Efficient Methods for Overlapping Group Lasso

Lei Yuan, Jun Liu, and Jieping Ye, *Senior Member*, *IEEE*

Abstract—The group Lasso is an extension of the Lasso for feature selection on (predefined) nonoverlapping groups of features. The nonoverlapping group structure limits its applicability in practice. There have been several recent attempts to study a more general formulation where groups of features are given, potentially with overlaps between the groups. The resulting optimization is, however, much more challenging to solve due to the group overlaps. In this paper, we consider the efficient optimization of the overlapping group Lasso penalized problem. We reveal several key properties of the proximal operator associated with the overlapping group Lasso, and compute the proximal operator by solving the smooth and convex dual problem, which allows the use of the gradient descent type of algorithms for the optimization. Our methods and theoretical results are then generalized to tackle the general overlapping group Lasso formulation based on the ℓ_q norm. We further extend our algorithm to solve a nonconvex overlapping group Lasso formulation based on the capped norm regularization, which reduces the estimation bias introduced by the convex penalty. We have performed empirical evaluations using both a synthetic and the breast cancer gene expression dataset, which consists of 8,141 genes organized into (overlapping) gene sets. Experimental results show that the proposed algorithm is more efficient than existing state-of-the-art algorithms. Results also demonstrate the effectiveness of the nonconvex formulation for overlapping group Lasso.

Index Terms—Sparse learning, overlapping group Lasso, proximal operator, difference of convex programming

1 Introduction

Problems with high dimensionality have become common over recent years. High dimensionality poses significant challenges in building interpretable models with high-prediction accuracy. Regularization has been commonly employed to obtain more stable and interpretable models. A well-known example is the penalization of the ℓ_1 norm of the estimator, known as the Lasso [1]. The ℓ_1 norm regularization has achieved great success in many applications. However, in some applications [2], we are interested in finding important explanatory factors in predicting the response variable, where each explanatory factor is represented by a group of input features. In such cases, the selection of important features corresponds to the selection of groups of features. As an extension of the Lasso, the group Lasso [2] based on the combination of the ℓ_1 norm and the ℓ_2 norm has been proposed for group feature selection, and many algorithms [3], [4], [5] have been proposed for efficient optimization. However, the nonoverlapping group structure in group Lasso limits its applicability in practice. For example, in microarray gene expression data analysis, genes may form overlapping groups as each gene may participate in multiple pathways [6].

Several recent works [6], [7], [8], [9], [10], [11], [12] have studied the overlapping group Lasso, where groups of features are given, potentially with overlaps between the groups. The resulting optimization is, however, much more challenging to solve due to the group overlaps. When solving the overlapping group Lasso problem, one can reformulate it as a second order cone program and solve it by a generic toolbox, which, however, does not scale well. Jenatton et al. [13] proposed an alternating algorithm called SLasso for solving the equivalent reformulation. However, SLasso involves an expensive matrix inversion at each alternating iteration, and there is no known global convergence rate for such an alternating procedure. A reformulation [14] was also proposed such that the original problem can be solved by the Alternating Direction Method of Multipliers (ADMM), which involves solving a linear system at each iteration and may not scale well for high-dimensional problems. Argyriou et al. [15] adopted the proximal gradient method for solving the overlapping group Lasso, and a fixed point method was developed to compute the proximal operator. Chen et al. [16] employed a smoothing technique to solve the overlapping group Lasso problem. Mairal et al. [9] proposed to solve the proximal operator associated with the overlapping group Lasso defined as the sum of the ℓ_∞ norms, which, however, is not applicable to the formulation considered in this paper.

In this paper, we develop an efficient algorithm for the overlapping group Lasso penalized problem via the accelerated gradient descent (AGD) method. The AGD method has recently received increasing attention in machine learning due to the fast convergence rate even for nonsmooth convex problems. One of the key operations is the computation of the proximal operator associated with the penalty. We reveal several key properties of the proximal operator associated with the overlapping group Lasso penalty and propose several possible reformulations that can be solved efficiently. The main contributions of this paper include: 1) We develop a low-cost prepossessing procedure to identify (and then remove) zero groups in the proximal operator, which dramatically reduces the size of

- *L. Yuan and J. Ye are with the Department of Computer Science and Engineering and the Center for Evolutionary Medicine and Informatics of the Biodesign Institute, Arizona State University, Tempe, AZ 85287. E-mail: {lei.yuan, jieping.ye}@asu.edu.*
- *J. Liu is with Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540.*

Manuscript received 15 Oct. 2011; revised 22 July 2012; accepted 17 Dec. 2012; published online 9 Jan. 2013.
Recommended for acceptance by E.P. Xing.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-10-0743.
Digital Object Identifier no. 10.1109/TPAMI.2013.17.

0162-8828/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

the problem to be solved; 2) we propose one dual formulation and two proximal splitting formulations for the proximal operator; and 3) for the dual formulation, we further derive the duality gap, which can be used to check the quality of the solution and determine the convergence of the algorithm.

In addition, we propose two extensions to the proposed algorithm. First, we generalize our method and theoretical results to solve the general overlapping group Lasso with the ℓ_q norm with $q > 1$ (when $q = 1$, there is no grouping effect). We then tackle a nonconvex overlapping group Lasso formulation based on the capped norm regularization. We propose to decompose the nonconvex capped norm penalty to the difference of two convex functions and solve the equivalent problem using DC programming. The subproblem of each DC step is equivalent to the original overlapping group Lasso problem.

We have performed empirical evaluations using both synthetic data and the breast cancer gene expression dataset, which consists of 8,141 genes organized into (overlapping) gene sets. Experimental results demonstrate the efficiency of the proposed algorithm in comparison with existing state-of-the-art algorithms. Results also demonstrate the effectiveness of the nonconvex overlapping group Lasso formulation.

Notations: $\|\cdot\|$ denotes the euclidean norm, and $\mathbf{0}$ denotes a vector of zeros. $\mathrm{SGN}(\cdot)$ and $\mathrm{sgn}(\cdot)$ are defined in a component wise fashion as: 1) If $t = 0$, then $\mathrm{SGN}(t) = [-1, 1]$ and $\mathrm{sgn}(t) = 0$; 2) if $t > 0$, then $\mathrm{SGN}(t) = \{1\}$ and $\mathrm{sgn}(t) = 1$; and 3) if $t < 0$, $\mathrm{SGN}(t) = \{-1\}$ and $\mathrm{sgn}(t) = -1$. $G_i \subseteq \{1, 2, \ldots, p\}$ denotes an index set, and $\mathbf{x}_{G_i}$ denote a subvector of $\mathbf{x}$ restricted to G_i.

2 The Overlapping Group Lasso

We consider the following overlapping group Lasso penalized problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) = l(\mathbf{x}) + \phi_{\lambda_2}^{\lambda_1}(\mathbf{x}), \tag{1}$$

where $l(\cdot)$ is a smooth convex loss function, e.g., the least squares loss,

$$\phi_{\lambda_2}^{\lambda_1}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| \tag{2}$$

is the overlapping group Lasso penalty, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are regularization parameters, $w_i > 0$, $i = 1, 2, \ldots, g$, $G_i \subseteq \{1, 2, \ldots, p\}$ contains the indices corresponding to the ith group of features, and $\|\cdot\|$ denotes the euclidean norm. We consider the general ℓ_q norm with $q > 1$ in Section 4. Note that the first term in (2) can be absorbed into the second term, which, however, will introduce p additional groups. The g groups of features are prespecified, and they may overlap. The penalty in (2) is a special case of the more general Composite Absolute Penalty (CAP) family [10]. When the groups are disjoint with $\lambda_1 = 0$ and $\lambda_2 > 0$, the model in (1) reduces to the group Lasso [2]. If $\lambda_1 > 0$ and $\lambda_2 = 0$, then the model in (1) reduces to the standard Lasso [1].

In this paper, we propose to make use of the AGD [17], [18], [19] for solving (1), due to its fast convergence rate. The algorithm is called "FoGLasso," which stands for *F*ast *o*verlapping *G*roup *Lasso*. One of the key steps in the proposed FoGLasso algorithm is the computation of the proximal operator associated with the penalty in (2), and we present an efficient algorithm for the computation in the next section.

In FoGLasso, we first construct a model for approximating $f(\cdot)$ at the point $\mathbf{x}$ as

$$f_{L,\mathbf{x}}(\mathbf{y}) = [l(\mathbf{x}) + \langle l'(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle] + \phi_{\lambda_1}^{\lambda_2}(\mathbf{y}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2, \tag{3}$$

where $L > 0$. The model $f_{L,\mathbf{x}}(\mathbf{y})$ consists of the first-order Taylor expansion of the smooth function $l(\cdot)$ at the point $\mathbf{x}$, the nonsmooth penalty $\phi_{\lambda_1}^{\lambda_2}(\mathbf{x})$, and a regularization term $\frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2$. Next, a sequence of approximate solutions $\{\mathbf{x}_i\}$ is computed as follows: $\mathbf{x}_{i+1} = \arg\min_{\mathbf{y}} f_{L_i,\mathbf{s}_i}(\mathbf{y})$, where the search point $\mathbf{s}_i$ is an affine combination of $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ as $\mathbf{s}_i = \mathbf{x}_i + \beta_i(\mathbf{x}_i - \mathbf{x}_{i-1})$, for a properly chosen coefficient β_i, L_i is determined by the line search according to the Armijo-Goldstein rule so that L_i should be appropriate for $\mathbf{s}_i$, i.e., $f(\mathbf{x}_{i+1}) \leq f_{L_i,\mathbf{s}_i}(\mathbf{x}_{i+1})$. A key building block in FoGLasso is the minimization of (3), whose solution is known as the proximal operator [20]. The computation of the proximal operator is the main technical contribution of this paper. The pseudocode of FoGLasso is summarized in Algorithm 1, where the proximal operator $\pi(\cdot)$ is defined in (4). In practice, we can terminate Algorithm 1 if the change of the function values corresponding to adjacent iterations is within a small value, say 10^{-5}.

Algorithm 1. The FoGLasso Algorithm

```
Input: $L_0 > 0, \mathbf{x}_0, k$
Output: $\mathbf{x}_{k+1}$
Initialize $\mathbf{x}_1 = \mathbf{x}_0$, $\alpha_{-1} = 0$, $\alpha_0 = 1$, and $L = L_0$.
for $i = 1$ to $k$ do
    Set $\beta_i = \frac{\alpha_{i-2} - 1}{\alpha_{i-1}}$, $\mathbf{s}_i = \mathbf{x}_i + \beta_i(\mathbf{x}_i - \mathbf{x}_{i-1})$
    Find the smallest $L = 2^j L_{i-1}, j = 0, 1, \ldots$ such that
    $f(\mathbf{x}_{i+1}) \leq f_{L,\mathbf{s}_i}(\mathbf{x}_{i+1})$ holds, where
        $\mathbf{x}_{i+1} = \pi_{\lambda_2/L}^{\lambda_1/L}(\mathbf{s}_i - \frac{1}{L} l'(\mathbf{s}_i))$
    Set $L_i = L$ and $\alpha_{i+1} = \frac{1 + \sqrt{1 + 4\alpha_i^2}}{2}$
end for
```

3 The Associated Proximal Operator and Its Efficient Computation

The proximal operator associated with the overlapping group Lasso penalty is defined as follows:

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\left\{ g_{\lambda_2}^{\lambda_1}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 + \phi_{\lambda_2}^{\lambda_1}(\mathbf{x})\right\}, \tag{4}$$

which is a special case of (1) by setting $l(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2$. It can be verified that the approximate solution $\mathbf{x}_{i+1} = \arg\min_{\mathbf{y}} f_{L_i,\mathbf{s}_i}(\mathbf{y})$ is given by $\mathbf{x}_{i+1} = \pi_{\lambda_2/L_i}^{\lambda_1/L_i}(\mathbf{s}_i - \frac{1}{L_i} l'(\mathbf{s}_i))$. The efficient computation of the proximal operator is key to many sparse learning algorithms [21], [22]. Next, we focus on the efficient computation of $\pi_{\lambda_2}^{\lambda_1}(\mathbf{v})$ in (4) for a given $\mathbf{v}$.

The rest of this section is organized as follows: In Section 3.1, we discuss some key properties of the proximal operator, based on which we propose a preprocessing technique that will significantly reduce the size of the problem. We then propose to solve it via the dual formulation in Section 3.2, and the duality gap is also derived. Several alternative methods for solving the proximal operator via proximal splitting methods are discussed in Section 3.3.

3.1 Key Properties of the Proximal Operator

Denote $\odot$ as the point-wise product. We first reveal several basic properties of the proximal operator $\pi_{\lambda_2}^{\lambda_1}(\mathbf{v})$.

Lemma 1. *Suppose that $\lambda_1, \lambda_2 \geq 0$, and $w_i > 0$, for $i = 1, 2, \ldots, g$. Let $\mathbf{x}^* = \pi_{\lambda_2}^{\lambda_1}(\mathbf{v})$. The following holds:*

1. *if $v_i > 0$, then $0 \leq x_i^* \leq v_i$;*
2. *if $v_i < 0$, then $v_i \leq x_i^* \leq 0$;*
3. *if $v_i = 0$, then $x_i^* = 0$;*
4. *$\mathrm{SGN}(\mathbf{v}) \subseteq \mathrm{SGN}(\mathbf{x}^*)$; and*
5. *$\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \mathrm{sgn}(\mathbf{v}) \odot \pi_{\lambda_2}^{\lambda_1}(|\mathbf{v}|)$.*

Proof. When $\lambda_1, \lambda_2 \geq 0$, and $w_i \geq 0$, for $i = 1, 2, \ldots, g$, the objective function $g_{\lambda_2}^{\lambda_1}(\cdot)$ is strictly convex; thus, $\mathbf{x}^*$ is the unique minimizer. We first show if $v_i > 0$, then $0 \leq x_i^* \leq v_i$. If $x_i^* > v_i$, then we can construct a $\hat{\mathbf{x}}$ as follows: $\hat{x}_j = x_j^*$, $j \neq i$, and $\hat{x}_i = v_i$. Similarly, if $x_i^* < 0$, then we can construct a $\hat{\mathbf{x}}$ as follows: $\hat{x}_j = x_j^*$, $j \neq i$, and $\hat{x}_i = 0$. It is easy to verify that $\hat{\mathbf{x}}$ achieves a lower objective function value than $\mathbf{x}^*$ in both cases. We can prove the second and the third properties using similar arguments. Finally, we can prove the fourth and the fifth properties using the definition of $\mathrm{SGN}(\cdot)$ and the first three properties. □

Next, we show that $\pi_{\lambda_2}^{\lambda_1}(\cdot)$ can be directly derived from $\pi_{\lambda_2}^{0}(\cdot)$ by soft-thresholding. Thus, we only need to focus on the case when $\lambda_1 = 0$. This simplifies the optimization in (4). It is an extension of the result for Fused Lasso by Friedman [23].

Theorem 1. *Let $\mathbf{u} = \mathrm{sgn}(\mathbf{v}) \odot \max(|\mathbf{v}| - \lambda_1, 0)$, and*

$$\pi_{\lambda_2}^{0}(\mathbf{u}) = \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ h_{\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| \right\}. \tag{5}$$

Then, the following holds: $\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \pi_{\lambda_2}^{0}(\mathbf{u})$.

Proof. Denote the unique minimizer of $h_{\lambda_2}(\cdot)$ as $\mathbf{x}^*$. The sufficient and necessary condition for the optimality of $\mathbf{x}^*$ is:

$$\mathbf{0} \in \partial h_{\lambda_2}(\mathbf{x}^*) = \mathbf{x}^* - \mathbf{u} + \partial \phi_{\lambda_2}^{0}(\mathbf{x}^*), \tag{6}$$

where $\partial h_{\lambda_2}(\mathbf{x})$ and $\partial \phi_{\lambda_2}^{0}(\mathbf{x})$ are the subdifferential sets of $h_{\lambda_2}(\cdot)$ and $\phi_{\lambda_2}^{0}(\cdot)$ at $\mathbf{x}$, respectively.

Next, we need to show $\mathbf{0} \in \partial g_{\lambda_2}^{\lambda_1}(\mathbf{x}^*)$. The subdifferential of $g_{\lambda_2}^{\lambda_1}(\cdot)$ at $\mathbf{x}^*$ is given by

$$\begin{aligned} \partial g_{\lambda_2}^{\lambda_1}(\mathbf{x}^*) &= \mathbf{x}^* - \mathbf{v} + \partial \phi_{\lambda_2}^{\lambda_1}(\mathbf{x}^*) \\ &= \mathbf{x}^* - \mathbf{v} + \lambda_1 \mathrm{SGN}(\mathbf{x}^*) + \partial \phi_{\lambda_2}^{0}(\mathbf{x}^*). \end{aligned} \tag{7}$$

It follows from the definition of $\mathbf{u}$ that $\mathbf{u} \in \mathbf{v} - \lambda_1 \mathrm{SGN}(\mathbf{u})$. Using the fourth property in Lemma 1, we have $\mathrm{SGN}(\mathbf{u}) \subseteq \mathrm{SGN}(\mathbf{x}^*)$. Thus,

$$\mathbf{u} \in \mathbf{v} - \lambda_1 \mathrm{SGN}(\mathbf{x}^*). \tag{8}$$

It follows from (6)-(8) that $\mathbf{0} \in \partial g_{\lambda_2}^{\lambda_1}(\mathbf{x}^*)$. □

It follows from Theorem 1 that we only need to focus on the optimization of (5) in the following discussion. The difficulty in the optimization of (5) lies in the large number of groups that may overlap. In practice, many groups will be zero, thus achieving a sparse solution (a sparse solution is desirable in many applications). However, the zero groups are not known in advance. The key question we aim to address is how we can identify as many zero groups as possible to reduce the complexity of the optimization. Next, we present a sufficient condition for a group to be zero.

Lemma 2. *Denote the minimizer of $h_{\lambda_2}(\cdot)$ in (5) by $\mathbf{x}^*$. If the ith group satisfies $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$, then $\mathbf{x}_{G_i}^* = \mathbf{0}$, i.e., the ith group is zero.*

Proof. We decompose $h_{\lambda_2}(\mathbf{x})$ into two parts as follows:

$$\begin{aligned} h_{\lambda_2}(\mathbf{x}) = &\left(\frac{1}{2}\|\mathbf{x}_{G_i} - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i \|\mathbf{x}_{G_i}\| \right) \\ &+ \left(\frac{1}{2}\|\mathbf{x}_{\overline{G}_i} - \mathbf{u}_{\overline{G}_i}\|^2 + \lambda_2 \sum_{j \neq i} w_j \|\mathbf{x}_{G_j}\| \right), \end{aligned} \tag{9}$$

where $\overline{G}_i = \{1, 2, \ldots, p\} - G_i$ is the complementary set of G_i. We consider the minimization of $h_{\lambda_2}(\mathbf{x})$ in terms of $\mathbf{x}_{G_i}$ when $\mathbf{x}_{\overline{G}_i} = \mathbf{x}_{\overline{G}_i}^*$ is fixed. It can be verified that if $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$, then $\mathbf{x}_{G_i}^* = \mathbf{0}$ minimizes both terms in (9) simultaneously. Thus, we have $\mathbf{x}_{G_i}^* = \mathbf{0}$. □

Lemma 2 may not identify many true zero groups due to the strong condition imposed. The lemma below weakens the condition in Lemma 2. Intuitively, for a group G_i, we first identify all existing zero groups that overlap with G_i, and then compute the overlapping index subset S_i of G_i as

$$S_i = \bigcup_{j \neq i, \mathbf{x}_{G_j}^* = \mathbf{0}} (G_j \cap G_i). \tag{10}$$

We can show that $\mathbf{x}_{G_i}^* = \mathbf{0}$ if $\|\mathbf{u}_{G_i - S_i}\| \leq \lambda_2 w_i$ is satisfied. Note that this condition is much weaker than the condition in Lemma 2, which requires that $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$.

Lemma 3. *Denote the minimizer of $h_{\lambda_2}(\cdot)$ by $\mathbf{x}^*$. Let S_i, a subset of G_i, be defined in (10). If $\|\mathbf{u}_{G_i - S_i}\| \leq \lambda_2 w_i$ holds, then $\mathbf{x}_{G_i}^* = \mathbf{0}$.*

Proof. Suppose that we have identified a collection of zero groups. By removing these groups, the original problem (5) can then be reduced to:

$$\min_{\mathbf{x}(I_1) \in \mathbb{R}^{|I_1|}} \frac{1}{2}\|\mathbf{x}(I_1) - \mathbf{u}(I_1)\|^2 + \lambda_2 \sum_{i \in \mathcal{G}_1} w_i \|\mathbf{x}_{G_i - S_i}\|,$$

where I_1 is the reduced index set, i.e., $I_1 = \{1, 2, \ldots, p\} - \bigcup_{i : \mathbf{x}_{G_i}^* = \mathbf{0}} G_i$, and $\mathcal{G}_1 = \{i : \mathbf{x}_{G_i}^* \neq \mathbf{0}\}$ is the index set of the remaining nonzero groups. Note that $\forall i \in \mathcal{G}_1$, $G_i - S_i \in I_1$. By applying Lemma 2 again, we show that if $\|\mathbf{u}_{G_i - S_i}\| \leq \lambda_2 w_i$ holds, then $\mathbf{x}_{G_i - S_i}^* = \mathbf{0}$. Thus, $\mathbf{x}_{G_i}^* = \mathbf{0}$. □

Lemma 3 naturally leads to an iterative procedure for identifying the zero groups: For each group G_i, if $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$, then we set $\mathbf{u}_{G_i} = \mathbf{0}$; we cycle through all groups repeatedly until $\mathbf{u}$ does not change. Let $p' = |\{u_i : u_i \neq 0\}|$ be the number of nonzero elements in $\mathbf{u}$, $g' = |\{\mathbf{u}_{G_i} : \mathbf{u}_{G_i} \neq \mathbf{0}\}|$ be the number of the nonzero groups, and $\mathbf{x}^*$ denote the minimizer of $h_{\lambda_2}(\cdot)$. It follows from Lemma 3 and Lemma 1 that if $u_i = 0$, then $x_i^* = 0$. Therefore, by applying the above iterative procedure, we can find the minimizer of (5) by solving a reduced problem that has $p' \leq p$ variables and $g' \leq g$ groups. With some abuse of notation, we still use (5) to denote the resulting reduced problem. In addition, from Lemma 1, we only focus on $\mathbf{u} > 0$ in the following discussion, and the analysis can be easily generalized to the general case.

3.2 Reformulation as an Equivalent Smooth Convex Optimization Problem

It follows from the first two properties of Lemma 1 that we can rewrite (5) as

$$\pi^0_{\lambda_2}(\mathbf{u}) = \arg \min_{\substack{\mathbf{x} \in \mathbb{R}^p \\ \mathbf{0} \preceq \mathbf{x} \preceq \mathbf{u}}} h_{\lambda_2}(\mathbf{x}), \tag{11}$$

where $\preceq$ denotes the element-wise inequality, and

$$h_{\lambda_2}(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\|,$$

and the minimizer of $h_{\lambda_2}(\cdot)$ is constrained to be nonnegative due to $\mathbf{u} > 0$ (refer to the discussion at the end of Section 3.1).

Making use of the dual norm of the euclidean norm $\|\cdot\|$, we can rewrite $h_{\lambda_2}(\mathbf{x})$ as:

$$h_{\lambda_2}(\mathbf{x}) = \max_{Y \in \Omega} \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \sum_{i=1}^{g} \langle \mathbf{x}, Y^i \rangle, \tag{12}$$

where Ω is defined as follows:

$$\Omega = \left\{ Y \in \mathbb{R}^{p \times g} : Y^i_{\overline{G_i}} = \mathbf{0}, \|Y^i\| \leq \lambda_2 w_i, i = 1, 2, \ldots, g \right\},$$

where $\overline{G_i}$ is the complementary set of G_i, Y is a sparse matrix satisfying $Y_{ij} = 0$ if the ith feature does not belong to the jth group, i.e., $i \notin G_j$, and Y^i denotes the ith column of Y. As a result, we can reformulate (11) as the following min-max problem:

$$\min_{\substack{\mathbf{x} \in \mathbb{R}^p \\ \mathbf{0} \preceq \mathbf{x} \preceq \mathbf{u}}} \max_{Y \in \Omega} \left\{ \psi(\mathbf{x}, Y) = \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \langle \mathbf{x}, Y\mathbf{e} \rangle \right\}, \tag{13}$$

where $\mathbf{e} \in \mathbb{R}^g$ is a vector of ones. It is easy to verify that $\psi(\mathbf{x}, Y)$ is convex in $\mathbf{x}$ and concave in Y, and the constraint sets are closed convex for both $\mathbf{x}$ and Y. Thus, (13) has a saddle point, and the min-max can be exchanged.

It is easy to verify that for a given Y, the optimal $\mathbf{x}$ minimizing $\psi(\mathbf{x}, Y)$ in (13) is given by

$$\mathbf{x} = \max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}). \tag{14}$$

Plugging (14) into (13), we obtain the following minimization problem with regard to Y:

$$\min_{Y \in \mathbb{R}^{p \times g} : Y \in \Omega} \{\omega(Y) = -\psi(\max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}), Y)\}. \tag{15}$$

Our methodology for minimizing $h_{\lambda_2}(\cdot)$, defined in (5), is to first solve (15) and then construct the solution to $h_{\lambda_2}(\cdot)$ via (14). Using standard optimization techniques, we can show that the function $\omega(\cdot)$ is continuously differentiable with Lipschitz continuous gradient. We include the detailed proof in Theorem 2 for completeness. Therefore, we convert the nonsmooth problem (11) to the smooth problem (15), making the smooth convex optimization tools applicable.

Theorem 2. *The function $\omega(Y)$ is convex and continuously differentiable with*

$$\omega'(Y) = -\max(\mathbf{u} - Y\mathbf{e}, \mathbf{0})\mathbf{e}^{\mathrm{T}}. \tag{16}$$

In addition, $\omega'(Y)$ is Lipschitz continuous with constant g, i.e.,

$$\|\omega'(Y_1) - \omega'(Y_2)\|_F \leq g\|Y_1 - Y_2\|_F, \quad \forall\ Y_1, Y_2 \in \mathbb{R}^{p \times g}. \tag{17}$$

To prove Theorem 2, we first present two technical lemmas. The first lemma is related to the optimal value function [24], [25], and it was used in a recent study [26] on infinite kernel learning.

Lemma 4 [24]. *Let X be a metric space and U be a normed space. Suppose that for all $\mathbf{x} \in X$, the function $\psi(\mathbf{x}, \cdot)$ is differentiable and that $\psi(\mathbf{x}, Y)$ and $D_Y\psi(\mathbf{x}, Y)$ (the partial derivative of $\psi(\mathbf{x}, Y)$ with respect to Y) are continuous on $X \times U$. Let Φ be a compact subset of X. Define the optimal value function as $\varphi(Y) = \inf_{\mathbf{x} \in \Phi} \psi(\mathbf{x}, Y)$. The optimal value function $\varphi(Y)$ is directionally differentiable. In addition, if $\forall Y \in U$, $\psi(\cdot, Y)$ has a unique minimizer $\mathbf{x}(Y)$ over Φ, then $\varphi(Y)$ is differentiable at Y and the gradient of $\varphi(Y)$ is given by $\varphi'(Y) = D_Y\psi(\mathbf{x}(Y), Y)$.*

The second lemma shows that the operator $\mathbf{y} = \max(\mathbf{x}, \mathbf{0})$ is nonexpansive.

Lemma 5. *$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, we have $\|\max(\mathbf{x}, \mathbf{0}) - \max(\mathbf{y}, \mathbf{0})\| \leq \|\mathbf{x} - \mathbf{y}\|$.*

Proof. The result follows because $|\max(x, 0) - \max(y, 0)| \leq |x - y|$, $\forall x, y \in \mathbb{R}$. □

Proof of Theorem 2. To prove the differentiability of $\omega(Y)$, we apply Lemma 4 with $X = \mathbb{R}^p$, $U = \mathbb{R}^{p \times g}$, and $\Phi = \{\mathbf{x} \in X : \mathbf{u} + \lambda_2 \sum w_i \mathbf{e} \geq \mathbf{x} \geq \mathbf{0}\}$. It is easy to verify that:

1. $\psi(\mathbf{x}, \cdot)$ is differentiable;
2. $\psi(\mathbf{x}, Y)$ and $D_Y\psi(\mathbf{x}, Y) = \mathbf{x}\mathbf{e}^{\mathrm{T}}$ are continuous on $X \times U$;
3. Φ is a compact subset of X; and
4. $\forall Y \in U$, $\psi(\mathbf{x}, Y)$ has a unique minimizer $\mathbf{x}(Y) = \max(\mathbf{u} - Y\mathbf{e}, \mathbf{0})$ over Φ.

Note that the last result follows from $\mathbf{u} > 0$ and $\mathbf{u} - Y\mathbf{e} \leq \mathbf{u} + \lambda_2 \sum w_i \mathbf{e}$, where the latter inequality utilizes $\|Y^i\| \leq \lambda_2 w_i$, and this indicates that $\mathbf{x}(Y) = \max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}) = \arg\min_{\mathbf{x}} \psi(\mathbf{x}, Y) = \arg\min_{\mathbf{x} \in \Phi} \psi(\mathbf{x}, Y)$. It follows from Lemma 4 that

$$\varphi(Y) = \inf_{\mathbf{x} \in \Phi} \psi(\mathbf{x}, Y) = \psi(\max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}), Y)$$

is differentiable with $\varphi'(Y) = \max(\mathbf{u} - Y\mathbf{e}, \mathbf{0})\mathbf{e}^{\mathrm{T}}$.

In (13), $\psi(\mathbf{x}, Y)$ is convex in $\mathbf{x}$ and concave in Y, and the constraint sets are closed convex for both $\mathbf{x}$ and Y; thus, the existence of the saddle point is guaranteed by the well-known von Neumann Lemma [18]. As a result,

$$\varphi(Y) = \inf_{\mathbf{x}\in\Phi} \psi(\mathbf{x}, Y) = \psi(\max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}), Y)$$

is concave and $\omega(Y) = -\varphi(Y)$ is convex. For any Y_1, Y_2, we have

$$\begin{aligned} &\|\omega'(Y_1) - \omega'(Y_2)\|_F \\ &\quad = \|\max(\mathbf{u} - Y_1\mathbf{e}, \mathbf{0})\mathbf{e}^{\mathrm{T}} - \max(\mathbf{u} - Y_2\mathbf{e}, \mathbf{0})\mathbf{e}^{\mathrm{T}}\|_F \\ &\quad \le \|\mathbf{e}\| \times \|\max(\mathbf{u} - Y_1\mathbf{e}, \mathbf{0}) - \max(\mathbf{u} - Y_2\mathbf{e}, \mathbf{0})\| \\ &\quad \le \|\mathbf{e}\| \times \|(Y_1 - Y_2)\mathbf{e}\| \\ &\quad \le g\|Y_1 - Y_2\|_F, \end{aligned} \tag{18}$$

where the second inequality follows from Lemma 5. We prove (17).

From Theorem 2, the problem in (15) is a constrained smooth convex optimization problem, and existing solvers for constrained smooth convex optimization can be applied. In this paper, we employ the AGD to solve (15) due to its fast convergence property. Note that the euclidean projection onto the set Ω can be computed in closed form. We would like to emphasize here that the problem (15) may have a much smaller size than (4).

3.2.1 Computing the Duality Gap

We show how to estimate the duality gap of the min-max problem (13), which can be used to check the quality of the solution and determine the convergence of the algorithm.

For any given approximate solution $\tilde{Y} \in \Omega$ for $\omega(Y)$, we can construct the approximate solution $\tilde{\mathbf{x}} = \max(\mathbf{u} - \tilde{Y}\mathbf{e}, \mathbf{0})$ for $h_{\lambda_2}(\mathbf{x})$. The duality gap for problem (13) at the point $(\tilde{\mathbf{x}}, \tilde{Y})$ can be computed as

$$\mathrm{gap}(\tilde{Y}) = \max_{Y\in\Omega} \psi(\tilde{\mathbf{x}}, Y) - \min_{\substack{\mathbf{x}\in\mathbb{R}^p \\ \mathbf{0}\preceq\mathbf{x}\preceq\mathbf{u}}} \psi(\mathbf{x}, \tilde{Y}). \tag{19}$$

The main result of this section is summarized in the following theorem.

Theorem 3. *Let* $\mathrm{gap}(\tilde{Y})$ *be the duality gap defined in (19). Then, the following holds:*

$$\mathrm{gap}(\tilde{Y}) = \sum_{i=1}^{g} (\lambda_2 w_i \|\tilde{\mathbf{x}}_{G_i}\| - \langle \tilde{\mathbf{x}}_{G_i}, \tilde{Y}^i_{G_i}\rangle). \tag{20}$$

In addition, we have

$$\omega(\tilde{Y}) - \omega(Y^*) \le \mathrm{gap}(\tilde{Y}), \tag{21}$$

$$h(\tilde{\mathbf{x}}) - h(\mathbf{x}^*) \le \mathrm{gap}(\tilde{Y}). \tag{22}$$

Proof. Denote $(\mathbf{x}^*, Y^*)$ as the optimal solution to the min-max problem (13). From (12)-(15), we have

$$-\omega(\tilde{Y}) = \psi(\tilde{\mathbf{x}}, \tilde{Y}) = \min_{\substack{\mathbf{x}\in\mathbb{R}^p \\ \mathbf{0}\preceq\mathbf{x}\preceq\mathbf{u}}} \psi(\mathbf{x}, \tilde{Y}) \le \psi(\mathbf{x}^*, \tilde{Y}), \tag{23}$$

$$\psi(\mathbf{x}^*, \tilde{Y}) \le \max_{Y\in\Omega} \psi(\mathbf{x}^*, Y) = \psi(\mathbf{x}^*, Y^*) = -\omega(Y^*), \tag{24}$$

$$h_{\lambda_2}(\mathbf{x}^*) = \psi(\mathbf{x}^*, Y^*) = \min_{\substack{\mathbf{x}\in\mathbb{R}^p \\ \mathbf{0}\preceq\mathbf{x}\preceq\mathbf{u}}} \psi(\mathbf{x}, Y^*) \le \psi(\tilde{\mathbf{x}}, Y^*), \tag{25}$$

$$\psi(\tilde{\mathbf{x}}, Y^*) \le \max_{Y\in\Omega} \psi(\tilde{\mathbf{x}}, Y) = h_{\lambda_2}(\tilde{\mathbf{x}}). \tag{26}$$

Incorporating (11), (23)-(26), we prove (20)-(22). □

In our experiments, we terminate the algorithm when the estimated duality gap is less than 10^{-10}.

3.3 Proximal Splitting Methods

Recently, a family of proximal splitting methods [27] has been proposed for converting a challenging optimization problem into a series of subproblems with a closed-form solution. We consider two reformulations of the proximal operator (4), based on the Dykstra-like Proximal Splitting Method and the ADMM. The efficiency of these two methods for overlapping group Lasso will be demonstrated in the next section.

3.3.1 Dykstra-Like Proximal Splitting Method

In the field of signal processing, one classical problem is the *convex feasibility problem*:

$$\text{find } x \in \bigcap_{i=1}^{m} C_i, \tag{27}$$

where C_is are convex sets. Efficient methods have been designed for (27), where at each iteration only one convex set is considered and the solution is updated iteratively by cycling through all convex sets. Under certain conditions, convergence is guaranteed. For our problem, since (5) can be considered as the projection of a vector $\mathbf{u}$ onto a collection of convex sets induced by the regularization components $w_i\|\mathbf{x}_{G_i}\|$, the proximal splitting ideas can be applied.

We define $f_i = \lambda\|\mathbf{x}_{G_i}\|$; the proximal operator in (5) can be rewritten as:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \sum_{i=1}^{g} w_i f_i. \tag{28}$$

Then, the Dykstra-like proximal algorithm can be summarized in Algorithm 2.

Algorithm 2. Dykstra-like Proximal Splitting Method

```
1: Set $\mathbf{x}_0 = \mathbf{u}$, $\mathbf{q}_{1,0}, \ldots, \mathbf{q}_{g,0} = \mathbf{x}_0$, $n = 0$
2: repeat
3:     for $i = 1, \ldots, g$ do
4:         $\mathbf{p}_{i,n} = \mathrm{prox}_{f_i}\mathbf{q}_{i,n}$
5:     end for
6:     $\mathbf{x}_{n+1} = \sum_{i=1}^{g} w_i\mathbf{p}_{i,n}$
7:     for $i = 1, \ldots, g$ do
8:         $\mathbf{q}_{i,n+1} = \mathbf{x}_{n+1} + \mathbf{q}_{i,n} - \mathbf{p}_{i,n}$
9:     end for
10:    $n = n + 1$
11: until Convergence
```

The last piece of the puzzle in Algorithm 2 is to solve $\mathbf{p} = \mathrm{prox}_{f_i}\mathbf{q}$, defined as:

$$\mathbf{p} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{q}\|^2 + \lambda\|\mathbf{x}_{G_i}\|.$$

Clearly, we have $\mathbf{p}_{\overline{G}_i} = \mathbf{q}_{\overline{G}_i}$. For index set G_i, a closed-form solution is known to exist:

$$\mathbf{p}_{G_i} = \frac{\max(\|\mathbf{q}_{G_i}\| - \lambda, 0)}{\|\mathbf{q}_{G_i}\|}\mathbf{q}_{G_i}.$$

3.3.2 Alternating Direction Method of Multipliers

Besides splitting the proximal operators, we can also bypass the difficulty brought by overlapping groups by introducing auxiliary variables, and reformulate (5) as

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}} \quad & \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda\sum_{i=1}^{g} w_i\|\mathbf{z}_i\| \\ \text{s.t.} \quad & \mathbf{z}_i = \mathbf{x}_{G_i}, \quad i=1,\ldots,g.\end{aligned} \tag{29}$$

We can therefore form the augmented Lagrangian as follows:

$$\begin{aligned}L_\rho(\mathbf{x},\mathbf{z},\mathbf{y}) = & \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda\sum_{i=1}^{g} w_i\|\mathbf{z}_i\| \\ & + \sum_{i=1}^{g}\mathbf{y}_i^T(\mathbf{z}_i - \mathbf{x}_{G_i}) + \frac{\rho}{2}\sum_{i=1}^{g}\|\mathbf{z}_i - \mathbf{x}_{G_i}\|^2.\end{aligned}$$

The ADMM consists of the following iterations:

$$\begin{aligned}\mathbf{x}^{k+1} &:= \arg\min_{\mathbf{x}} L_\rho(\mathbf{x},\mathbf{z}^k,\mathbf{y}^k), \\ \mathbf{z}^{k+1} &:= \arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{k+1},\mathbf{z},\mathbf{y}^k), \\ \mathbf{y}_i^{k+1} &:= \mathbf{y}_i^k + \rho(\mathbf{z}_i^{k+1} - \mathbf{x}_{G_i}^{k+1}).\end{aligned} \tag{30}$$

One nice property of ADMM is each iterative step admits a closed-form solution. We define $\oslash$ as the point-wise division, $\mathbf{e}$ the p-dimensional vector with all ones, and the indicator vector $\tilde{\mathbf{e}}_i$ such that $\tilde{\mathbf{e}}_i(j)=1$ if $j\in G_i$ and 0 otherwise. We further define $\tilde{\mathbf{y}}_i, \tilde{\mathbf{z}}_i \in \mathbb{R}^p$ such that $\tilde{\mathbf{y}}_i(G_i) = \mathbf{y}_i, \tilde{\mathbf{y}}_i(G_i^C)=0$ and $\tilde{\mathbf{z}}_i(G_i) = \mathbf{z}_i, \tilde{\mathbf{z}}_i(G_i^C) = 0$. For updating $\mathbf{x}$, we have:

$$\begin{aligned}\frac{\partial}{\partial\mathbf{x}}L_\rho(\mathbf{x},\mathbf{z}^k,\mathbf{y}^k) = & \mathbf{x} - \mathbf{u} - \sum_{i=1}^{g}\tilde{\mathbf{y}}_i^k + \rho\left(\sum_{i=1}^{g}\tilde{\mathbf{e}}_i\right)\odot\mathbf{x} \\ & -\rho\left(\sum_{i=1}^{g}\tilde{\mathbf{z}}_i^k\right),\end{aligned}$$

and therefore

$$\mathbf{x}^{k+1} = \left(\mathbf{u} + \sum_{i=1}^{g}\tilde{\mathbf{y}}_i^k + \rho\sum_{i=1}^{g}\tilde{\mathbf{z}}_i^k\right)\oslash\left(\mathbf{e} + \rho\sum_{i=1}^{g}\tilde{\mathbf{e}}_i\right).$$

For updating $\mathbf{z}_i$, we use the subdifferential method: $\mathbf{z}^*$ is the optimal solution if and only if 0 belongs to the subdifferential set $\partial L_\rho(\mathbf{x}^{k+1},\mathbf{z}^*,\mathbf{y}^k)$. Decoupling the problem with respect to groups, we have:

$$0 \in \mathbf{z}_i^{k+1} - \mathbf{x}_{G_i}^{k+1} + \frac{1}{\rho}\mathbf{y}_i^k + \frac{\lambda w_i}{\rho}\partial\|\mathbf{z}_i^{k+1}\|,$$

where

$$\partial\|\mathbf{z}_i^{k+1}\| = \begin{cases}\frac{\mathbf{z}_i^{k+1}}{\|\mathbf{z}_i^{k+1}\|} & \|\mathbf{z}_i^{k+1}\| \neq 0 \\ \{\mathbf{t} | \mathbf{t}\in\mathbb{R}^{|G_i|}, \|\mathbf{t}\|\leq 1\} & \|\mathbf{z}_i^{k+1}\| = 0.\end{cases}$$

Thus, we have:

$$\mathbf{z}_i^{k+1} = \frac{\max\{\|\tilde{\mathbf{x}}_{G_i}^{k+1}\| - \tilde{\lambda}_i, 0\}}{\|\tilde{\mathbf{x}}_{G_i}^{k+1}\|}\tilde{\mathbf{x}}_{G_i}^{k+1},$$

where

$$\tilde{\mathbf{x}}_{G_i}^{k+1} = \mathbf{x}_{G_i}^{k+1} - \frac{1}{\rho}\mathbf{y}_i^k, \quad \tilde{\lambda}_i = \frac{\lambda w_i}{\rho}.$$

Reformulation that uses ADMM to solve (1). Boyd et al. [14] suggested that the original overlapping group lasso problem (1) can be reformulated and solved by ADMM directly. We include the implementation of ADMM in our comparative study, and the details are provided in the appendix, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.17, for completeness.

4 ℓ_q Norm Overlapping Group Lasso

In this section, we extend our previous results to solving the overlapping group lasso formulation (1) based on the ℓ_q norm with $q > 1$. Specifically, we extend the group lasso penalty (2) to

$$\phi_{q,\lambda_2}^{\lambda_1}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2\sum_{i=1}^{g} w_i\|\mathbf{x}_{G_i}\|_q. \tag{31}$$

To extend to the ℓ_q norm case, the only change to Algorithm 1 is to generalize the proximal operator:

$$\pi_{q,\lambda_2}^{\lambda_1}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\left\{g_{q,\lambda_2}^{\lambda_1}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \phi_{q,\lambda_2}^{\lambda_1}(\mathbf{x})\right\}. \tag{32}$$

In the rest of the section, we extend the properties of the proximal operator as well as the dual method from the ℓ_2 norm case to the general ℓ_q norm case.

4.1 Properties of the ℓ_q Proximal Operator

First of all, it is easy to verify that Lemma 1 and Theorem 1 hold for all $q > 1$ given that the ℓ_q norm is convex.

Denote the dual norm of the ℓ_q norm as $\ell_{\overline{q}}$ with $1/q + 1/\overline{q} = 1$ and $h_{q,\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \phi_{q,\lambda_2}^{0}(\mathbf{x})$. We then extend the preprocessing techniques in the following two lemmas.

Lemma 6. *Denote the minimizer of $h_{q,\lambda_2}(\cdot)$ by $\mathbf{x}^*$. If the ith group satisfies $\|\mathbf{u}_{G_i}\|_{\overline{q}} \leq \lambda_2 w_i$, then $\mathbf{x}_{G_i}^* = \mathbf{0}$, i.e., the ith group is zero.*

Proof. We decompose $h_{q,\lambda_2}(\mathbf{x})$ into two parts as follows:

$$\begin{aligned}h_{q,\lambda_2}(\mathbf{x}) = & \left(\frac{1}{2}\|\mathbf{x}_{G_i} - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i\|\mathbf{x}_{G_i}\|_q\right) \\ & + \left(\frac{1}{2}\|\mathbf{x}_{\overline{G}_i} - \mathbf{u}_{\overline{G}_i}\|^2 + \lambda_2\sum_{j\neq i} w_j\|\mathbf{x}_{G_j}\|_q\right),\end{aligned} \tag{33}$$

where $\overline{G}_i = \{1,2,\ldots,p\} - G_i$ is the complementary set of G_i. We consider the minimization of $h_{q,\lambda_2}(\mathbf{x})$ in terms of $\mathbf{x}_{G_i}$ when $\mathbf{x}_{\overline{G}_i} = \mathbf{x}_{\overline{G}_i}^*$ is fixed.

Clearly, $\mathbf{x}_{\overline{G}_i}^* = 0$ minimizes the second term in (33). Therefore, we just need to show that $\mathbf{x}_{G_i}^* = 0$ minimizes $y(\mathbf{x}_{G_i})$ defined as the following:

$$y(\mathbf{x}_{G_i}) = \frac{1}{2}\|\mathbf{x}_{G_i} - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i\|\mathbf{x}_{G_i}\|_q.$$

Given any direction $d\in\mathbb{R}^{|G_i|}$, we take the directional derivative of y at point $\mathbf{0}$:

$$\begin{aligned} Dy(\mathbf{0})[d] &= \lim_{\alpha\downarrow 0} \frac{y(\alpha d) - y(\mathbf{0})}{\alpha} \\ &= \lim_{\alpha\downarrow 0} \frac{\frac{1}{2}\|\alpha d - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i \|\alpha d\|_q - \frac{1}{2}\|\mathbf{u}_{G_i}\|^2}{\alpha} \\ &= -\langle d, \mathbf{u}_{G_i}\rangle + \lambda_2 w_i \|d\|_q \\ &\ge -\|d\|_q \|\mathbf{u}_{G_i}\|_{\bar{q}} + \lambda_2 w_i \|d\|_q \\ &\ge 0 \quad \forall d, \end{aligned}$$

where the last inequality follows because $\|\mathbf{u}_{G_i}\|_{\bar{q}} \le \lambda_2 w_i$. Thus, $\mathbf{x}^*_{G_i} = 0$. □

Similarly to Lemma 3, we have:

Lemma 7. *Denote the minimizer of* $h_{q,\lambda_2}(\cdot)$ *by* $\mathbf{x}^*$. *Let* S_i*, a subset of* G_i*, be defined in (10). If* $\|\mathbf{u}_{G_i - S_i}\|_{\bar{q}} \le \lambda_2 w_i$ *holds, then* $\mathbf{x}^*_{G_i} = \mathbf{0}$.

The proof is similar to Lemma 3 based on the result in Lemma 6.

4.2 Extending the Dual Method to the ℓ_q Case

When reformulating (32) as an equivalent smooth problem, we only need to make two changes:

- The feasible region of the dual variable Y is generalized as:

$$\Omega_q = \{Y \in \mathbb{R}^{p\times g} : Y^i_{\overline{G_i}} = \mathbf{0}, \|Y^i\|_{\bar{q}} \le \lambda_2 w_i, \\ i = 1, 2, \ldots, g\},$$

- During the optimization process, we need to compute the euclidean projection onto the ℓ_q ball, which can be calculated efficiently [28].

The duality gap is now calculated as

$$\mathrm{gap}_p(\tilde{Y}) = \sum_{i=1}^{g} (\lambda_2 w_i \|\tilde{\mathbf{x}}_{G_i}\|_q - \langle \tilde{\mathbf{x}}_{G_i}, \tilde{Y}^i_{G_i}\rangle).$$

It is easy to verify that this value can still be used to check the convergence of our proposed dual method in the ℓ_q norm case.

5 Overlapping Group Lasso via the Capped Norm

In this section, we consider the following problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} l(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_0 + \lambda_2 \sum_{i=1}^{g} w_i I(\|\mathbf{x}_{G_i}\| \neq 0), \tag{34}$$

where $I(\cdot)$ is the indicator function. Note that this is an NP-hard problem, where convex relaxation such as (1) is normally applied. However, due to the looseness of convex relaxation, the ℓ_1-norm type regularization will introduce bias to the parameter estimation [29], [30]. Several recent works use the nonconvex capped norm that is closer to the ℓ_0 norm than the ℓ_1 norm as follows [29], [30], [31]:

$$\begin{aligned} \|\mathbf{x}\|_0 &\approx \sum_{j=1}^{p} \min\left(1, \frac{|\mathbf{x}_j|}{\theta_1}\right), \\ \sum_{i=1}^{g} w_i I(\|\mathbf{x}_{G_i}\| \neq 0) &\approx \sum_{i=1}^{g} w_i \min\left(1, \frac{\|\mathbf{x}_{G_i}\|}{\theta_2}\right), \end{aligned} \tag{35}$$

for some small $\theta_1, \theta_2 > 0$. Parameter estimation using the nonconvex capped norm has been studied. It has been shown that under appropriate conditions, the local solution obtained by using the capped norm has better statistical property than the one based on the convex ℓ_1 norm penalty.

The approximation used in (35) is still nonconvex. Following [29], [30], [31], we use the following two decompositions:

$$\begin{aligned} &\sum_{j=1}^{p} \min\left(1, \frac{|\mathbf{x}_j|}{\theta_1}\right) \\ &\quad = \frac{1}{\theta_1}\left[\|\mathbf{x}\|_1 - \sum_{j=1}^{p} \max(|\mathbf{x_j}| - \theta_1, 0)\right] \end{aligned} \tag{36}$$

and

$$\begin{aligned} &\sum_{i=1}^{g} w_i \min\left(1, \frac{\|\mathbf{x}_{G_i}\|}{\theta_2}\right) \\ &\quad = \frac{1}{\theta_2}\left[\sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| - \sum_{i=1}^{g} w_i \max(\|\mathbf{x}_{G_i}\| - \theta_2, 0)\right]. \end{aligned} \tag{37}$$

Combining (35), (36), and (37), we can approximate (34) as

$$\min_{\mathbf{x}\in\mathbb{R}^p} l(\mathbf{x}) + \frac{\lambda_1}{\theta_1}\|\mathbf{x}\|_1 + \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| - P(\mathbf{x}) - D(\mathbf{x}), \tag{38}$$

where

$$P(\mathbf{x}) = \frac{\lambda_1}{\theta_1}\sum_{j=1}^{p} \max(|\mathbf{x}_j| - \theta_1, 0)$$

and

$$\begin{aligned} D(\mathbf{x}) &= \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i D_i(\mathbf{x}_{G_i}) \\ &= \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i \max(\|\mathbf{x}_{G_i}\| - \theta_2, 0). \end{aligned}$$

Note that both P and D are convex functions, and therefore we have converted the problem into a "difference of two convex functions" (DC) programming.

It can be shown that

$$\frac{\partial}{\partial \mathbf{x}_j} P(\mathbf{x}) \ni \begin{cases} \frac{\lambda_1}{\theta_1}\mathrm{sgn}(\mathbf{x}_j) & |\mathbf{x}_j| > \theta_1 \\ 0 & |\mathbf{x}_j| \le \theta_1 \end{cases}$$

and

$$\frac{\partial}{\partial \mathbf{x}_{G_i}} D_i(\mathbf{x}_{G_i}) \ni \begin{cases} \frac{\mathbf{x}_{G_i}}{\|\mathbf{x}_{G_i}\|} & \|\mathbf{x}_{G_i}\| > \theta_2 \\ \mathbf{0} & \|\mathbf{x}_{G_i}\| \le \theta_2. \end{cases}$$

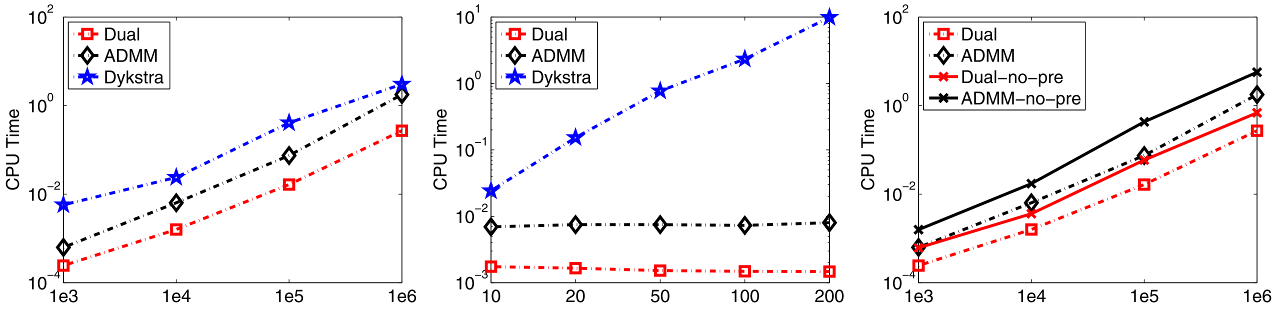

Fig. 1. Time comparison for computing the proximal operators. The group number is fixed in the left figure and the problem size is fixed in the middle figure. The right figure illustrates the effectiveness of the preprocessing.

We then propose to solve (38) using the DC programming, and the details are provided in Algorithm 3.

Algorithm 3. DC Programming for Overlapping Group Lasso with the Capped Norm
Input: $\theta_1, \theta_2 > 0, \mathbf{x}^0, k$
Output: $\mathbf{x}^{k+1}$

```
1: Initialize $\mathbf{x}^1 = \mathbf{x}^0$
2: for $i = 1$ to $k$ do
3:     Choose $U^k \in \partial P(\mathbf{x}^k)$ and $V^k \in \partial D(\mathbf{x}^k)$
4:     Solve
```

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\left\{ l(\mathbf{x}) + \frac{\lambda_1}{\theta_1}\|\mathbf{x}\|_1 + \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i\|\mathbf{x}_{G_i}\| - \langle U^k + V^k, \mathbf{x}\rangle \right\} \tag{39}$$

```
5: Set $k \leftarrow k + 1$
6: end for
```

The subproblem (39) can be solved using Algorithm 1. Therefore, by solving a sequence of overlapping group lasso problems, we can find a local solution for (38).

6 Experiments

In this section, we present extensive experiments to demonstrate the efficiency of our proposed methods. We use both synthetic datasets and a real-world dataset, and the evaluation is done in various problem size and precision settings. The proposed algorithms are mainly implemented in Matlab, with the proximal operator implemented in standard C for improved efficiency. The source codes can be found online [22].

Several state-of-the-art methods are also included for comparison purposes, including SLasso developed by Jenatton et al. [13] (with key components implemented in C), the ADMM reformulation suggested by Boyd et al. [14], the Prox-Grad method proposed by Chen et al. [16], and the Picard-Nesterov algorithm [15].

6.1 Synthetic Data

6.1.1 Efficiency of Calculating the Proximal Operator

In the first set of simulation, we consider only the key component of our algorithm, the proximal operator. The group indices are predefined such that $G_1 = \{1, 2, \ldots, 10\}$, $G_2 = \{6, 7, \ldots, 20\}, \ldots$, with each group overlapping half of the previous group. The target vector $\mathbf{v} \in \mathbb{R}^p$ in (4) is generated randomly such that $\mathbf{v}_i \sim N(0, 1)$. We fix $\lambda_1 = 1$ and $\lambda_2 = 10$.

One hundred examples are generated for each set of fixed problem size p and group size g, and for each particular random example we first run the dual method till the gap is less than 10^{-8}, then we run ADMM and the Dykstra method until a smaller function value is attained. The results are summarized in Fig. 1. As we can observe from the figure, the dual formulation yields the best performance, followed closely by ADMM, and then the Dykstra method. We can also observe that our method scales very well to high-dimensional problems because even with $p = 10^6$, the proximal operator can be computed in a few seconds. It is also not surprising that the Dykstra method is much more sensitive to the number of groups, which equals the number of projections in one Dykstra step.

To illustrate the effectiveness of our preprocessing technique, we repeat the previous experiment by removing the preprocessing step. The results are shown in the right plot of Fig. 1. As we can observe from the figure, the proposed preprocessing technique effectively reduces the computational time. As is evident from Fig. 1, the dual formulation proposed in Section 3.2 consistently outperforms other proximal splitting methods. In the following experiments, only the dual method with the preprocessing step will be used for computing the proximal operator, and our method will then be called as "FoGLasso."

6.1.2 Sparse Pattern Recovery

Although the focus of this paper is on the efficiency of the proposed algorithm, it is also interesting to see if the overlapping group Lasso formulation can recover the underlying sparse pattern. For a given problem size n, p, and group size g, we first define the overlapping groups as in Section 6.1.1. We then generate the ground-truth model $\mathbf{x}_0$ with each entry sampled i.i.d. from a standard Gaussian distribution. Next, we randomly set half of the predefined groups and half of the remaining entries of $\mathbf{x}_0$ to be 0. We sample the entries of the data matrix $A \in \mathbb{R}^{n\times p}$ i.i.d. from a standard Gaussian distribution, and the response vector $\mathbf{b}$ is obtained from $\mathbf{b} = A\mathbf{x}_0 + \epsilon$, where $\epsilon \sim N(0, \sigma^2 I_{n\times n})$.

We solve (1) with the least squares loss $l(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2$, and we set $w_i = \sqrt{|G_i|}$ and $\lambda_1 = \lambda_2 = \gamma \times \lambda_1^{\max}$, where $|G_i|$ denotes the size of the ith group G_i, $\lambda_1^{\max} = \|A^{\mathrm{T}}\mathbf{b}\|_\infty$ (the zero point is a solution to (1) if $\lambda_1 \geq \lambda_1^{\max}$), and γ is chosen from the set $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. Denote the

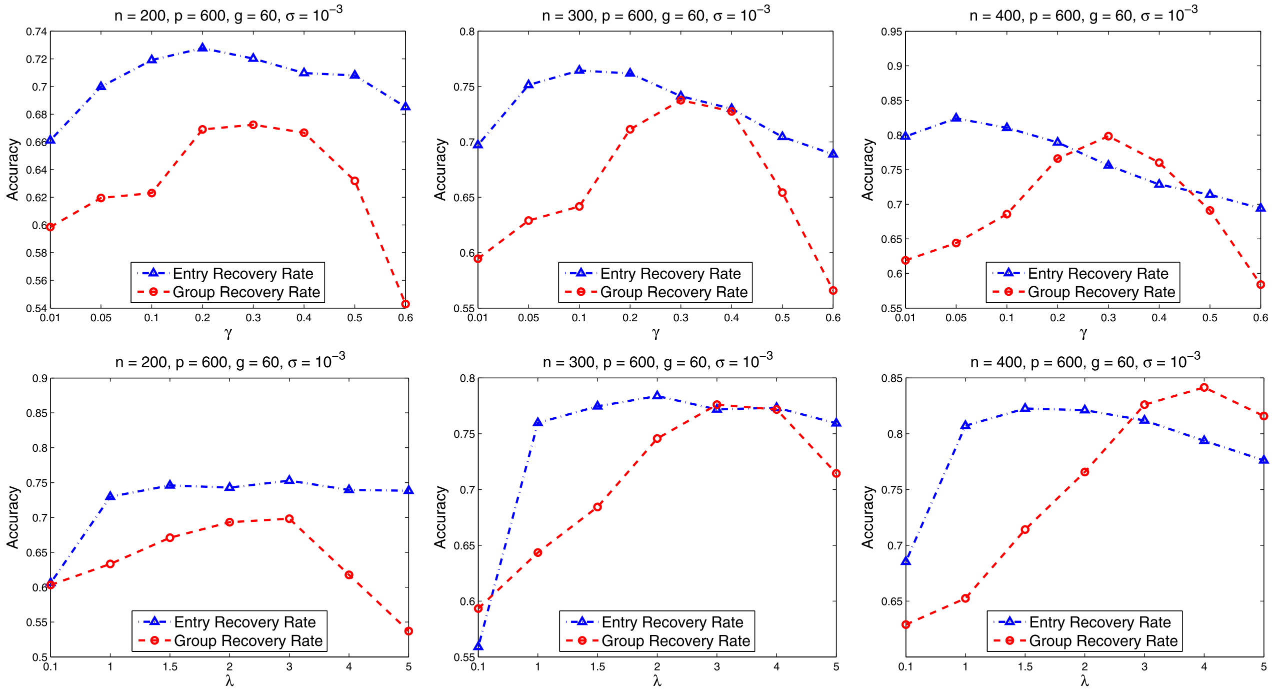

Fig. 2. Performance of sparse pattern recovery of the convex overlapping group Lasso formulation (1) (top row) and the nonconvex overlapping group Lasso formulation with the capped norm (38) (bottom row) on synthetic data with different problem sizes.

obtained solution as $\mathbf{x}$. The following two criteria are used to evaluate the recovery performance:

- Entry recovery rate:

  $$\Pr\{\|\mathbf{x}_0(i)\|_0 = \|\mathbf{x}(i)\|_0\},$$

  which is the entry-wise accuracy of sparse pattern recovery.
- Group recovery rate:

  $$\Pr\{I(\|\mathbf{x}_0(G_i)\| = 0) = I(\|\mathbf{x}(G_i)\| = 0)\},$$

  where $I(\cdot)$ is the indicator function. This can be considered as the group-wise accuracy of sparse pattern recovery.

We set $n \in \{200, 300, 400\}$, $p = 600$, $\sigma = 10^{-3}$, and $g = 60$. For each γ value, 100 random instances are generated and the average performance for different problem sizes is reported in the top row of Fig. 2.

Using similar problem settings, we also evaluate the recovery performance of the overlapping group Lasso formulation with the capped norm. We set $\theta_1 = \theta_2 = 0.01$, and instead of using a ratio, we set $\lambda_1 = \lambda_2 = \lambda \in \{0.1, 1, 1.5, 2, 3, 4, 5\}$. The results are summarized in the bottom row of Fig. 2.

We can observe from Fig. 2 that as we increase the sample size, the performance generally improves. The best performance is normally attained in the middle of the parameter space, where the solution is not too dense (mostly nonzeros) or too sparse (mostly zeros). Our preliminary evaluation shows that using the nonconvex formulation indeed improves the pattern recovery rate compared to the original formulation. For example, when the sample size is 400, the best group recovery rate for the original formulation is 0.81, while for the formulation with the capped norm, the rate is about 0.85.

Fig. 2 illustrates the recovery performance across the parameter space. Here, we also provide results with parameters selected via cross validation, which is usually done in practice. For each randomly generated example, we first use fourfold cross validation to select the parameter with the smallest error. We then use this parameter to obtain the model $\mathbf{x}$ and compare it to the ground truth to obtain recovery performance. We repeat this process 100 times and the results are summarized in Table 1. As we can see in Table 1, using the nonconvex formulation also improves the pattern recovery rate when the parameters are selected using cross validation. Further evaluation of this nonconvex formulation in real-world applications will be our future work.

6.2 Gene Expression Data

We have also conducted experiments to evaluate the efficiency of the proposed algorithm using the breast cancer gene expression dataset [32], which consists of 8,141 genes in 295 breast cancer tumors (78 metastatic and 217 nonmetastatic). For the sake of analyzing microarrays in terms of biologically meaningful gene sets, different approaches have been used to organize the genes into (overlapping)

TABLE 1
Cross-Validation Performance of Sparse Pattern Recovery of the Convex Overlapping Group Lasso Formulation and the Nonconvex Overlapping Group Lasso Formulation Based on the Capped Norm on Synthetic Data with Different Problem Sizes

	Convex		Non-convex	
n	Entry Rate	Group Rate	Entry Rate	Group Rate
300	0.71	0.60	0.77	0.71
400	0.80	0.61	0.82	0.70

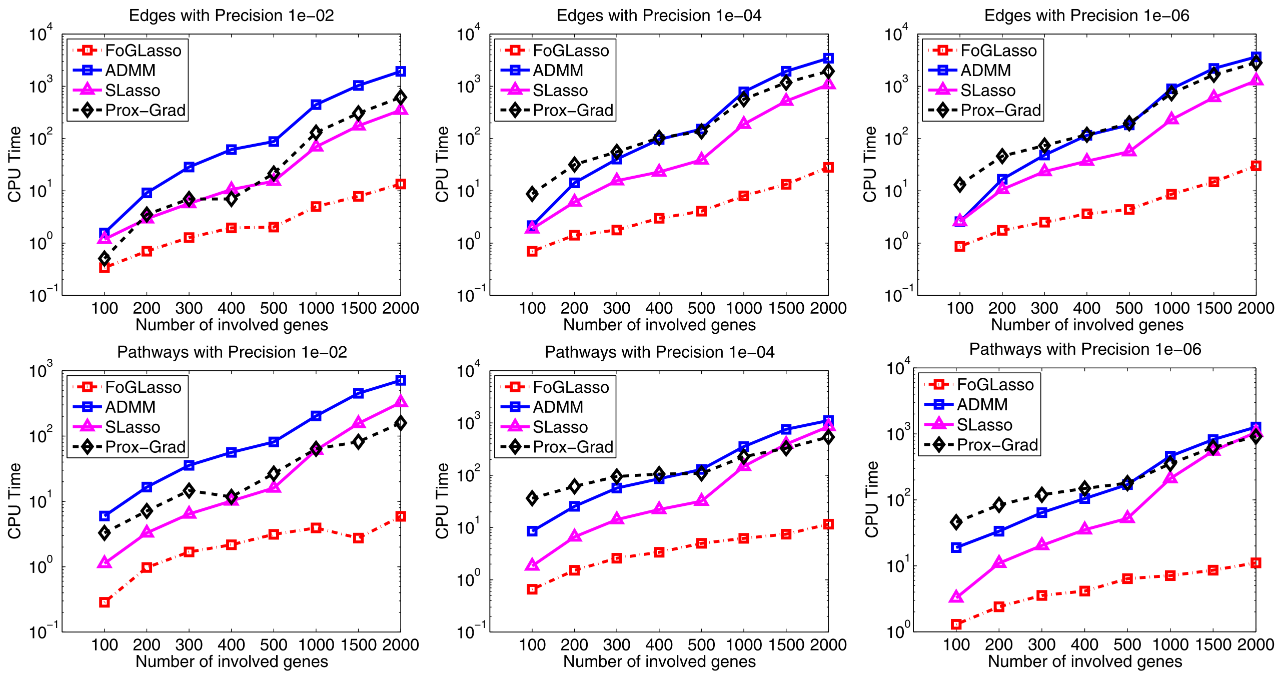

Fig. 3. Comparison of SLasso [13], ADMM [14], Prox-Grad [16], and our proposed FoGLasso algorithm in terms of computational time (in seconds and in the logarithmic scale) when different numbers of genes (variables) are involved. Different precision levels are used for comparison.

gene sets. In our experiments, we follow Jacob et al. [6] and employ the following two approaches for generating the overlapping gene sets (groups): pathways [33] and edges [34]. For pathways, the canonical pathways from the Molecular Signatures Database (MSigDB) [33] are used. It contains 639 groups of genes, of which 637 groups involve the genes in our study. The statistics of the 637 gene groups are summarized as follows: The average number of genes in each group is 23.7, the largest gene group has 213 genes, and 3,510 genes appear in these 637 groups with an average appearance frequency of about 4. For edges, the network built by Chuang et al. [34] will be used, and we follow Jacob et al. [6] to extract 42,594 edges from the network, leading to 42,594 overlapping gene sets of size 2. All 8,141 genes appear in the 42,594 groups with an average appearance frequency of about 10. Here, we set $\lambda_1 = \lambda_2 = \gamma \times \lambda_1^{\max}$, where γ is chosen from the set

$$\{5 \times 10^{-1}, 2 \times 10^{-1}, 1 \times 10^{-1}, 5 \times 10^{-2}, 2 \times 10^{-2}, 1 \times 10^{-2}, 5 \times 10^{-3}, 2 \times 10^{-3}, 1 \times 10^{-3}\}.$$

6.2.1 Comparison with SLasso, Prox-Grad, and ADMM

We first compare our proposed FoGLasso with the SLasso algorithm [13], ADMM [14], and Prox-Grad [16]. The comparisons are based on the computational time because all these methods have efficient Matlab implementations with key components written in C. For a given γ, we first run SLasso till a certain precision level is reached, and then run the others until they achieve an objective function value smaller than or equal to that of SLasso. The precision level here is used as the convergence condition for SLasso such that when the change of objective function value is smaller than a certain value, the algorithm terminates. Different precision levels of the solutions are evaluated such that a fair comparison can be made. We vary the number of genes involved and report the total computational time (seconds) for all nine regularization parameters in Fig. 3. We can observe that: 1) For all precision levels, our proposed FoGLasso is much more efficient than SLasso, ADMM and Prox-Grad; 2) the advantage of FoGLasso over other three methods in efficiency grows with the increasing number of genes (variables), for example, with the grouping by pathways, FoGLasso is about 25 and 70 times faster than SLasso for 1,000 and 2,000 genes, respectively; and 3) the efficiency on edges is inferior to that on pathways due to the larger number of overlapping groups. An additional scalability study of our proposed method using larger problem sizes can be found in Table 2.

6.2.2 Comparison with Picard-Nesterov

Since Picard-Nesterov was implemented purely in Matlab, a computational time comparison might not be fair. Therefore, only the number of iterations required for convergence is reported, as both methods adopt the first order method. Also note that, unlike the previous three methods (SLasso, ADMM, and Prox-Grad), the preprocessing technique can be applied to Picard-Nesterov because it solves the same proximal operator in each iteration. To further validate the

TABLE 2
Scalability Study of the FoGLasso Algorithm under Different Numbers (p) of Genes Involved

p	3000	4000	5000	6000	7000	8141
pathways	37.6	48.3	62.5	68.7	86.2	99.7
edges	58.8	84.8	102.7	140.8	173.3	247.8

The reported results are the total computational time (seconds) for all nine regularization parameter values.

TABLE 3
Comparison of FoGLasso, Picard-Nesterov, and Picard-Nesterov with Our Proposed Preprocessing Technique Using Different Numbers (p) of Genes and Various Precision Levels

Precision Level	10^{-2}			10^{-4}			10^{-6}		
p	100	200	400	100	200	400	100	200	400
FoGLasso	81	189	353	192	371	1299	334	507	1796
	288	401	921	404	590	1912	547	727	2387
Picard-Nesterov	78	176	325	181	304	1028	318	504	1431
	8271	6.8e4	2.2e5	2.6e4	1.0e5	7.8e5	5.1e4	1.3e5	1.1e6
Picard-Nesterov-PreProc	78	176	325	181	304	1028	318	504	1431
	2683	3.8e4	1.1e5	8427	6.4e4	4.9e5	1.9e4	8.2e4	7.3e5

For each particular method, the first row denotes the number of outer iterations required for convergence, while the second row represents the total number of inner iterations.

effectiveness of our preprocessing technique, we apply it to Picard-Nesterov as an independent method for comparison.

We use edges to generate the groups, and vary the problem size from 100 to 400 using the same set of regularization parameters. For each problem, we record both the number of outer iterations (the gradient steps) and the total number of inner iterations (the steps required for computing the proximal operators). The average number of iterations among all the regularization parameters is summarized in Table 3. As we can observe from the table, though Picard-Nesterov often takes less outer iterations to converge, it takes a lot more inner iterations to compute the proximal operator. It is easy to verify that the inner iterations in the Picard-Nesterov method and our proposed method have the same complexity of $O(pg)$. In terms of the preprocessing technique, we can see that in all cases the number of gradient steps remains exactly the same, while the number of inner iterations is significantly reduced. This verifies that by using the proposed preprocessing technique, the proximal operator yields the same solution while solving a smaller problem.

6.2.3 Computation of the Proximal Operator

In this experiment, we run FoGLasso on the breast cancer dataset using all 8,141 genes. We terminate FoGLasso if the change of the objective function value is less than 10^{-5}. We use the 42,594 edges to generate the overlapping groups. We set $\rho = 0.01$. The results are shown in Fig. 4. The left plot shows that the objective function value decreases rapidly in the proposed FoGLasso. In the middle plot, we report the percentage of the identified zero groups by applying Lemma 3. Our result shows that: 1) After 16 iterations, 50 percent of the zero groups are correctly identified; and 2) after 50 iterations, 80 percent of the zero groups are identified. Therefore, with Lemma 3, we can significantly reduce the problem size of the subsequent dual reformulation (see Section 3.1). In the right plot of Fig. 4, we present the number of inner iterations for solving the proximal operator via the dual reformulation. We observe from the figure that the number of inner iterations decreases. This is because: 1) The size of the reduced problem decreases when many zero groups are identified (see the middle plot); and 2) in solving the dual reformulation, we can apply the Y computed in the previous iteration as the "warm" start for computing the proximal operator in the next iteration.

6.2.4 Convergence with Inexact Proximal Operator

With an inexact proximal operator, the optimal convergence rate of the AGD might not be guaranteed [35], [36]. However, recent work [37] has shown that if the error introduced in the proximal operator decreases at a certain rate, the convergence rate of AGD remains the same as in the exact case. Specifically, if we denote the duality gap in the k step as $\epsilon_k = \text{gap}(\tilde{Y}_k)$, AGD will converge at the optimal rate $O(\frac{1}{k^2})$ if ϵ_k is of order $O(\frac{1}{k^{2+\delta}})$ with $\delta > 0$.

One advantage of the proposed dual method is that the error of the proximal operator can be easily controlled by the duality gap. Here, we continue to use the gene expression dataset in Section 6.2.3 to evaluate how the error in the proximal operator affects the performance of

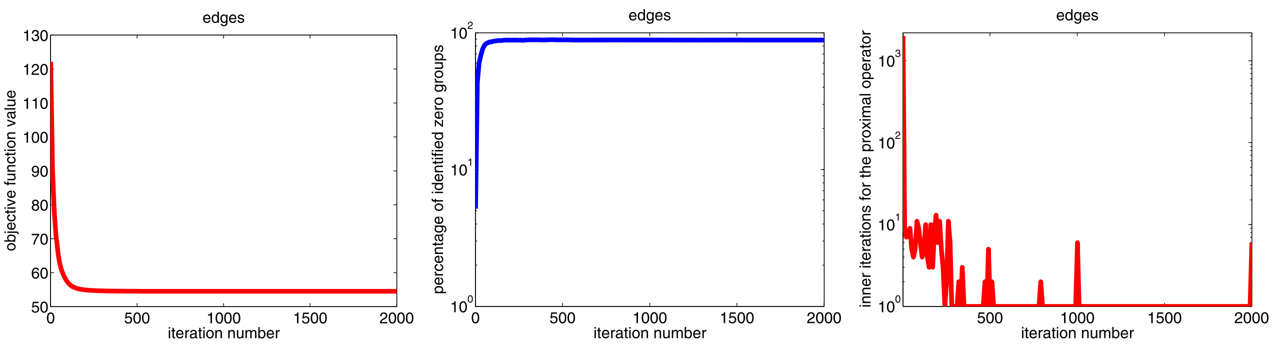

Fig. 4. Performance of the computation of the proximal operator in FoGLasso. The left plot shows the objective function value during the FoGLasso iteration. The middle plot shows the percentage of the identified zero groups by applying Lemma 3. The right plot shows the number of inner iterations for achieving the duality gap less than 10^{-10} when one solves the proximal operator via the dual reformulation (see Section 3.2).

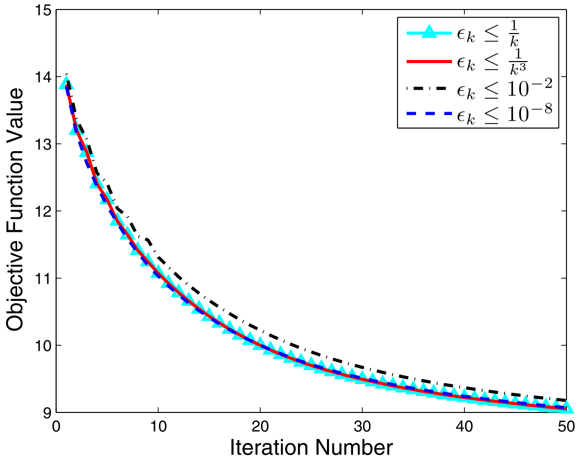

Fig. 5. Illustration of the objective function values of the first 50 iterations with different stopping criteria used for computing the proximal operator.

the algorithm in practice. We use the following ways to terminate the calculation of (4):

- $\epsilon_k \leq \frac{1}{k^n}$. Terminate the calculation until the duality gap is below $\frac{1}{k^n}$.
- $\epsilon_k \leq \epsilon$. Terminate the calculation until the duality gap is below a fixed value ϵ.

We use the objective function value against the number of iterations as the evaluation criterion, and the performance for different termination conditions is illustrated in Fig. 5.

As we can see from Fig. 5, the convergence of the objective value does not change dramatically with different termination conditions. For example, setting $\epsilon \leq \frac{1}{k}$ and $\epsilon \leq \frac{1}{k^3}$ performs equally well. This might be due to the fact that in most cases our proximal operator takes only one step to converge even with a small duality gap, e.g., 10^{-10} (as shown in the right plot of Fig. 4).

6.3 Discussions

Throughout this section, we have performed extensive experiments to illustrate the empirical performance of our proposed method. It is also interesting to analyze the relationship between the proposed method and the existing methods. In general, the methods for solving overlapping group lasso formulation can be divided into three groups:

- *AGD (FISTA) with proximal operator*, such as FoGLasso and Picard-Nesterov. These methods solve the nonsmooth optimization by first-order methods that involve the computation of a proximal operator. When an analytical solution exists for the corresponding proximal operator, an optimal convergence rate of $O(\frac{1}{k^2})$ can be achieved. For problems such as overlapping group lasso, where no closed-form solution is known for the proximal operator, the optimal convergence rate can only be guaranteed when the error of solving the proximal operator can be controlled at each iteration. For both FoGLasso and Picard-Nesterov, the complexity of the inner iteration for computing the proximal operator is $O(pg)$. This group of methods often work quite well when the proximal operator can be solved efficiently, while one disadvantage is that for a new class of problems, one needs to design a dedicated solver for computing the new proximal operator, which can be challenging for certain cases.
- *AGD with Nesterov's smoothing technique*, such as Prox-Grad. For nonsmooth problems, the smoothing technique can guarantee a convergence rate of $O(\frac{1}{k})$, with per iteration cost being $O(p^2 + pg)$ [16]. One advantage of Prox-Grad is that it can be easily applied to a wide range of structured sparse learning models, including overlapping group lasso and graph-induced lasso. However, Prox-Grad involves a smoothing parameter μ, which can affect the speed of the algorithm and needs to be tuned properly.
- *ADMM*. The worst-case convergence rate of ADMM is $O(\frac{1}{\sqrt{k}})$, and the actual speed of the implementation may rely on the choice of the penalty parameter ρ. In each iteration, ADMM solves a $p \times p$ linear system, which can be solved in $O(p^2)$ when the Cholesky decomposition of the matrix for the system can be precomputed. Therefore, the per-iteration cost of ADMM is $O(p^2 + pg)$. ADMM is known to work well in certain problems such as trace norm minimization [38].

7 Conclusion

In this paper, we consider the efficient optimization of the overlapping group Lasso penalized problem based on the AGD method. We reveal several key properties of the proximal operator associated with the overlapping group Lasso, and compute the proximal operator via solving the smooth and convex dual problem. Numerical experiments on both a synthetic and the breast cancer datasets demonstrate the efficiency of the proposed algorithm. Although with an inexact proximal operator, the optimal convergence rate of the AGD might not be guaranteed [35], [36], the algorithm performs quite well empirically. Our algorithm is extended to tackle the generalized ℓ_q norm, as well as a nonconvex formulation based on the capped norm regularization. Our preliminary results show that the capped norm leads to improved sparse pattern recovery. In the future, we plan to extend the theoretical analysis in [29], [30], [31] to the overlapping group Lasso formulation considered in this paper. In addition, we plan to apply the proposed algorithm to other real-world applications involving overlapping groups.

Acknowledgments

This work was supported by US National Science Foundation (NSF) IIS-0953662, CCF-1025177, MCB-1026710, and National Natural Science Foundation of China (NSFC) 60905035 and 61035003.

References

[1] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Royal Statistical Soc. Series B*, vol. 58, no. 1, pp. 267-288, 1996.

[2] M. Yuan and Y. Lin, "Model Selection and Estimation in Regression with Grouped Variables," *J. Royal Statistical Soc. Series B*, vol. 68, no. 1, pp. 49-67, 2006.

[3] H. Liu, M. Palatucci, and J. Zhang, "Blockwise Coordinate Descent Procedures for the Multi-Task Lasso, with Applications to Neural Semantic Basis Discovery," *Proc. 26th Ann. Int'l Conf. Machine Learning*, 2009.

[4] J. Liu, S. Ji, and J. Ye, "Multi-Task Feature Learning via Efficient $\ell_{2,1}$-Norm Minimization," *Proc. 25h Conf. Uncertainty in Artificial Intelligence,* 2009.

[5] L. Meier, S. Geer, and P. Bühlmann, "The Group Lasso for Logistic Regression," *J. Royal Statistical Soc.: Series B,* vol. 70, pp. 53-71, 2008.

[6] L. Jacob, G. Obozinski, and J. Vert, "Group Lasso with Overlap and Graph Lasso," *Proc. 26th Ann. Int'l Conf. Machine Learning,* 2009.

[7] H.D. Bondell and B.J. Reich, "Simultaneous Regression Shrinkage, Variable Selection and Clustering of Predictors with Oscar," *Biometrics,* vol. 64, pp. 115-123, 2008.

[8] S. Kim and E.P. Xing, "Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity," *Proc. Int'l Conf. Machine Learning,* 2010.

[9] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network Flow Algorithms for Structured Sparsity," *Proc. Advances in Neural Information Processing Systems,* pp. 1558-1566, 2010.

[10] P. Zhao, G. Rocha, and B. Yu, "The Composite Absolute Penalties Family for Grouped and Hierarchical Variable Selection," *Annals of Statistics,* vol. 37, no. 6A, pp. 3468-3497, 2009.

[11] S. Mosci, S. Villa, A. Verri, and L. Rosasco, "A Primal-Dual Algorithm for Group Sparse Regularization with Overlapping Groups," *Proc. Advances in Neural Information Processing Systems,* 2010.

[12] Z. Qin and D. Goldfarb, "Structured Sparsity via Alternating Direction Methods," *Arxiv preprint arXiv:1105.0728,* 2011.

[13] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured Variable Selection with Sparsity-Inducing Norms," *arXiv:0904.3523,* 2009.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.* Now Publishers. Inc., 2010.

[15] A. Argyriou, C. Micchelli, M. Pontil, L. Shen, and Y. Xu, "Efficient First Order Methods for Linear Composite Regularizers," *Arxiv preprint arXiv:1104.1436,* 2011.

[16] X. Chen, Q. Lin, S. Kim, J. Carbonell, and E. Xing, "Smoothing Proximal Gradient Method for General Structured Sparse Learning," *Annals of Applied Statistics,* vol. 6, no. 2, pp. 719-752, 2012.

[17] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sciences,* vol. 2, no. 1, pp. 183-202, 2009.

[18] A. Nemirovski, *Efficient Methods in Convex Programming,* 1994.

[19] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer Academic Publishers, 2004.

[20] J.-J. Moreau, "Proximité et Dualité dans un Espace Hilbertien," *Bull. Soc. Math. France,* vol. 93, pp. 273-299, 1965.

[21] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal Methods for Sparse Hierarchical Dictionary Learning," *Proc. Int'l Conf. Machine Learning,* 2010.

[22] J. Liu and J. Ye, *SLEP: Sparse Learning with Efficient Projections,* Arizona State Univ., http://www.public.asu.edu/jye02/Software/SLEP, 2009.

[23] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise Coordinate Optimization," *Annals of Applied Statistics,* vol. 1, no. 2, pp. 302-332, 2007.

[24] J.F. Bonnans and A. Shapiro, "Optimization Problems with Perturbations: A Guided Tour," *SIAM Rev.,* vol. 40, no. 2, pp. 228-264, 1998.

[25] J.M. Danskin, *The Theory of Max-Min and Its Applications to Weapons Allocation Problems.* Springer-Verlag, 1967.

[26] Y. Ying, C. Campbell, and M. Girolami, "Analysis of SVM with Indefinite Kernels," *Proc. Advances in Neural Information Processing Systems,* pp. 2205-2213, 2009.

[27] P. Combettes and J. Pesquet, "Proximal Splitting Methods in Signal Processing," *Arxiv preprint arXiv:0912.3522,* 2009.

[28] J. Liu and J. Ye, "Efficient l1/lq Norm Regularization," *Arxiv preprint arXiv:1009.4766,* 2010.

[29] T. Zhang, "Multi-Stage Convex Relaxation for Feature Selection," *Arxiv preprint arXiv:1106.0565,* 2011.

[30] X. Shen, W. Pan, and Y. Zhu, "Likelihood-Based Selection and Sharp Parameter Estimation," *J. Am. Statistical Assoc.,* vol. 107, no. 497, pp. 223-232, 2012.

[31] T. Zhang, "Analysis of Multi-Stage Convex Relaxation for Sparse Regularization," *J. Machine Learning Research,* vol. 11, pp. 1081-1107, 2010.

[32] M.J. Van de Vijver et al., "A Gene-Expression Signature as a Predictor of Survival in Breast Cancer," *The New England J. Medicine,* vol. 347, no. 25, pp. 1999-2009, 2002.

[33] A. Subramanian et al., "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles," *Proc. Nat'l Academy of Sciences USA,* vol. 102, no. 43, pp. 15545-15550, 2005.

[34] H.Y. Chuang, E. Lee, Y.T. Liu, D. Lee, and T. Ideker, "Network-Based Classification of Breast Cancer Metastasis," *Molecular Systems Biology,* vol. 3, no. 140, 2007.

[35] R. Rockafellar, "Monotone Operators and the Proximal Point Algorithm," *SIAM J. Control and Optimization,* vol. 14, pp. 877-898, 1976.

[36] B. He and X. Yuan, "An Accelerated Inexact Proximal Point Algorithm for Convex Minimization," *J. Optimization Theory and Applications,* vol. 154, p. 536, 2010.

[37] M. Schmidt, N.L. Roux, and F. Bach, "Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization," *Proc. Advances in Neural Information Processing Systems,* 2011.

[38] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor Completion for Estimating Missing Values in Visual Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 35, no. 1, pp. 208-220, Jan. 2013.

Lei Yuan received the bachelor's degree from Nanjing University of Posts and Telecommunications in 2008, and is currently working toward the PhD degree in the Department of Computer Sciences and Engineering, Arizona State University. His research interests include machine learning, data mining, structured sparse learning, and their applications to biomedical informatics.

Jun Liu received the BS degree from Nantong Institute of Technology (now Nantong University) in 2002, and the PhD degree from Nanjing University of Aeronautics and Astronautics (NUAA) in November 2007. During February 2008-February 2011, he was a postdoctoral researcher in the Biodesign Institute, Arizona State University. He is currently a research scientist at Siemens Corporate Research. His research interests include dimensionality reduction, sparse learning, and large-scale optimization. He has authored or coauthored more than 30 scientific papers.

Jieping Ye received the PhD degree in computer science from the University of Minnesota, Twin Cities, in 2005. He is an associate professor of computer science and engineering at Arizona State University (ASU). His research interests include machine learning, data mining, and biomedical informatics. He received the Outstanding Student Paper Award at ICML in 2004, the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, the US National Science Foundation (NSF) CAREER Award in 2010, the KDD Best Research Paper Award honorable mention in 2010, and the KDD Best Research Paper Nomination in 2011 and 2012. He is a senior member of the IEEE.