non-degeneracy assumption is crucial for proving support identification results, where we note that a similar assumption has been used in [9, 27]. In fact, Assumption 4.9 is an extension of the assumption used by these authors that is applicable to the overlapping group- $\ell_1$  case.

We also need minimal control over the sequences  $\{\nu_k\}$  and  $\{\rho_k\}$  from Lemma 4.3.

**Assumption 4.10.** The sequences  $\{\nu_k\}$  and  $\{\rho_k\}$  defined in Lemma 4.3 can be chosen to satisfy  $\sup_{k \geq 0} \nu_k =: \nu_{\max} < \infty$  and  $\inf_{k \geq 0} \rho_k =: \rho_{\min} > 0$ .

Let us also choose any  $\nu^* \in (0, \infty)$  and  $\rho^* \in (0, \infty)$  satisfying

$$\text{dist}(\hat{y}, \hat{\mathcal{Y}}(x^*, \alpha^*)) \leq \nu^* \|A\hat{y} - Ay\|^{\rho^*} \quad \text{for all } (\hat{y}, y) \in (\mathcal{F}_d, \hat{\mathcal{Y}}(x^*, \alpha^*)), \quad (55)$$

which are guaranteed to exist because of Lemma 4.3 with  $(x_k, \alpha_k) = (x^*, \alpha^*)$ . Using these quantities, we may now bound the distance between any *exact* solution of the dual problem (40) and  $\hat{\mathcal{Y}}(x^*, \alpha^*)$ , and the distance between the *inexact* solution of (40) computed by Algorithm 1 and the set  $\hat{\mathcal{Y}}(x_k, \alpha_k)$ .

**Lemma 4.11.** *Under Assumptions 4.5–4.10 the following hold.*

(i) *With  $\nu^*$  and  $\rho^*$  from (55) and  $\alpha^*$  from Lemma 4.7, we have for each  $k \in \mathbb{N}$  that*

$$\text{dist}(\hat{y}_k^*, \hat{\mathcal{Y}}(x^*, \alpha^*)) \leq \nu^* \left( \frac{\|x_k^* - x^*\|}{\alpha_k} + \left(L_g + \frac{1}{\alpha_k}\right) \|x_k - x^*\| \right)^{\rho^*} \quad \text{for all } \hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k).$$

(ii) *The  $\hat{y}_{k+1}$  returned by Algorithm 2 satisfies*

$$\text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x_k, \alpha_k)) \leq \nu_k \left( \frac{2\epsilon_k}{\alpha_k} \right)^{\rho_k/2}.$$

*Proof.* To prove part (i), let  $\hat{y}_k^* \in \mathcal{Y}(x_k, \alpha_k)$ . We may now define  $y^* = \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_k^*)$ . It now follows from (55) with  $(\hat{y}, y) = (\hat{y}_k^*, y^*)$ , Lemma 4.2(i), the triangle inequality, and Assumption 1.1 that

$$\begin{aligned} & \text{dist}(\hat{y}_k^*, \hat{\mathcal{Y}}(x^*, \alpha^*)) \\ & \leq \nu^* \|A\hat{y}_k^* - Ay^*\|^{\rho^*} = \nu^* \left\| \frac{x_k^* - u_k}{\alpha_k} - \frac{(x^* - u^*)}{\alpha^*} \right\|^{\rho^*} = \nu^* \left\| \frac{x_k^* - x_k}{\alpha_k} + \nabla f(x_k) - \nabla f(x^*) \right\|^{\rho^*} \\ & \leq \nu^* \left( \frac{\|x_k^* - x_k\|}{\alpha_k} + L_g \|x_k - x^*\| \right)^{\rho^*} \leq \nu^* \left( \frac{\|x_k^* - x^*\|}{\alpha_k} + \left(L_g + \frac{1}{\alpha_k}\right) \|x_k - x^*\| \right)^{\rho^*}, \end{aligned}$$

which establishes the claim in part (i) to be true.

We now prove part (ii). If we define  $h_k(z) := \frac{\alpha_k}{2} \|z\|^2 + u_k^T z$ , then it follows that  $\phi_d(\hat{y}; x_k, \alpha_k) = -h_k(A\hat{y})$ . Also, to simplify notation, let us define  $p_{k+1} := \text{Proj}_{\hat{\mathcal{Y}}(x_k, \alpha_k)}(\hat{y}_{k+1})$ . Using this notation, we can observe from the properties that the  $\hat{y}_{k+1}$  returned by Algorithm 2 must satisfy that

$$\phi_d(p_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k) \leq \phi(x_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k) \leq \epsilon_k.$$

Combining this inequality with strong convexity of  $h_k$  and  $p_{k+1} \in \hat{\mathcal{Y}}(x_k, \alpha_k)$  gives

$$\begin{aligned} \epsilon_k & \geq \phi_d(p_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k) = h_k(A\hat{y}_{k+1}) - h_k(Ap_{k+1}) \\ & \geq \nabla h_k(Ap_{k+1})^T A(\hat{y}_{k+1} - p_{k+1}) + \frac{\alpha_k}{2} \|A\hat{y}_{k+1} - Ap_{k+1}\|^2 \\ & \geq -\nabla \phi_d(p_{k+1}; x_k, \alpha_k)^T (\hat{y}_{k+1} - p_{k+1}) + \frac{\alpha_k}{2} \|A\hat{y}_{k+1} - Ap_{k+1}\|^2 \geq \frac{\alpha_k}{2} \|A\hat{y}_{k+1} - Ap_{k+1}\|^2. \end{aligned}$$

The previous inequality, Lemma 4.3 with  $(\hat{y}, y) = (\hat{y}_{k+1}, p_{k+1})$ , and Lemma 3.6 gives

$$\text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x_k, \alpha_k)) \leq \nu_k (\|A\hat{y}_{k+1} - Ap_{k+1}\|^2)^{\rho_k/2} \leq \nu_k \left( \frac{2\epsilon_k}{\alpha_k} \right)^{\rho_k/2},$$

thus completing the proof of part (ii).  $\square$

For the remainder of this section, our analysis applies to two difference scenarios that are defined below based on the rate at which  $\{\epsilon_k\} \rightarrow 0$ . The first scenario uses

$$\theta := (1 - \alpha_0 \mu_f) \in [\eta, 1), \quad (56)$$

with the inclusion following from Lemma 4.7 and  $\mu_f \leq L_g$  since they imply that  $1 > \theta \equiv 1 - \alpha_0 \mu_f \geq 1 - ((1 - \eta)/L_g)L_g = \eta$ . We can now state our two scenarios.

*Strategy 1.* For some  $\psi \in (0, 1)$ , we have  $\epsilon_k \leq \min\{\frac{\alpha_0}{2}, \psi^{2k}\theta^{2(k+1)}\}$  for all  $k \in \mathbb{N}$ .

*Strategy 2.* For some  $\omega \in (0, 1)$ , we have  $\epsilon_{k+1} \leq \omega^2\epsilon_k$  for all  $k \in \mathbb{N}$  with  $\epsilon_0 \leq \alpha_0/2$ . By applying this bound recursively, it follows that  $\epsilon_k \leq \omega^{2k}\epsilon_0 \leq \alpha_0/2$  for all  $k \in \mathbb{N}$ .

We consider both of these scenarios for the following reasons. To implement Strategy 1 the value of  $\theta$  must be known, which means that the strong convexity parameter  $\mu_f$  must be known. (Of course, in practice, one could also attempt to estimate  $\mu_f$ .) If this parameter is known, this strategy may be a good choice. On the other hand, Strategy 2 can be implemented without any knowledge of the strong convexity parameter, which is an advantage, but we shall see that our support identification result depends on the size of  $\omega$  relative to  $\theta$ . Finally, let us note that implementing either strategy requires controlling the size of each element of the sequence  $\{\epsilon_k\}$ . This can be done for each  $k \in \mathbb{N}$  by checking the value of  $\epsilon_k$  returned by Algorithm 2 when called by Algorithm 1. If the value returned is not small enough, then Algorithm 1 could recall Algorithm 2 but this time using as an initial iterate the value returned the previous time it was called. This “recall if necessary” procedure can be repeated as necessary until a sufficiently small  $\epsilon_k$  is obtained as required by the chosen strategy.

We now prove a rate of convergence for both  $x_k^*$  and  $x_{k+1}$  to the solution  $x^*$ .

**Lemma 4.12.** *Let Assumption 1.1 and Assumption 4.5 hold. It follows that*

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + \sqrt{2\alpha_0} \sum_{i=0}^k \frac{\sqrt{\epsilon_i}}{\theta^{i+1}} \right) \text{ for all } k \in \mathbb{N} \quad (57)$$

where  $\theta$  is defined in (56). Consequently, the following rates of convergence hold:

(i) *If the sequence  $\{\epsilon_k\}$  satisfies Strategy 1, then*

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} (\|x_0 - x^*\| + \sqrt{2\alpha_0}/(1 - \psi)) \text{ for all } k \in \mathbb{N} \text{ and} \quad (58)$$

$$\|x_k^* - x^*\| \leq \theta^{k+1} (\|x_0 - x^*\| + \sqrt{2\alpha_0}/(1 - \psi) + \sqrt{2\alpha_0}\psi) \text{ for all } k \in \mathbb{N}. \quad (59)$$

(ii) *If the sequence  $\{\epsilon_k\}$  satisfies Strategy 2, then the following hold:*

(a) *If  $\omega < \theta$ , then*

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + \frac{\sqrt{2\alpha_0\epsilon_0}}{(\theta - \omega)} \right) \text{ for all } k \in \mathbb{N}, \quad (60)$$

$$\|x_k^* - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + \frac{\sqrt{2\alpha_0\epsilon_0}}{(\theta - \omega)} + \frac{\sqrt{2\alpha_0\epsilon_0}}{\omega} \right) \text{ for all } k \in \mathbb{N}. \quad (61)$$

(b) *If  $\omega > \theta$ , then*

$$\|x_{k+1} - x^*\| \leq \omega^{k+1} \left( \|x_0 - x^*\| + \frac{\sqrt{2\alpha_0\epsilon_0}}{\omega - \theta} \right) \text{ for all } k \in \mathbb{N} \text{ and} \quad (62)$$

$$\|x_k^* - x^*\| \leq \omega^{k+1} \left( \|x_0 - x^*\| + \frac{\sqrt{2\alpha_0\epsilon_0}}{\omega - \theta} + \frac{\sqrt{2\alpha_0\epsilon_0}}{\omega} \right) \text{ for all } k \in \mathbb{N}. \quad (63)$$

(c) *If  $\omega = \theta$ , then*

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + (k+1) \frac{\sqrt{2\alpha_0\epsilon_0}}{\theta} \right) \text{ for all } k \in \mathbb{N} \text{ and} \quad (64)$$

$$\|x_k^* - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + (k+1) \frac{\sqrt{2\alpha_0\epsilon_0}}{\theta} + \sqrt{2\alpha_0\epsilon_0}\omega^k \right) \text{ for all } k \in \mathbb{N}. \quad (65)$$

*Proof.* We first prove (57) using a procedure similar to [25, Proposition 3]. It follows from Lemma 4.7, optimality of  $x^*$ , and defining  $p_k = \text{prox}_{\alpha^*r}(x_k - \alpha^*\nabla f(x_k))$  that

$$\|x_{k+1} - x^*\|^2 = \|x_{k+1} - p_k + p_k - \text{prox}_{\alpha^*r}(x^* - \alpha^*\nabla f(x^*))\|^2. \quad (66)$$

This equality, the Cauchy-Schwarz inequality, and non-expansivity of the proximal operator yields

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_{k+1} - p_k\|^2 + \|p_k - \text{prox}_{\alpha^* r}(x^* - \alpha^* \nabla f(x^*))\|^2 \\ &\quad + 2 \|x_{k+1} - p_k\| \|p_k - \text{prox}_{\alpha^* r}(x^* - \alpha^* \nabla f(x^*))\| \\ &\leq \|x_{k+1} - p_k\|^2 + \|x_k - x^* - \alpha^* (\nabla f(x_k) - \nabla f(x^*))\|^2 \\ &\quad + 2 \|x_{k+1} - p_k\| \|x_k - x^* - \alpha^* (\nabla f(x_k) - \nabla f(x^*))\|. \end{aligned}$$

Combining this inequality, Lemma 3.2, and the fact that, for all  $k \in \mathbb{N}$ , we know  $\alpha_k = \alpha^*$  and  $x_{k+1} = \hat{x}_{k+1}$  returned by Algorithm 2 satisfy  $x_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha^*)$  yields

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq 2\alpha^* \epsilon_k + \|x_k - x^* - \alpha^* (\nabla f(x_k) - \nabla f(x^*))\|^2 \\ &\quad + 2\sqrt{2\alpha^* \epsilon_k} \|x_k - x^* - \alpha^* (\nabla f(x_k) - \nabla f(x^*))\|. \end{aligned}$$

To bound the norm that appears in the previous inequality, we may use the Cauchy-Schwarz inequality and [19, Theorem 2.1.12] to obtain

$$\begin{aligned} &\|x_k - x^* - \alpha^* (\nabla f(x_k) - \nabla f(x^*))\|^2 \\ &= \|x_k - x^*\|^2 + (\alpha^*)^2 \|\nabla f(x_k) - \nabla f(x^*)\|^2 - 2\alpha^* (x_k - x^*)^T (\nabla f(x_k) - \nabla f(x^*)) \\ &\leq \|x_k - x^*\|^2 + (\alpha^*)^2 \|\nabla f(x_k) - \nabla f(x^*)\|^2 \\ &\quad - 2\alpha^* \left( \frac{L_g \mu_f}{L_g + \mu_f} \|x_k - x^*\|^2 + \frac{1}{\mu_f + L_g} \|\nabla f(x_k) - \nabla f(x^*)\|^2 \right) \\ &= \left( 1 - \frac{2\alpha^* L_g \mu_f}{L_g + \mu_f} \right) \|x_k - x^*\|^2 + \alpha^* \left( \alpha^* - \frac{2}{L_g + \mu_f} \right) \|\nabla f(x_k) - \nabla f(x^*)\|^2. \end{aligned} \tag{67}$$

From  $\eta \in (0, 1)$  and Lemma 4.7 we have  $\alpha^* \in (0, (1 - \eta)/L_g) \subset (0, 1/L_g)$ . Combining this with  $\mu_f \leq L_g$  we have  $\alpha^* < 2/(L_g + \mu_f)$ , and from strong convexity of  $f$  that  $\|\nabla f(x_k) - \nabla f(x^*)\|^2 \geq \mu_f^2 \|x_k - x^*\|^2$ . Combining these two facts with (67) gives

$$\begin{aligned} &\|x_k - x^* - \alpha^* (\nabla f(x_k) - \nabla f(x^*))\|^2 \\ &\leq \left( 1 - \frac{2\alpha^* L_g \mu_f}{L_g + \mu_f} \right) \|x_k - x^*\|^2 + \alpha^* \mu_f^2 \left( \alpha^* - \frac{2}{L_g + \mu_f} \right) \|x_k - x^*\|^2 \\ &= \left( 1 - \left( \frac{2\alpha^* L_g \mu_f}{L_g + \mu_f} + \frac{2\alpha^* \mu_f^2}{L_g + \mu_f} \right) + (\alpha^*)^2 \mu_f^2 \right) \|x_k - x^*\|^2 \\ &= (1 - \alpha^* \mu_f)^2 \|x_k - x^*\|^2 = \theta^2 \|x_k - x^*\|^2, \end{aligned} \tag{68}$$

where the last equality follows from Lemma 4.7. Combining (66) and (68) gives

$$\|x_{k+1} - x^*\|^2 \leq \theta^2 \|x_k - x^*\|^2 + 2\sqrt{2\alpha^* \epsilon_k} \theta \|x_k - x^*\| + 2\alpha^* \epsilon_k = (\theta \|x_k - x^*\| + \sqrt{2\alpha^* \epsilon_k})^2.$$

Taking the square root of the previous inequality and applying it recursively shows

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \|x_0 - x^*\| + \sum_{i=0}^k \theta^{k-i} \sqrt{2\alpha_i \epsilon_i} \quad \text{for all } k \in \mathbb{N}.$$

If we now use basic algebra and the fact that  $\alpha_i = \alpha_0$  for all  $i \in \mathbb{N}$ , we arrive at

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + \sqrt{2\alpha_0} \sum_{i=0}^k \frac{\sqrt{\epsilon_i}}{\theta^{i+1}} \right) \quad \text{for all } k \in \mathbb{N},$$

which proves that the inequality in (57) holds.

Before proving the remaining results, let us make a few observations. First, since  $\alpha_k = \alpha_0$  for all  $k \in \mathbb{N}$  (see Lemma 4.7) it follows from the construction of Algorithm 1 that  $x_{k+1} = \hat{x}_{k+1}$  for all  $k \in \mathbb{N}$ .

Combining this fact with the triangle inequality, Lemma 3.2 with  $x_{k+1} = \hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  (which holds by construction of Algorithm 1), and the fact that  $\alpha_k = \alpha_0$  for all  $k \in \mathbb{N}$  shows that

$$\|x_k^* - x^*\| \leq \|x_k^* - x_{k+1}\| + \|x_{k+1} - x^*\| \leq \sqrt{2\alpha_0\epsilon_k} + \|x_{k+1} - x^*\|. \quad (69)$$

We now consider specific choices for the sequence  $\{\epsilon_k\}$  to derive the remaining results.

**Part (i).** Let  $\{\epsilon_k\}$  satisfy the bound in Strategy 1. Applying this bound on  $\epsilon_k$  for all  $k \in \mathbb{N}$  to the right-hand side of (57) leads to a geometric sum with factor  $\psi$ , from which (58) follows. Combining (58) with the assumed bound on  $\epsilon_k$  for all  $k \in \mathbb{N}$ , (69), and  $\psi \in (0, 1)$  proves that (59) holds.

**Part (ii).** Let  $\{\epsilon_k\}$  satisfy Strategy 2 so that  $\epsilon_k \leq \omega^{2k}\epsilon_0$ . This fact and (57) gives

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \left( \|x_0 - x^*\| + \frac{\sqrt{2\alpha_0\epsilon_0}}{\theta} \sum_{i=0}^k \left(\frac{\omega}{\theta}\right)^i \right). \quad (70)$$

We can now consider the three sub-parts (a), (b), and (c) in turn.

To prove part (ii)(a), suppose that  $\omega < \theta$ . It follows that  $\sum_{i=0}^k (\omega/\theta)^i \leq \sum_{i=0}^{\infty} (\omega/\theta)^i = \frac{1}{1-\omega/\theta}$ . Combining this inequality with (70) shows that (60) holds. Combining (60) with (69),  $\epsilon_k \leq \omega^{2k}\epsilon_0$ , and  $\omega < \theta$  gives (61), completing this proof.

To prove part (ii)(b), suppose that  $\omega > \theta$ . Let us first observe by using basic algebra and the fact that a geometric series with factor  $\theta/\omega \in (0, 1)$  is finite that

$$\theta^{k+1} \sum_{i=0}^k \left(\frac{\omega}{\theta}\right)^i = \omega^{k+1} \sum_{i=1}^{k+1} \left(\frac{\theta}{\omega}\right)^i = \omega^k \theta \sum_{i=0}^k \left(\frac{\theta}{\omega}\right)^i \leq \omega^k \theta \sum_{i=0}^{\infty} \left(\frac{\theta}{\omega}\right)^i = \frac{\omega^k \theta}{1 - \theta/\omega} = \frac{\omega^{k+1} \theta}{\omega - \theta}.$$

This inequality can be combined with (70) and  $\omega > \theta$  to conclude that

$$\|x_{k+1} - x^*\| \leq \theta^{k+1} \|x_0 - x^*\| + \omega^{k+1} \frac{\sqrt{2\alpha_0\epsilon_0}}{\omega - \theta} \leq \omega^{k+1} \left( \|x_0 - x^*\| + \frac{\sqrt{2\alpha_0\epsilon_0}}{\omega - \theta} \right),$$

thus proving that (62) holds. Then, combining (62) with (69) and  $\epsilon_k \leq \omega^{2k}\epsilon_0$  gives (63).

To prove part (ii)(c), suppose that  $\omega = \theta$ . In this case, we know that  $\sum_{i=0}^k (\omega/\theta)^i = k+1$ . Combining this equality with (70) establishes the inequality in (64). Then, combining (64) with (69) and  $\epsilon_k \leq \omega^{2k}\epsilon_0$  gives (65), thus completing the proof.  $\square$

**Lemma 4.13.** *Let Assumption 4.5 and Assumption 4.9 hold. Then, for all  $k \in \mathbb{N}$ , the vector  $\hat{y}_{k+1}$  returned by Algorithm 2 satisfies the following bounds:*

(i) *If the sequence  $\{\epsilon_k\}$  satisfies Strategy 1, then*

$$\begin{aligned} \text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x^*, \alpha^*)) &\equiv \|\hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1})\| \\ &= \mathcal{O}([\theta^{\min\{\rho_{\min}, \rho^*\}}]^k) \quad \text{for all } k \in \mathbb{N}. \end{aligned}$$

(ii) *If the sequence  $\{\epsilon_k\}$  satisfies Strategy 2, then*

$$\begin{aligned} \text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x^*, \alpha^*)) &\equiv \left\| \hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \\ &= \begin{cases} \mathcal{O}([\max\{\omega^{\rho_{\min}}, \theta^{\rho^*}\}]^k) & \text{if } \omega < \theta, \\ \mathcal{O}([\omega^{\min\{\rho_{\min}, \rho^*\}}]^k) & \text{if } \omega > \theta, \\ \mathcal{O}((k\theta^k)^{\min\{\rho_{\min}, \rho^*\}}) & \text{if } \omega = \theta. \end{cases} \end{aligned}$$



*Proof.* Let us define  $p_{k+1} = \text{Proj}_{\hat{\mathcal{Y}}(x_k, \alpha_k)}(\hat{y}_{k+1})$  and  $p_{k+1}^* = \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(p_{k+1})$ . The triangle inequality and non-expansiveness of the projection operator shows that

$$\begin{aligned} & \left\| \hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \\ &= \left\| \hat{y}_{k+1} - p_{k+1} + p_{k+1} - p_{k+1}^* + p_{k+1}^* - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \\ &\leq \left\| \hat{y}_{k+1} - p_{k+1} \right\| + \left\| p_{k+1} - p_{k+1}^* \right\| + \left\| p_{k+1}^* - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \\ &\leq 2 \left\| \hat{y}_{k+1} - p_{k+1} \right\| + \left\| p_{k+1} - p_{k+1}^* \right\| = 2 \text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x_k, \alpha_k)) + \text{dist}(p_{k+1}, \hat{\mathcal{Y}}(x^*, \alpha^*)). \end{aligned}$$

This inequality, Lemma 4.11, and  $\alpha_k = \alpha^*$  for all  $k \in \mathbb{N}$  (see Lemma 4.7) show that

$$\left\| \hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \leq 2\nu_k \left( \frac{2\epsilon_k}{\alpha^*} \right)^{\frac{\rho_k}{2}} + \nu^* \left( \frac{\|x_k^* - x^*\|}{\alpha^*} + (L_g + \frac{1}{\alpha^*}) \|x_k - x^*\| \right)^{\rho^*}. \quad (71)$$

Note that regardless of which case in the statement of the lemma we are considering (i.e., Strategy 1 for part (i) or Strategy 2 for part (ii)), we know that  $\epsilon_k \leq \alpha_0/2 \equiv \alpha^*/2$  for all  $k \in \mathbb{N}$ . This bound, (71), and Assumption 4.10 together show that

$$\left\| \hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \leq 2\nu_{\max} \left( \frac{2\epsilon_k}{\alpha^*} \right)^{\frac{\rho_{\min}}{2}} + \nu^* \left( \frac{\|x_k^* - x^*\|}{\alpha^*} + (L_g + \frac{1}{\alpha^*}) \|x_k - x^*\| \right)^{\rho^*}. \quad (72)$$

We can now consider the two cases in the statement of the lemma.

For part (i),  $\{\epsilon_k\}$  satisfies Strategy 1 so that  $\epsilon_k \leq \min\{\alpha_0/2, \psi^{2k}\theta^{2(k+1)}\}$  for all  $k \in \mathbb{N}$ . Combining this bound, (72), Lemma 4.12(i),  $\theta \in [\eta, 1)$ , and  $\psi \in (0, 1)$  gives

$$\begin{aligned} \left\| \hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| &= \mathcal{O}[(\psi^k \theta^{k+1})^{\rho_{\min}}] + [\mathcal{O}(\theta^{k+1}) + \mathcal{O}(\theta^k)]^{\rho^*} \\ &= \mathcal{O}([\theta^{\rho_{\min}}]^k) + \mathcal{O}([\theta^{\rho^*}]^k) = \mathcal{O}([\theta^{\min\{\rho_{\min}, \rho^*\}}]^k), \end{aligned}$$

which completes the proof for part (i) of this lemma.

For part (ii),  $\{\epsilon_k\}$  satisfies Strategy 2 so that  $\epsilon_k \leq \omega^{2k}\epsilon_0$  for all  $k \in \mathbb{N}$ . This bound, (72), and Lemma 4.12 yield

$$\begin{aligned} & \left\| \hat{y}_{k+1} - \text{Proj}_{\hat{\mathcal{Y}}(x^*, \alpha^*)}(\hat{y}_{k+1}) \right\| \\ &= \begin{cases} \mathcal{O}[(\omega^k)^{\rho_{\min}}] + [\mathcal{O}(\theta^{k+1}) + \mathcal{O}(\theta^k)]^{\rho^*} & \text{if } \omega < \theta, \\ \mathcal{O}[(\omega^k)^{\rho_{\min}}] + [\mathcal{O}(\omega^{k+1}) + \mathcal{O}(\omega^k)]^{\rho^*} & \text{if } \omega > \theta, \\ \mathcal{O}[(\omega^k)^{\rho_{\min}}] + [\mathcal{O}((k+1)\theta^{k+1}) + \mathcal{O}(k\theta^k)]^{\rho^*} & \text{if } \omega = \theta, \end{cases} \\ &= \begin{cases} \mathcal{O}([\max\{\omega^{\rho_{\min}}, \theta^{\rho^*}\}]^k) & \text{if } \omega < \theta, \\ \mathcal{O}([\omega^{\min\{\rho_{\min}, \rho^*\}}]^k) & \text{if } \omega > \theta, \\ \mathcal{O}([(k\theta^k)^{\min\{\rho_{\min}, \rho^*\}}]) & \text{if } \omega = \theta, \end{cases} \end{aligned}$$

which completes the proof for part (ii) of this lemma.  $\square$

We are now ready to present our main support identification theorem.

**Theorem 4.14.** *Let Assumption 4.5–Assumption 4.10 hold, and define*

$$\Theta := \begin{cases} \min\{1, \min_{i \in \mathcal{S}(x^*)} \|[x^*]_{g_i}\|\} & \text{if } \mathcal{S}(x^*) \neq \emptyset, \\ 1 & \text{otherwise.} \end{cases}$$

*Then, the following results hold:*

(i) If  $\{\epsilon_k\}$  satisfies Strategy 1, then  $\mathcal{S}(x_{k+1}) = \mathcal{S}(x^*)$  for all  $k \geq K_1$  with

$$K_1 := \max \left\{ \mathcal{O} \left( \frac{\log \Theta}{\log \theta} \right), \mathcal{O} \left( \frac{\log \delta^*}{\log \max \{ [\theta^{\min\{\rho_{\min}, \rho^*}] \}, [\Psi\theta]^{2\iota} \}} \right) \right\}.$$

(ii) If  $\{\epsilon_k\}$  satisfies Strategy 2, then  $\mathcal{S}(x_{k+1}) = \mathcal{S}(x^*)$  for all  $k \geq K_2$  with

$$K_2 := \begin{cases} \max \left( \mathcal{O} \left( \frac{\log \Theta}{\log \theta} \right), \mathcal{O} \left( \frac{\log \delta^*}{\log(\max\{\omega^{\rho_{\min}}, \theta^{\rho^*}, \omega^{2\iota}\})} \right) \right) & \text{if } \omega < \theta, \\ \max \left( \mathcal{O} \left( \frac{\log \Theta}{\log \omega} \right), \mathcal{O} \left( \frac{\log \delta^*}{\log(\max\{\omega^{\min\{\rho_{\min}, \rho^*\}}, \omega^{2\iota}\})} \right) \right) & \text{if } \omega > \theta, \\ \max(\mathcal{O}(C_\Theta), \mathcal{O}(C_{\delta^*})) & \text{if } \omega = \theta, \end{cases}$$

$C_\Theta$  is the smallest nonnegative integer such that  $(k+1)\omega^{k+1} < \Theta$  for all  $k \geq C_\Theta$ , and  $C_{\delta^*}$  is the smallest nonnegative integer such that  $(k\omega^k)^{\min\{\rho_{\min}, \rho^*\}} + \omega^{2\iota k} < \delta^*$  for all  $k \geq C_{\delta^*}$ .

*Proof.* The proof begins by establishing two claims, which we consider one at a time.

**Claim 1.** We claim that  $\|[\hat{y}_{k+1}]_{\mathcal{M}(i)}\| < [\lambda]_i - \epsilon_{k-1}^\iota$  and  $[x_{k+1}]_{g_i} = 0$  for all  $i \notin \mathcal{S}(x^*)$  and  $k \geq k_1$  with

$$k_1 = \begin{cases} \mathcal{O} \left( \frac{\log \delta^*}{\log \max\{[\theta^{\min\{\rho_{\min}, \rho^*}] \}, [\Psi\theta]^{2\iota}\}} \right) & \text{if Strategy 1 is used,} \\ \mathcal{O} \left( \frac{\log \delta^*}{\log(\max\{\omega^{\rho_{\min}}, \theta^{\rho^*}, \omega^{2\iota}\})} \right) & \text{if Strategy 2 is used and } \omega < \theta, \\ \mathcal{O} \left( \frac{\log \delta^*}{\log(\max\{\omega^{\min\{\rho_{\min}, \rho^*\}}, \omega^{2\iota}\})} \right) & \text{if Strategy 2 is used and } \omega > \theta, \\ \mathcal{O}(C_{\delta^*}) & \text{if Strategy 2 is used and } \omega = \theta. \end{cases} \quad (73)$$

To prove this, note that if  $\mathcal{S}(x^*) = [n_g]$  (i.e., if no  $i \notin \mathcal{S}(x^*)$  exists), then the claim trivially holds for all  $k \geq 0$ , which agrees with the definition of  $k_1$  since  $\delta^* = 1$  and  $C_{\delta^*} = 0$  in this case. Therefore, for the remainder of the proof, we assume that  $\mathcal{S}(x^*) \subsetneq [n_g]$ . It follows from Lemma 4.13 that

$$\begin{aligned} & \text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x^*, \alpha^*)) + \epsilon_{k-1}^\iota \\ &= \begin{cases} \mathcal{O}([\theta^{\min\{\rho_{\min}, \rho^*}]^k) + \mathcal{O}([\Psi\theta]^{2\iota k}) & \text{if Strategy 1 is used,} \\ \mathcal{O}([\max\{\omega^{\rho_{\min}}, \theta^{\rho^*}\}]^k) + \mathcal{O}(\omega^{2\iota k}) & \text{if Strategy 2 is used and } \omega < \theta, \\ \mathcal{O}([\omega^{\min\{\rho_{\min}, \rho^*\}]^k) + \mathcal{O}(\omega^{2\iota k}) & \text{if Strategy 2 is used and } \omega > \theta, \\ \mathcal{O}((k\theta^k)^{\min\{\rho_{\min}, \rho^*\}}) + \mathcal{O}(\omega^{2\iota k}) & \text{if Strategy 2 is used and } \omega = \theta. \end{cases} \end{aligned}$$

Thus, there exists a constant  $k_1$  satisfying (73) that makes the right-hand side of the previous inequality less than  $\delta^*$ , and therefore both  $\epsilon_{k-1}^\iota < \delta^*$  and  $\text{dist}(\hat{y}_{k+1}, \hat{\mathcal{Y}}(x^*, \alpha^*)) < \delta^* - \epsilon_{k-1}^\iota$  must hold for all  $k \geq k_1$ . This second inequality implies that, for each  $k \geq k_1$ , there exists  $p_k^* \in \hat{\mathcal{Y}}(x^*, \alpha^*)$  such that  $\|\hat{y}_{k+1} - p_k^*\| < \delta^* - \epsilon_{k-1}^\iota$ . This inequality and Assumption 4.9 together imply

$$\|[\hat{y}_{k+1}]_{\mathcal{M}(i)}\| \leq \|[\hat{y}_{k+1}]_{\mathcal{M}(i)} - [p_k^*]_{\mathcal{M}(i)}\| + \|[p_k^*]_{\mathcal{M}(i)}\| < \delta^* - \epsilon_k^\iota + [\lambda]_i - \delta^* = [\lambda]_i - \epsilon_{k-1}^\iota$$

for all  $i \notin \mathcal{S}(x^*)$  and  $k \geq k_1$ , which proves the first part of the claim.

To prove the second part of the claim, first note from how  $\hat{y}_{k+1}$  is computed in Algorithm 2 that there exists an integer  $t_k > 0$  such that  $\hat{y}_{k+1} = \hat{y}_{k, t_k}$ . Combining this fact with the first part of the claim that we just proved, we find that  $\|[\hat{y}_{k+1}]_{\mathcal{M}(i)}\| = \|[\hat{y}_{k, t_k}]_{\mathcal{M}(i)}\| < [\lambda]_i - \epsilon_{k-1}^\iota$  for all  $i \notin \mathcal{S}(x^*)$  and  $k \geq k_1$ . Using this inequality and Line 7 of Algorithm 2, it follows that  $[\hat{x}_{k, t_k}]_{g_i} = 0$  for all  $i \notin \mathcal{S}(x^*)$  and  $k \geq k_1$ . Thus,

for all  $i \notin \mathcal{S}(x^*)$  and  $k \geq k_1$ , we have  $[\hat{x}_{k+1}]_{g_i} = [\hat{x}_{k,t_k}]_{g_i} = 0$ . This result and Assumption 4.6 imply that  $[x_{k+1}]_{g_i} = 0$  for all  $i \notin \mathcal{S}(x^*)$  and  $k \geq k_1$  since  $x_{k+1} = \hat{x}_{k+1}$ .

**Claim 2.** We claim that  $[x_{k+1}]_{g_i} \neq 0$  for all  $i \in \mathcal{S}(x^*)$  and  $k \geq k_2$  with

$$k_2 = \begin{cases} \mathcal{O}\left(\frac{\log \Theta}{\log \theta}\right) & \text{if (Strategy 1 is used) or (Strategy 2 is used and } \omega < \theta), \\ \mathcal{O}\left(\frac{\log \Theta}{\log \omega}\right) & \text{if Strategy 2 is used and } \omega > \theta, \\ \mathcal{O}(C_\Theta) & \text{if Strategy 2 is used and } \omega = \theta. \end{cases} \quad (74)$$

To prove this, note that if  $\mathcal{S}(x^*) = \emptyset$ , then the claim trivially holds for all  $k \geq 0$ , which agrees with the definition of  $k_2$  since  $\Theta = 1$  and  $C_\Theta = 0$  in this case. Thus, for the remainder of the proof, we assume that  $\mathcal{S}(x^*) \neq \emptyset$ . It follows from Lemma 4.12 that

$$\begin{aligned} & \|x_{k+1} - x^*\| \\ &= \begin{cases} \mathcal{O}(\theta^{k+1}) & \text{if (Strategy 1 is used) or (Strategy 2 is used and } \omega < \theta), \\ \mathcal{O}(\omega^{k+1}) & \text{if Strategy 2 is used and } \omega > \theta \text{ hold,} \\ \mathcal{O}((k+1)\theta^{k+1}) & \text{if Strategy 2 is used and } \omega = \theta \text{ hold.} \end{cases} \end{aligned}$$

Thus, there exists a constant  $k_2$  satisfying (74) that makes the right-hand side of the above inequality less than  $\Theta$ . The fact that  $\|x_{k+1} - x^*\| < \Theta$  for  $k \geq k_2$  implies  $\|[x_{k+1}]_{g_i} - [x^*]_{g_i}\| < \Theta$  for all  $i \in \mathcal{S}(x^*)$  and  $k \geq k_2$ . This further suggests  $[x_{k+1}]_{g_i} \neq 0$  for all  $i \in \mathcal{S}(x^*)$  and  $k \geq k_2$ .

Parts (i) and (ii) follow since  $\mathcal{S}(x_{k+1}) = \mathcal{S}(x^*)$  for all  $k \geq \max\{k_1, k_2\}$ .  $\square$

We remark that if we do not make any assumption on the rate of convergence of  $\{\epsilon_k\}$  to zero (e.g., do not assume that a strategy such as Strategy 1 or Strategy 2 is used), then a small modification of the above analysis establishes that support identification still occurs. However, we are no longer able to give a bound on the maximum number of iterations before support identification will occur.

## 5 Numerical Results

In this section, we discuss our numerical tests. In Section 5.1 we discuss the formulation of the test problem, in Section 5.2 we discuss the details of our implementations, and in Section 5.3 we discuss the tests performed and the results of those tests.

### 5.1 Problem Formulation

We focus on the learning task of binary classification. To this end, we consider problem (1) where the function  $f$  is the binary logistic loss defined as

$$f(x) = \frac{1}{N} \sum_{i=1}^N \log \left( 1 + e^{-y_i x^T d_i} \right),$$

where  $d_i \in \mathbb{R}^n$  is the  $i$ th data point,  $N$  is the number of data points in the data set, and  $y_i \in \{-1, 1\}$  is the class label for the  $i$ th data point. Such an approach is commonly used in machine learning applications.

Data sets for the logistic regression problem were obtained from the LIBSVM repository.<sup>2</sup> We excluded all multi-class (greater than two) classification instances, instances with less than 100 features, and all data sets that were too large ( $\geq 8$ GB) to be loaded into memory. For the adult data (a1a–a9a) and webpage data (w1a–w8a), we used only the largest instances, namely a9a and w8a. This left us with the final set of 11 data sets found in Table 1. If the source of the data indicated that a data set was scaled, we used

<sup>2</sup><https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets>

it without modification. When the website did not indicate that scaling for a data set was used, we scaled each column of the feature data (i.e., feature-wise scaling) into the range  $[-1, 1]$  by dividing its entries by the largest entry in absolute value.

We consider the group regularizer  $r$  defined in (30). As for defining the groups, we use a similar strategy as that used in [29] to define different overlapping groups, where the overlapping of groups is controlled by parameters  $\mathbf{ratio} \in (0, 1)$  and  $\mathbf{grpsize} \in \mathbb{N}$ . The product of  $\mathbf{ratio}$  and  $\mathbf{grpsize}$  determines the number of elements in each group that overlap with its neighboring groups. For example, if  $n = 13$ ,  $\mathbf{ratio} = 0.2$ ,  $\mathbf{grpsize} = 5$ , then the groups are  $g_1 = \{1, 2, 3, 4, 5\}$ ,  $g_2 = \{5, 6, 7, 8, 9\}$ , and  $g_3 = \{9, 10, 11, 12, 13\}$ , which have overlap of size  $1 = \mathbf{ratio} \cdot \mathbf{grpsize}$ . In addition, we consider different solution sparsity levels, which are achieved by adjusting the parameters  $\{[\lambda]_i\}_{i=1}^{n_g}$ . Specifically, for each  $i \in [n_g]$ , we define  $[\lambda]_i = \Lambda \sqrt{|g_i|}$  for some  $\Lambda < \Lambda_{\min}$ , where  $\Lambda_{\min}$  is the minimum positive number such that the solution to the problem with  $[\lambda]_i = \Lambda_{\min} \sqrt{|g_i|}$  for all  $i \in [n_g]$  is  $x = 0$ . Since  $\Lambda_{\min}$  cannot be computed analytically, we test a range of values and choose the smallest one that gives the zero solution; as a starting guess for  $\Lambda_{\min}$ , we use the one derived from the non-overlapping group- $\ell_1$  case as discussed in [31, equation (23)]. For the tests described later in this section, we make specific choices for the parameters  $\mathbf{ratio}$ ,  $\mathbf{grpsize}$ , and  $\Lambda$ .

Table 1: The first column (data set) gives the name of the data set. The second column (N) and third column (n) indicate the number of data points and number of features (equivalently, the number of optimization variables), respectively. The fourth column (scale) provides the feature-wise scaling used: each feature is either scaled into the given interval or scaled to have mean zero ( $\mu = 0$ ) and variance one ( $\sigma^2 = 1$ ). The fifth column (who) indicates whether the data set came pre-scaled from the LIBSVM website (“website”), or whether it did not come pre-scaled and we scaled it (“us”) as described in Section 5.1.

data set	N	n	scale	who
a9a	32561	123	$[0, 1]$	website
colon-cancer	62	2000	$(\mu, \sigma^2) = (0, 1)$	website
duke breast-cancer	44	7192	$(\mu, \sigma^2) = (0, 1)$	website
gisette	6000	5000	$[-1, 1]$	website
leukemia	38	7129	$(\mu, \sigma^2) = (0, 1)$	website
madelon	2000	500	$[-1, 1]$	us
mushrooms	8124	112	$[0, 1]$	website
news20.binary	19996	1355191	$[0, 1]$	website
rcv1.binary	20242	47236	$[0, 1]$	website
real-sim	72309	20958	$[0, 1]$	website
w8a	49749	300	$[0, 1]$	website

## 5.2 Implementation details

In Algorithm 1, the values of the input parameters that we use are  $\xi = 0.5$ ,  $\eta = 10^{-3}$  and  $\zeta = 0.8$ , and the remaining parameters  $\gamma_1$  and  $\gamma_2$  will be described later since they are tuned. We set  $x_0 \leftarrow 0$  and  $\alpha_0 \leftarrow 1$ . For both  $\mathbf{option} = \mathbf{option\_1}$  and  $\mathbf{option} = \mathbf{option\_2}$ , Algorithm 1 decreases the value of the PG parameter for the next iteration using a multiplicative factor when  $\mathbf{flag}_k = \mathbf{decrease\_}\alpha$ . In practice, this strategy might be conservative as the PG parameter can become relatively small after encountering several consecutive iterations that trigger a decrease in the PG parameter. Instead, in our implementation, we update the PG parameter at the end of each iteration of Algorithm 1 following an idea from [5]. Specifically, for the  $k$ th iteration, if the inequality in Line 13 is not satisfied with  $j = 0$  (meaning that no backtracking is performed), we increase the PG parameter as  $\alpha_{k+1} \leftarrow 1.1\alpha_k$ ; otherwise, we decrease the PG parameter as  $\alpha_{k+1} \leftarrow \zeta\alpha_k$ . While this PG parameter update strategy works better than the basic strategy in Algorithm 1, it is not covered by our analysis in Section 3. However, a simple modification is to allow the PG parameter to increase a finite number of times, which would be covered by our analysis with a larger constant in the complexity result. We compare the two adaptive termination conditions  $\mathbf{option\_1}$  and  $\mathbf{option\_2}$  with the absolute termination condition from [25], i.e., setting  $\epsilon_k = \frac{\mathbf{const}}{k^3}$  for some positive number  $\mathbf{const}$  (its value will be given later since it is tuned), which we will refer to as  $\mathbf{option\_3}$ . For  $\mathbf{option\_3}$ , since there is no guarantee

that  $\hat{x}_{k+1} - x_k$  is a descent direction for  $f + r$ , we use a strategy considered in [25] where the PG parameter is initially set to  $\alpha_0$  and every time the inequality  $f(\hat{x}_{k+1}) \leq f(x_k) + \nabla f(x_k)^T(\hat{x}_{k+1} - x_k) + \frac{1}{\alpha_k} \|\hat{x}_{k+1} - x_k\|^2$  is violated by the approximate solution  $\hat{x}_{k+1}$ , the PG parameter is decreased by setting  $\alpha_{k+1} \leftarrow \zeta \alpha_k$  and we set  $x_{k+1} \leftarrow x_k$ .

Let us discuss the four termination conditions that we use in Algorithm 1.

1. **Maximum time.** Algorithm 1 is terminated if 12 hours is reached.
2. **Maximum iterations.** Algorithm 1 is terminated if  $10^6$  iterations is reached.
3. **Approximate solution found.** Algorithm 1 is terminated if

$$\frac{\|\hat{x}_{k+1} - x_k\| + \sqrt{2\alpha_k(\phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k))}}{\min(1, \alpha_k)} \leq \epsilon_{\text{tol}} := 10^{-5}, \quad (75)$$

which we proceed to justify. Note that strong duality implies that the pair  $(\hat{x}_{k+1}, \hat{y}_{k+1})$  returned by Algorithm 2 satisfies  $\phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi(T(x_k, \alpha_k); x_k, \alpha_k) \leq \phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k)$  so that  $\hat{x}_{k+1}$  is by definition (see (5)) an  $\bar{\epsilon}$ -PG update with  $\bar{\epsilon} = \phi(\hat{x}_{k+1}; x_k, \alpha_k) - \phi_d(\hat{y}_{k+1}; x_k, \alpha_k)$ . It follows from this fact, the reverse triangular inequality, and Lemma 3.2 that

$$\begin{aligned} \|T(x_k, \alpha_k) - x_k\| - \|\hat{x}_{k+1} - x_k\| &\leq \|T(x_k, \alpha_k) - x_k + x_k - \hat{x}_{k+1}\| \\ &= \|T(x_k, \alpha_k) - \hat{x}_{k+1}\| \leq \sqrt{2\alpha_k \bar{\epsilon}}. \end{aligned}$$

Combining this with the definitions of  $\chi_k$  and  $\bar{\epsilon}$ , and (75), it follows that

$$\chi_k = \frac{\|T(x_k, \alpha_k) - x_k\|}{\alpha_k} \leq \frac{\|\hat{x}_{k+1} - x_k\|}{\alpha_k} + \frac{\sqrt{2\alpha_k \bar{\epsilon}}}{\alpha_k} \leq \frac{\|\hat{x}_{k+1} - x_k\| + \sqrt{2\alpha_k \bar{\epsilon}}}{\min(1, \alpha_k)} \leq \epsilon_{\text{tol}},$$

i.e.,  $\hat{x}_k$  is an approximate solution since  $\chi_k \leq \epsilon_{\text{tol}}$ . This justifies (75).

4. **Numerical difficulties.** Algorithm 1 is terminated if numerical difficulties are encountered when the subproblem solver Algorithm 2 is called. In particular, while running our initial tests, we occasionally observed  $\epsilon_k$  to be on the order of  $10^{-12}$  when the sequence  $\{x_k\}$  approached a stationary point. In this case, Algorithm 2 was sometimes unable to return an approximate primal-dual pair  $(\hat{x}_{k+1}, \hat{y}_{k+1})$  that satisfied its approximate optimality conditions before reaching its iteration limit of 5000 iterations; the resulting  $\hat{x}_{k+1}$  in this scenario was often dense. To address this numerical challenge, when Algorithm 2 is terminated due to reaching its iteration limit, a reference point is used to (potentially) set additional groups of  $\hat{x}_{k+1}$  to zero. Concretely, we first define

$$\text{ref}(k) := \max\{\bar{k} : \bar{k} \leq k \text{ and Algorithm 2 terminated successfully in iteration } \bar{k}\},$$

which allows us to define, for each  $i \in [n_g]$ , the ‘‘corrected’’ iterate

$$[\hat{x}_{k+1}^{\text{corrected}}]_{g_i} = \begin{cases} [\hat{x}_{k+1}]_{g_i} & \text{if } [x_{\text{ref}(k)}]_{g_i} \neq 0, \\ 0 & \text{if } [x_{\text{ref}(k)}]_{g_i} = 0. \end{cases}$$

We use the corrected iterate instead of  $\hat{x}_{k+1}$  if it has a better objective value:

$$\hat{x}_{k+1} \leftarrow \begin{cases} \hat{x}_{k+1}^{\text{corrected}} & \text{if } \phi(\hat{x}_{k+1}^{\text{corrected}}; x_k, \alpha_k) \leq \phi(\hat{x}_{k+1}; x_k, \alpha_k), \\ \hat{x}_{k+1} & \text{otherwise.} \end{cases}$$

If two correction steps are needed, we terminate for numerical difficulties.

We now discuss Algorithm 2. The values for the input parameters are  $\xi_2 = 0.5$  and  $\eta_2 = 10^{-3}$ . Algorithm 2 is terminated when the inequality in Line 9 holds, the inequality in Line 13 holds, or 5000 iterations are performed. In the last case, a “correction” step is attempted as described above. For  $k = 0$ , we set  $\sigma \leftarrow 1$  and  $\hat{y}_{0,0} \leftarrow 0$ . For all future iterations  $k \geq 1$ , we set  $\sigma \leftarrow \sigma_{k-1,t}$  and  $\hat{y}_{k,0} \leftarrow \hat{y}_{k-1,t}$ , where  $t$  is the final iteration of Algorithm 2 when called during iteration  $k - 1$ . Finally, when `option_3` is considered (see the discussion above in this section), Algorithm 2 is terminated when the inequality in Line 9 or Line 13 (with the right-hand sides replaced by `const/k3`) holds since this implies that  $\hat{x}_{k+1} \in \mathcal{T}_{\epsilon_k}(x_k, \alpha_k)$  with  $\epsilon_k = \text{const}/k^3$ , as needed.

### 5.3 Tests and results

We have a Python implementation that is available upon request. All experiments were conducted on the Computational Optimization Research Laboratory (COR@L) cluster at Lehigh University with an AMD Opteron Processor 6128 2.0 GHz CPU.

#### 5.3.1 Comparison of the different termination conditions

We compare the performance of the three algorithm variants `option_1`, `option_2`, and `option_3`. For this test, we generate overlapping groups as described in Section 5.1 using values `ratio`  $\in \{0.1, 0.2, 0.3\}$ , `grpsize`  $\in \{10, 100\}$ , and  $\Lambda \in \{0.1\Lambda_{\min}, 0.01\Lambda_{\min}\}$ , and the 11 data sets given in Table 1. In total, this gives 132 test instances comprised of 12 different group constructions for each of the 11 data sets.

We tune values  $\gamma_1$  for `option_1`,  $\gamma_2$  for `option_2`, and `const` for `option_3` on the 72 test instances that correspond to the smaller data sets a9a, colon-cancer, duke breast-cancer, leukemia, mushrooms, and w8a. We searched for best values for  $\gamma_1$  and  $\gamma_2$  from the set  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ , and a best value for `const` from the set  $\{10^i\}_{i=0}^4$  (these sets were chosen based on preliminary testing). When deciding the best values for  $\gamma_1$ ,  $\gamma_2$  and `const`, we considered an algorithms performance on the 72 problem instances in terms of CPU time (averaged over three runs to account for randomness in the timings). This procedure resulted in the tuned values  $\gamma_1 = 0.2$ ,  $\gamma_2 = 0.5$ , and `const` = 1000. Empirically, the performances of `option_1` and `option_2`, which use relative criteria each iteration, were less sensitive to parameter tuning when compared to `option_3`, which uses an absolute criterion each iteration.

Using the above tuned parameters, we tested `option_1`, `option_2`, and `option_3` across the entire 132 problem instances. A summary of the termination statuses (see Section 5.2) returned by the three algorithms is found in Table 2. These results show that all three variants successful find an approximate solution to the majority of problem instances. We note that all three variants terminated due to the maximum iteration limit on all 12 instances of the data set madelon, and that all instances for which an algorithm reached the maximum time limit was for the data set news20.binary.

Table 2: Termination status summary for the algorithm variants `option_1`, `option_2`, and `option_3` on the 132 test instances described in Section 5.3.1. See Section 5.2 for the precise meaning of each termination condition.

	approximate solution found	maximum iteration limit	maximum time limit	numerical difficulties
<code>option_1</code>	108	16	7	1
<code>option_2</code>	107	15	8	2
<code>option_3</code>	107	16	9	0

Since Table 2 verifies that all three algorithm variants are relatively robust, we now consider various performance metrics in detail. First we compare the computing times.

**CPU time comparison.** Figure 1 contains performance profiles based on [17] for comparing the computing times of two algorithms on a collection of problem instances. For a plot that compares algorithms `option_i`

and `option_j` for  $\{i, j\} \subset \{1, 2, 3\}$  with  $i \neq j$ , each bar corresponds to a problem instance, with the height of the bar given by

$$-\log_2 \left( \frac{\text{time required by option}_i}{\text{time required by option}_j} \right). \quad (76)$$

Thus, an upward pointing (blue) bar indicates that `option_i` took less time than `option_j` to solve that test problem. In contrast, a downward pointing (red) bar means that `option_i` took more time than `option_j` to solve that test problem. In either case, the size of the bar indicates the magnitude of the outperformance. If both algorithms fail to solve a problem instance, that instance is not included in the plot. For problem instances that are successfully solved by only one of the two algorithms, the height of the bar is set according to the following procedure. For each  $p \in \{1, 2, \dots, 132\}$  and  $i \in \{1, 2, 3\}$ , let  $t_{p,i}$  denote the time consumed by `option_i` on the  $p$ th problem and  $\mathcal{S}_{i,j}$  denote the set of all problem instances that are successfully solved by both `option_i` and `option_j`. Then, if exactly one of `option_i` and `option_j` fails on problem instance  $p$ , the height of the associated bar is set as  $1.5 \max_{j \in \mathcal{S}_{i,j}} \{|\log_2(t_{p,i}/t_{p,j})|\}$  pointing in the direction of the algorithm that successfully solved the problem. For each given plot, a single performance metric can be computed for each of the two competing algorithms by computing the areas of their respective bars (the width of each bar is one so the the area of each bar is equal to its height); these areas are indicated in the legend of each plot. As shown in Fig 1, the adaptive criteria used by `option_1` and `option_2` outperforms the absolute criterion used in `option_3`.

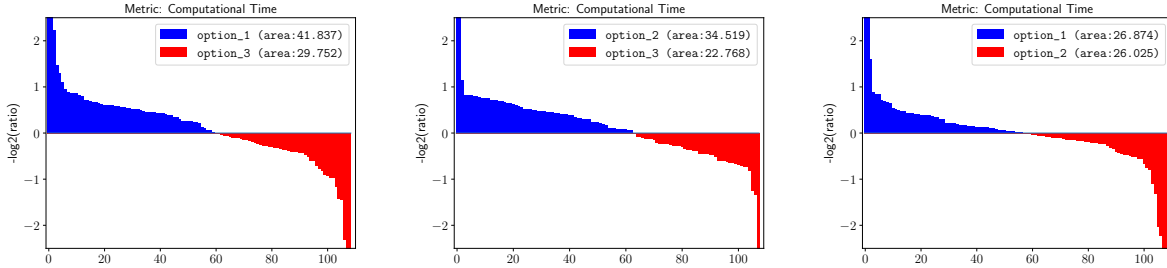


Figure 1: A performance profile for CPU time (seconds). In each plot, we exclude problem instances for which both algorithms fail.

**Final objective value and sparsity.** For problem instance  $p \in \{1, 2, \dots, 132\}$  and algorithm `option_i` for  $i \in \{1, 2, 3\}$ , we let  $F_{p,i}$  denote the final objective value returned by `option_i` when solving problem  $p$ . If  $F_i - F_j < -10^{-6}$ , then we say `option_i` obtained a “better” objective value compared to `option_j`; if  $F_i - F_j > 10^{-6}$ , then we say `option_i` obtained a “worse” objective value compared to `option_j`; and if  $|F_i - F_j| \leq 10^{-6}$ , then we consider them to have performed the “same”.

One must also consider the sparsity of the returned solutions. If the solution returned by `option_i` is sparser than the solution returned by `option_j`, we say that `option_i` performed “better” than `option_j`; if the solution returned by `option_i` is denser than the solution returned by `option_j`, we say that `option_i` performed “worse” than `option_j`; if the solutions returned by both algorithms have the same level of sparsity, we say that the two algorithms performed the “same”.

The results from our tests can be found in Table 3. From these results, we may first conclude that the adaptive termination criteria (`option_1` and `option_2`) outperform the absolute termination criterion (`option_3`) in achieving a lower objective value. Second, the algorithms based on an adaptive termination criteria (`option_1` and `option_2`) slightly outperform the algorithm using an absolute termination criterion (`option_3`) in terms of obtaining sparser solutions. Third, `option_1` slightly outperforms `option_2` in terms of achieving a better final objective value but at the expense of being slightly outperformed by `option_2` in terms of final solution sparsity.

Table 3: A comparison of solution sparsity and final objective value for `option_1`, `option_2`, and `option_3` over the entire 132 problem instances. For example, the 6 in the first row (“`option_1` versus `option_3`”) indicates that `option_1` returned a sparser solution (i.e., a “better” solution) than `option_3` on 6 problem instances. Similarly, the 24 in that same row means that `option_1` returned a lower objective function value (i.e., a “better” final objective value) than `option_3` on 24 problem instances.

	sparsity			objective value		
	better	same	worse	better	same	worse
<code>option_1</code> versus <code>option_3</code>	6	121	5	24	103	5
<code>option_2</code> versus <code>option_3</code>	8	118	6	23	103	6
<code>option_1</code> versus <code>option_2</code>	4	118	10	10	115	7

### 5.3.2 Subproblem solver comparison: Algorithm 2 versus projected gradient ascent

We now explore the support identification property of our subproblem solver Algorithm 2. It is clear from the previous section that Algorithm 1 with any `option`  $\in$   $\{\text{option}_1, \text{option}_2, \text{option}_3\}$  returns sparse solutions when Algorithm 2 is used as the subproblem solver. Here, we compare the sparsity of the solutions produced by Algorithm 1 when our proposed subproblem solver Algorithm 2 is replaced by a projected gradient ascent (PGA) method. The results of our tests are in Table 4; we only compare PGA to `option_1` since `option_1` and `option_2` perform similarly.

For each data set in Table 1 we created a single test instance by first setting `ratio`  $\leftarrow$  0.1 and `grpsize`  $\leftarrow$  10 (see Section 5.1). To ensure that a sparse solution existed for each problem instance, we selected  $\Lambda$  (see Section 5.1) differently for each data set as indicated in the second column of Table 4. Note that the three largest data sets from Table 1 are excluded from Table 4 because the PGA subproblem solver was not able to achieve any reasonable solution within the time limit (see Section 5.2).

Table 4 indicates that when Algorithm 1 uses either Algorithm 2 or the PGA method as the subproblem solver the same objective value (up to 6 digits of accuracy) is achieved. However, while Algorithm 2 consistently finds sparse solutions, the subproblem solver PGA consistently finds dense solutions. Although it is possible to enhance the PGA method through a post-processing step that forces blocks of variables with small magnitude to zero, this would require the careful tuning of a threshold parameter. Our method does not require any such post-processing to achieve sparse solutions. Although not the focus of this section, we observed that Algorithm 1 converged in fewer iterations when Algorithm 2 is used as the subproblem solver compared to when PGA is used. It is clear that using Algorithm 2 is crucial to obtaining sparse solutions.

Table 4: The test results for Algorithm 1 when using Algorithm 2 or the PGA algorithm as its subproblem solvers. Columns “#z”, “#nz”, and “ $F$ ” give the number of zero groups, the number of non-zero groups, and the final objective value, respectively.

data set	$\Lambda$	Algorithm 2 ( <code>option_1</code> )			PGA		
		#z	#nz	$F$	#z	#nz	$F$
a9a	0.013458	12	2	0.508337	0	14	0.508337
colon-cancer	0.017751	213	10	0.336270	1	222	0.336270
duke breast-cancer	0.016198	779	13	0.246910	2	790	0.246910
gisette	0.012003	536	20	0.402671	2	554	0.402671
leukemia	0.020514	781	11	0.258627	0	792	0.258627
madelon	0.000402	19	37	0.666079	0	56	0.666112
mushrooms	0.009528	10	3	0.316138	0	13	0.316138
w8a	0.006687	24	10	0.429029	0	34	0.429029



## 6 Conclusion

We proposed a PG framework to solve optimization problems that use a regularizer whose proximal operator does not have a closed-form solution. To address this challenge, we proposed two adaptive criteria for approximately solving the PG subproblem, and then proved a worst-case complexity result for the maximum number of iterations before an approximate first-order solution is computed. We also designed an enhanced solver for the PG subproblem when the group- $\ell_1$  regularizer is used. When this subproblem solver is used within our overall PG method, we provide an upper bound on the number of iterations before optimal support identification occurs. Numerical experiments on regularized logistic regression problems illustrate the effectiveness of our approach for efficiently finding structured sparse solutions.

## References

- [1] Kristin L. Ayers and Heather J. Cordell. SNP selection in genome-wide and candidate gene studies via penalized logistic regression. *Genetic Epidemiology*, 34(8):879–891, 2010.
- [2] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.
- [3] Amir Beck. *First-order methods in optimization*, volume 25. SIAM, Philadelphia, PA, 2017.
- [4] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [5] Stephen R Becker, Emmanuel J Candès, and Michael C Grant. Templates for convex cone problems with applications to sparse signal recovery. *Math. Prog. Comp.*, 3(3):165, 2011.
- [6] Dimitri Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, Massachusetts, 2 edition, 1999.
- [7] D.P. Bertsekas, A. Nedić, and A.E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific optimization and computation series. Athena Scientific, Belmont, Mass, 2003.
- [8] Gavin C. Cawley and Nicola L. C. Talbot. Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *Bioinformatics*, 22(19):2348–2355, 2006.
- [9] Frank E. Curtis, Yutong Dai, and Daniel P. Robinson. A subspace acceleration method for minimization involving a group sparsity-inducing regularizer. *SIAM Journal on Optimization*, 32(2):545–572, 2022.
- [10] Marc Fuentes, Jérôme Malick, and Claude Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Comp. Opt. and Applications*, 53(3):755–769, December 2012.
- [11] Alfredo N Iusem. On the convergence properties of the projected gradient method for convex optimization. *Computational & Applied Mathematics*, 22:37–52, 2003.
- [12] Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.
- [13] Ching-pei Lee. Accelerating inexact successive quadratic approximation for regularized optimization through manifold identification. *arXiv preprint arXiv:2012.02522*, 2020.
- [14] Ching-pei Lee and Stephen J Wright. Inexact successive quadratic approximation for regularized optimization. *Computational Optimization and Applications*, 72(3):641–674, 2019.
- [15] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research*, 18(1):7854–7907, 2018.

- [16] Zhi-Quan Luo and Jong-Shi Pang. Error bounds for analytic systems and their applications. *Mathematical Programming*, 67(1):1–28, 1994.
- [17] José Luis Morales. A numerical study of limited memory BFGS methods. *Applied Mathematics Letters*, 15(4):481–487, 2002.
- [18] Boris S Mordukhovich and Nguyen Mau Nam. *An easy path to convex analysis and applications*, volume 6. Morgan & Claypool Publishers, San Rafael, CA., 2013.
- [19] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*. Springer Science & Business Media, ., 2004.
- [20] Julie Nutini, Mark Schmidt, and Warren Hare. “active-set complexity” of proximal gradient: How long does it take to find the sparsity pattern? *Opt. Letters*, 13(4):645–655, 2019.
- [21] Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*, 2011.
- [22] Julian Rasch and Antonin Chambolle. Inexact first-order primal–dual algorithms. *Computational Optimization and Applications*, 76(2):381–430, 2020.
- [23] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.
- [24] Saverio Salzo and Silvia Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex analysis*, 19(4):1167–1192, 2012.
- [25] Mark Schmidt, Nicolas Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *Advances in Neural Information Processing Systems*, 24:1458–1466, 2011.
- [26] Ting Shen, Jiehui Jiang, Wei Lin, Jingjie Ge, Ping Wu, Yongjin Zhou, Chuantao Zuo, Jian Wang, Zhuangzhi Yan, and Kuangyu Shi. Use of overlapping group lasso sparse deep belief network to discriminate Parkinson’s disease and normal control. *Frontiers in Neuroscience*, 13:396, 2019.
- [27] Yifan Sun, Halyun Jeong, Julie Nutini, and Mark Schmidt. Are we there yet? manifold identification of gradient-related proximal methods. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1110–1119, Naha, Okinawa, Japa, 2019. PMLR.
- [28] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [29] Silvia Villa, Lorenzo Rosasco, Sofia Mosci, and Alessandro Verri. Proximal methods for the latent group lasso penalty. *Computational Optimization and Applications*, 58(2):381–407, 2014.
- [30] Silvia Villa, Saverio Salzo, Luca Baldassarre, and Alessandro Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.
- [31] Yi Yang and Hui Zou. A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, 25(6):1129–1141, Nov 2015.
- [32] Lei Yuan, Jun Liu, and Jieping Ye. Efficient methods for overlapping group lasso. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2104–2116, 2013.
- [33] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.
- [34] Yaohui Zeng and Patrick Breheny. Overlapping group logistic regression with applications to genetic pathway selection. *Cancer Informatics*, 15:CIN–S40043, 2016.

- [35] Ji Zhu and Trevor Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004.
- [36] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.