Efficient Sparse Group Feature Selection via Nonconvex Optimization

Shuo Xiang[†‡] SHUO.XIANG@ASU.EDU
Xiaotong Shen[*] SHENX002@UMN.EDU
Jieping Ye[†‡] JIEPING.YE@ASU.EDU

[†]Computer Science and Engineering, Arizona State University, Tempe, AZ 85287
[‡]Center for Evolutionary Medicine and Informatics, Arizona State University, Tempe, AZ 85287
[*]School of Statistics, University of Minnesota, Minneapolis, MN 55347

Abstract

Sparse feature selection has been demonstrated to be effective in handling high-dimensional data. While promising, most of the existing works use convex methods, which may be suboptimal in terms of the accuracy of feature selection and parameter estimation. In this paper, we expand a nonconvex paradigm to sparse group feature selection, which is motivated by applications that require identifying the underlying group structure and performing feature selection simultaneously. The main contributions of this article are twofold: (1) computationally, we introduce a nonconvex sparse group feature selection model and present an efficient optimization algorithm, of which the key step is a projection with two coupled constraints; (2) statistically, we show that the proposed model can reconstruct the oracle estimator. Therefore, consistent feature selection and parameter estimation can be achieved. Numerical results on synthetic and real-world data suggest that the proposed nonconvex method compares favorably against its competitors, thus achieving desired goal of delivering high performance.

1. Introduction

During the past decade, sparse feature selection has been extensively investigated, on both optimization algorithms (Bach et al., 2010) and statistical properties (Tibshirani, 1996; Zhao & Yu, 2006; Bickel et al., 2009). When the data possesses certain group structure, sparse modeling has been explored in (Yuan & Lin, 2006; Meier et al., 2008; Huang & Zhang, 2010) for group feature selection. The group lasso (Yuan & Lin, 2006) proposes an L_2-regularization method for each group, which ultimately yields a group-wisely sparse model. The utility of such a method has been demonstrated in detecting splice sites (Yang et al., 2010)—an important step in gene finding and theoretically justified in (Huang & Zhang, 2010). The sparse group lasso (Friedman et al., 2010) enables to encourage sparsity at the level of both features and groups simultaneously. In the literature, most approaches use convex methods to pursue the grouping effect due to globality of the solution and tractable computation. However, this may lead to suboptimal results. Recent studies demonstrate that nonconvex methods (Fan & Li, 2001; Wang et al., 2007; Breheny & Huang, 2009; Huang et al., 2009; 2012), particularly the truncated L_1-penalty (Shen et al., 2012; Mazumder et al., 2011; Zhang, 2011), may deliver superior performance than the standard L_1-formulation. In addition, (Shen et al., 2012) suggests that a constrained nonconvex formulation is slightly more preferable than its regularization counterpart due to theoretical merits. In this paper, we investigate the sparse group feature selection through a constrained nonconvex formulation. Ideally, we wish to optimize the following L_0-model:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} I(|x_j|\neq 0) \leq s_1 \\ & \sum_{j=1}^{|G|} I(\|\boldsymbol{x}_{G_j}\|_2 \neq 0) \leq s_2, \end{aligned} \tag{1}$$

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

where $\boldsymbol{A}$ is an n by p data matrix with its columns representing different features. $\boldsymbol{x} = (x_1, \cdots, x_p)$ is partitioned into $|G|$ non-overlapping groups $\{\boldsymbol{x}_{G_i}\}$ and $I(\cdot)$ is the indicator function. The advantage of the L_0-model (1) lies in its complete control on two levels of sparsity (s_1, s_2), which are the numbers of features and groups respectively. However, problems like (1) are known to be NP-hard (Natarajan, 1995) due to the discrete nature.

This paper develops an efficient nonconvex method, which is a computational surrogate of the L_0-method described above and has theoretically guaranteed performance. We contribute in two aspects: (i) computationally, we present an efficient optimization algorithm, of which the key step is a projection with two coupled constraints. (ii) statistically, the proposed method retains the merits of the L_0 approach (1) in the sense that the oracle estimator can be reconstructed, which leads to consistent feature selection and parameter estimation.

The rest of this paper is organized as follows. Section 2 presents our nonconvex formulation with its optimization algorithm explored in Section 3. We analyze the theoretical properties of our formulation in Section 4 and discuss the significance of this work in Section 5. Section 6 demonstrates the efficiency of the proposed method as well as the performance on real-world applications. Section 7 concludes the paper with a discussion of future research.

2. Nonconvex Formulation and Computation

One major difficulty of solving (1) comes from nonconvex and discrete constraints, which require enumerating all possible combinations of features and groups to achieve the optimal solution. Therefore we approximate these constraints by their continuous computational surrogates:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} J_\tau(|x_j|) \le s_1, \quad \sum_{i=1}^{|G|} J_\tau(\|\boldsymbol{x}_{G_i}\|_2) \le s_2, \end{aligned} \tag{2}$$

where $J_\tau(z) = \min(|z|/\tau, 1)$ is a truncated L_1-function approximating the L_0-function (Shen et al., 2012; Zhang, 2010), and $\tau > 0$ is a tuning parameter such that $J_\tau(z)$ approximates the indicator function $I(|z| \neq 0)$ as τ approaches zero.

To solve the nonconvex problem (2), we develop a Difference of Convex (DC) algorithm (Tao & An, 1997) based on a decomposition of each nonconvex constraint function into a difference of two convex functions:

$$\sum_{j=1}^{p} J_\tau(|x_j|) = S_1(\boldsymbol{x}) - S_2(\boldsymbol{x}),$$

where

$$S_1(\boldsymbol{x}) = \frac{1}{\tau}\sum_{j=1}^{p}|x_j|, \quad S_2(\boldsymbol{x}) = \frac{1}{\tau}\sum_{j=1}^{p}\max\{|x_j|-\tau, 0\}$$

are convex in $\boldsymbol{x}$. Then each trailing convex function, say $S_2(\boldsymbol{x})$, is replaced by its affine minorant at the previous iteration

$$S_1(\boldsymbol{x}) - S_2(\hat{\boldsymbol{x}}^{(m-1)}) - \nabla S_2(\hat{\boldsymbol{x}}^{(m-1)})^T(\boldsymbol{x}-\hat{\boldsymbol{x}}^{(m-1)}), \tag{3}$$

which yields an upper approximation of the constraint function $\sum_{j=1}^{p} J_\tau(|x_j|)$ as follows:

$$\frac{1}{\tau}\sum_{j=1}^{p}|x_j|\cdot I(|\hat{x}_j^{(m-1)}| \le \tau) + \sum_{j=1}^{p} I(|\hat{x}_j^{(m-1)}| > \tau) \le s_1. \tag{4}$$

Similarly, the second nonconvex constraint in (2) can be approximated by

$$\frac{1}{\tau}\sum_{j=1}^{|G|}\|\boldsymbol{x}_{G_j}\|_2\cdot I(\|\hat{\boldsymbol{x}}_{G_j}^{(m-1)}\|_2 \le \tau) + \sum_{j=1}^{|G|} I(\|\hat{\boldsymbol{x}}_{G_j}^{(m-1)}\|_2 > \tau) \le s_2. \tag{5}$$

Note that both (4) and (5) are convex constraints, which result in a convex subproblem as follows:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \frac{1}{\tau}\|\boldsymbol{x}^{T_1(\hat{\boldsymbol{x}}^{(m-1)})}\|_1 \le s_1 - (p - |T_1(\hat{\boldsymbol{x}}^{(m-1)})|) \\ & \frac{1}{\tau}\|\boldsymbol{x}^{T_3(\hat{\boldsymbol{x}}^{(m-1)})}\|_G \le s_2 - (|G| - |T_2(\hat{\boldsymbol{x}}^{(m-1)})|), \end{aligned} \tag{6}$$

where T_1, T_2 and T_3 are the support sets[1] defined as:

$$\begin{aligned} & T_1(\boldsymbol{x}) = \{i : |x_i| \le \tau\}, \quad T_2(\boldsymbol{x}) = \{i : \|\boldsymbol{x}_{G_i}\|_2 \le \tau\} \\ & T_3(\boldsymbol{x}) = \{i : x_i \in \boldsymbol{x}_{G_j}, j \in T_2(\boldsymbol{x})\}, \end{aligned}$$

$\|\boldsymbol{x}^{T_1}\|_1$ and $\|\boldsymbol{x}^{T_3}\|_G$ denote the corresponding value restricted on T_1 and T_3 respectively, and $\|\boldsymbol{x}\|_G = \sum_{i=1}^{|G|}\|\boldsymbol{x}_{G_i}\|_2$. Solving (6) would provide us an updated solution, denoted as $\hat{\boldsymbol{x}}^{(m)}$, which leads to a refined formulation of (6). Such procedure is iterated until the objective value stops decreasing. The DC algorithm is summarized in Algorithm 1, from which we can see that efficient computation of (6) is critical to the overall DC routine. We defer detailed discussion of this part to Section 3.

[1] Support sets indicate that the elements outside these sets have no effect on the particular items in the constraints of (6).

Algorithm 1 DC programming for solving (2)

Input: $\boldsymbol{A}$, $\boldsymbol{y}$, s_1, s_2
Output: solution $\boldsymbol{x}$ to (2)
1: Initialize $\hat{\boldsymbol{x}}^{(0)}$.
2: **for** $m = 1, 2, \cdots$ **do**
3: Compute $\hat{\boldsymbol{x}}^{(m)}$ by optimizing (6).
4: Update T_1, T_2 and T_3.
5: **if** the objective stops decreasing **then**
6: **return** $\boldsymbol{x} = \hat{\boldsymbol{x}}^{(m)}$
7: **end if**
8: **end for**

3. Optimization Procedures

As mentioned in Section 2, efficient computation of the convex subproblem (6) is of critical importance for the proposed DC algorithm. Note that (6) has an identical form of the constrained sparse group lasso problem:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \\ \text{subject to} & \|\boldsymbol{x}\|_1 \leq s_1 \\ & \|\boldsymbol{x}\|_G \leq s_2 \end{array} \tag{7}$$

except that $\boldsymbol{x}$ is restricted to the two support sets. As to be shown in Section 3.3, an algorithm for solving (6) can be obtained through only a few modifications on that of (7). Therefore, we first focus on solving (7). Notice that if problem (7) has only one constraint, the solution is well-established (Duchi et al., 2008; Bach et al., 2010). However, the two coupled constraints here make the optimization problem more challenging to solve.

3.1. Accelerated Gradient Method

For large-scale problems, the dimensionality of data can be very high, therefore first-order optimization is often preferred. We adapt the well-known accelerated gradient method (AGM) (Nesterov, 2007; Beck & Teboulle, 2009), which is commonly used due to its fast convergence rate.

To apply AGM to our formulation (7), the crucial step is to solve the following Sparse Group Lasso Projection (SGLP):

$$\begin{array}{lll} \underset{\boldsymbol{x}}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2 & \\ \text{subject to} & \|\boldsymbol{x}\|_1 \leq s_1 & (C_1) \\ & \|\boldsymbol{x}\|_G \leq s_2 & (C_2), \end{array} \tag{8}$$

which is an Euclidean projection onto a convex set and a special case of (7) when A is the identity. For convenience, let C_1 and C_2 denote the above two constraints in what follows.

Since the AGM is a standard framework whose efficiency mainly depends on that of the projection step, we leave the detailed description of AGM in the supplement and introduce the efficient algorithm for this projection step (8).

3.2. Efficient Projection

We begin with some special cases of (8). If only C_1 exists, (8) becomes the well-known L_1-ball projection (Duchi et al., 2008), whose optimal solution is denoted as $\mathcal{P}_1^{s_1}(\boldsymbol{v})$, standing for the projection of $\boldsymbol{v}$ onto the L_1-ball with radius s_1. On the other hand, if only C_2 is involved, it becomes the group lasso projection, denoted as $\mathcal{P}_G^{s_2}$. Moreover, we say a constraint is *active*, if and only if an equality holds at the optimal solution x^*; otherwise, it is *inactive*.

Preliminary results are summarized in Lemma 1:

Lemma 1. *Denote a global minimizer of (8) as $\boldsymbol{x}^*$. Then the following results hold:*

1. *If both C_1 and C_2 are inactive, then $\boldsymbol{x}^* = \boldsymbol{v}$.*
2. *If C_1 is the only active constraint, i.e., $\|\boldsymbol{x}^*\|_1 = s_1$, $\|\boldsymbol{x}^*\|_G < s_2$, then $\boldsymbol{x}^* = \mathcal{P}_1^{s_1}(\boldsymbol{v})$*
3. *If C_2 is the only active constraint, i.e., $\|\boldsymbol{x}^*\|_1 < s_1$, $\|\boldsymbol{x}^*\|_G = s_2$, then $\boldsymbol{x}^* = \mathcal{P}_G^{s_2}(\boldsymbol{v})$*

3.2.1. Computing $\boldsymbol{x}^*$ from the optimal dual variables

Lemma 1 describes a global minimizer when either constraint is inactive. Next we consider the case in which both C_1 and C_2 are active. By the convex duality theory (Boyd & Vandenberghe, 2004), there exist unique non-negative dual variables λ^* and η^* such that x^* is also the global minimizer of the following regularized problem:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2 + \lambda^*\|\boldsymbol{x}\|_1 + \eta^*\|\boldsymbol{x}\|_G, \tag{9}$$

whose solution is given by the following Theorem.

Theorem 1 ((Friedman et al., 2010)). *The optimal solution $\boldsymbol{x}^*$ of (9) is given by*

$$\boldsymbol{x}^*_{G_i} = \max\{\|\boldsymbol{v}^{\lambda^*}_{G_i}\|_2 - \eta^*, 0\}\frac{\boldsymbol{v}^{\lambda^*}_{G_i}}{\|\boldsymbol{v}^{\lambda^*}_{G_i}\|_2} \quad i = 1, 2, \cdots, |G| \tag{10}$$

where $\boldsymbol{v}^{\lambda^*}_{G_i}$ is computed via soft-thresholding (Donoho, 2002) $\boldsymbol{v}_{G_i}$ with threshold λ^* as follows:

$$\boldsymbol{v}^{\lambda^*}_{G_i} = SGN(\boldsymbol{v}_{G_i}) \cdot \max\{|\boldsymbol{v}_{G_i}| - \lambda^*, 0\},$$

where $SGN(\cdot)$ is the sign function and all the operations are taken element-wisely.

Theorem 1 gives an analytical solution of $\boldsymbol{x}^*$ in an ideal situation when the values of λ^* and η^* are given. Unfortunately, this is not the case and the values of λ^* and η^* need to be computed directly from (8). Based on Theorem 1, we have the following conclusion characterizing the relations between the dual variables:

Corollary 1. *The following equations hold:*

$$\|\boldsymbol{x}^*\|_1 = \sum_{i=1}^{|G|} \max\{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_2 - \eta^*, 0\} \frac{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_1}{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_2} = s_1 \quad (11)$$

$$\|\boldsymbol{x}^*\|_G = \sum_{i=1}^{|G|} \max\{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_2 - \eta^*, 0\} = s_2 \quad (12)$$

Suppose λ^* is given, then computing η^* from (12) amounts to solving a median finding problem, which can be done in linear time (Duchi et al., 2008).

Finally, we treat the case of unknown λ^* (thus unknown η^*). We propose an efficient bisection approach to compute it.

3.2.2. Computing λ^*: bisection

Given an initial guess (estimator) of λ^*, says $\hat{\lambda}$, one may perform bisection to locate the optimal λ^*, provided that there exists an oracle procedure indicating if the optimal value is greater than $\hat{\lambda}$[2]. This bisection method can estimate λ^* in logarithm time. Next, we shall design an oracle procedure.

Let the triples

$$(\boldsymbol{x}^*, \lambda^*, \eta^*) = \mathrm{SGLP}(\boldsymbol{v}, s_1, s_2)$$

be the optimal solution of (8) with both constraints active, i.e., $\|\boldsymbol{x}^*\|_1 = s_1$, $\|\boldsymbol{x}^*\|_G = s_2$, with (λ^*, η^*) be the optimal dual variables. Consider the following two sparse group lasso projections:

$$(\boldsymbol{x}, \lambda, \eta) = \mathrm{SGLP}(\boldsymbol{v}, s_1, s_2),$$
$$(\boldsymbol{x}', \lambda', \eta') = \mathrm{SGLP}(\boldsymbol{v}, s_1', s_2').$$

The following key result holds.

Theorem 2. *If $\lambda \le \lambda'$ and $s_2 = s_2'$, then $s_1 \ge s_1'$.*

Theorem 2 gives the oracle procedure with its proof presented in the supplement. For a given estimator $\hat{\lambda}$, we compute its corresponding $\hat{\eta}$ from (12) and then $\hat{s_1}$ from (11), satisfying $(\hat{\boldsymbol{x}}, \hat{\lambda}, \hat{\eta}) = \mathrm{SGLP}(\boldsymbol{v}, \hat{s_1}, s_2)$. Then $\hat{s_1}$ is compared with s_1. Clearly, by Theorem 2, if $\hat{s_1} \le s_1$, the estimator $\hat{\lambda}$ is no less than λ^*. Otherwise, $\hat{s_1} > s_1$ means $\hat{\lambda} < \lambda^*$. In addition, from (11) we know that $\hat{s_1}$ is a continuous function of $\hat{\lambda}$. Together with the monotonicity given in Theorem 2, a bisection approach can be employed to calculate λ^*. Algorithm 2 gives a detailed description of this procedure.

[2] An upper bound and a lower bound of λ^* should be provided in order to perform the bisection. These bounds can be easily derived from the assumption that both C_1 and C_2 are active.

Algorithm 2 Sparse Group Lasso Projection Algorithm

Input: $\boldsymbol{v}$, s_1, s_2
Output: an optimal solution $\boldsymbol{x}$ to the Sparse Group Projection Problem

Function SGLP($\boldsymbol{v}$, s_1, s_2)
if $\|\boldsymbol{x}\|_1 \le s_1$ **and** $\|\boldsymbol{x}\|_G \le s_2$ **then**
  return $\boldsymbol{v}$
end if
$\boldsymbol{x}_{C_1} = \mathcal{P}_1^{s_1}(\boldsymbol{v})$
$\boldsymbol{x}_{C_2} = \mathcal{P}_G^{s_2}(\boldsymbol{v})$
$\boldsymbol{x}_{C_{12}} =$ bisec($\boldsymbol{v}$, s_1, s_2)
if $\|\boldsymbol{x}_{C_1}\|_G \le s_2$ **then**
  return $\boldsymbol{x}_{C_1}$
else if $\|\boldsymbol{x}_{C_2}\|_1 \le s_1$ **then**
  return $\boldsymbol{x}_{C_2}$
else
  return $\boldsymbol{x}_{C_{12}}$
end if

Function bisec($\boldsymbol{v}$, s_1, s_2)
Initialize *up*, *low* and *tol*
while $up - low > tol$ **do**
  $\hat{\lambda} = (low + up)/2$
  if (12) has a solution $\hat{\eta}$ given $v^{\hat{\lambda}}$ **then**
    calculate $\hat{s_1}$ using $\hat{\eta}$ and $\hat{\lambda}$.
    if $\hat{s_1} \le s_1$ **then**
      $up = \hat{\lambda}$
    else
      $low = \hat{\lambda}$
    end if
  else
    $up = \hat{\lambda}$
  end if
end while
$\lambda^* = up$
Solve (12) to get η^*
Calculate $\boldsymbol{x}^*$ from λ^* and η^* `via` (10)
return $\boldsymbol{x}^*$

3.3. Solving Restricted version of (7)

Finally, we modify the above procedures to compute the optimal solution of the restricted problem (6). To apply the accelerated gradient method, we consider

the following projection step:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 & \\ \text{subject to} & \|\boldsymbol{x}^{T_1}\|_1 \le s_1 & (C_1) \\ & \|\boldsymbol{x}^{T_3}\|_G \le s_2 & (C_2). \end{array} \tag{13}$$

Our first observation is: $T_3(\boldsymbol{x}) \subset T_1(\boldsymbol{x})$, since if an element of $\boldsymbol{x}$ lies in a group whose L_2-norm is less than τ, then the absolute value of this element must also be less than τ. Secondly, from the decomposable nature of the objective function, we conclude that:

$$x_j^* = \begin{cases} v_j & \text{if } j \in (T_1)^c \\ v_j^{\lambda^*} & \text{if } j \in T_1 \backslash T_3, \end{cases}$$

since there are no constraints on x_j if it is outside T_1 and involves only the L_1-norm constraint if $j \in T_1 \backslash T_3$. Following routine calculations as in (Duchi et al., 2008), we obtain the following results similar to (11) and (12):

$$s_1 = \sum_{i \in T_2} \max\{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_2 - \eta^*, 0\} \frac{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_1}{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_2} + \sum_{j \in T_1 \backslash T_3} v_j^{\lambda^*} \tag{14}$$

$$s_2 = \sum_{i \in T_2} \max\{\|\boldsymbol{v}_{G_i}^{\lambda^*}\|_2 - \eta^*, 0\}. \tag{15}$$

Based on (14) and (15), we design a similar bisection approach to compute λ^* and thus $(\boldsymbol{x}^*)^{T_3}$, as in Algorithm 2. Details can be found in the supplement.

Since the projection (13) does not possess an closed-form, it is instructive to discuss the convergence property of overall accelerated gradient method. Follow the discussion in (Schmidt et al., 2011), we can provide sufficient conditions for a guaranteed convergence rate. Moreover, we found in practice that a reasonable convergence property can be obtained as long as the precision level for the computation of the projection is small, as revealed in Section 6.

Remark Problem (7) can also be solved using the Alternating Direction Method of Multiplier (ADMM) (Boyd et al., 2011) instead of the accelerated gradient method (AGM). However, our evaluations show that AGM with our projection algorithm is more efficient than ADMM.

4. Theoretical Results

This section investigates theoretical aspects of the proposed method. More specifically, we demonstrate that the oracle estimator $\hat{\boldsymbol{x}}^o$, the least squares estimator based on the true model, can be reconstructed. As a result, consistent selection as well as optimal parameter estimation can be achieved.

For better presentation, we introduce some notations that would be only utilized in this section. Let $C = (G_{i_1}, \cdots, G_{i_k})$ be the collection of groups that contain nonzero elements. Let $A_{G_j} = A_{G_j}(\boldsymbol{x})$ and $A = A(\boldsymbol{x})$ denote the indices of nonzero elements of $\boldsymbol{x}$ in group G_j and in entire $\boldsymbol{x}$ respectively. Define

$$\mathcal{S}_{j,i} = \{\boldsymbol{x} \in \mathcal{S} : (A_C, C) \neq (A_{Cf^0}, C^0), |A| = j, |C| = i\},$$

where $\mathcal{S}$ is the feasible region of (2) and C^0 represents the true nonzero groups.

The following assumptions are used to obtain consistent reconstruction of the oracle estimator:

Assumption 1 (Separation condition)**.** *Define*

$$C_{\min}(\boldsymbol{x}^0) = \inf_{\boldsymbol{x} \in \mathcal{S}} \frac{-\log(1 - h^2(\boldsymbol{x}, \boldsymbol{x}^0))}{\max(|C^0 \setminus C|, 1)},$$

then for some constant $c_1 > 0$*,*

$$C_{\min}(\boldsymbol{x}^0) \ge c_1 \frac{\log|G| + \log s_1^0}{n},$$

where

$$h(\boldsymbol{x}, \boldsymbol{x}^0) = \Big(\frac{1}{2}\int (g^{1/2}(\boldsymbol{x}, y) - g^{1/2}(\boldsymbol{x}^0, y))^2 d\mu(y)\Big)^{1/2}$$

is the Hellinger-distance for densities with respect to a dominating measure μ*.*

Assumption 2 (Complexity of the parameter space)**.** *For some constants* $c_0 > 0$ *and any* $0 < t < \varepsilon \le 1$*,*

$$H(t, \mathcal{F}_{j,i}) \le c_0 \max((\log(|G| + s_1^0))^2, 1) |\mathcal{B}_{j,i}| \log(2\varepsilon/tff),$$

where $\mathcal{B}_{j,i} = \mathcal{S}_{j,i} \cap \{\boldsymbol{x} \in h(\boldsymbol{x}, \boldsymbol{x}^0) \le 2\varepsilon\}$ *is a local parameter space and* $\mathcal{F}_{j,i} = \{g^{1/2}(\boldsymbol{x}, y) : \boldsymbol{x} \in \mathcal{B}_{j,i}\}$ *is a collection of square-root densities.* $H(\cdot, \mathcal{F})$ *is the bracketing Hellinger metric entropy of space* $\mathcal{F}$ *(Kolmogorov & Tihomirov, 1961).*

Assumption 3. *For some positive constants* d_1, d_2, d_3 *with* $d_1 > 10$*,*

$$-\log(1 - h^2(\boldsymbol{x}, \boldsymbol{x}^0)) \ge -d_1 \log(1 - h^2(\boldsymbol{x}^\tau, \boldsymbol{x}^0)) - d_3 \tau^{d_2} p,$$

where $\boldsymbol{x}^\tau = (x_1 I(|x_1| \ge \tau), \cdots, x_p I(|x_p| \ge \tau))$*.*

With these assumptions hold, we can conclude the following non-asymptotic probability error bound regarding the reconstruction of the oracle estimator $\hat{\boldsymbol{x}}^o$. The proof is provided in the supplement.

Theorem 3. *Suppose that Assumptions 2 and 3 hold. For a global minimizer of* (2) $\hat{\boldsymbol{x}}$ *with* $(s_1, s_2) = (s_1^0, s_2^0)$

and $\tau \leq \left(\frac{(d_1-10)C_{\min}(\boldsymbol{x}^0)}{d_3 d}\right)^{1/d_2}$, *the following result hold:*

$$\mathbb{P}\Big(\hat{\boldsymbol{x}} \neq \hat{\boldsymbol{x}}^o\Big) \leq \exp\Big(-c_2 n C_{\min}(\boldsymbol{x}^0) + 2(\log|G| + \log s_1^0)\Big).$$

Moreover, with Assumption 1 hold, $\mathbb{P}\Big(\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}^o\Big) \to 1$ *and*

$$Eh^2(\hat{\boldsymbol{x}}, \boldsymbol{x}^o) = (1 + o(1)) \max(Eh^2(\hat{\boldsymbol{x}}^o, \boldsymbol{x}^0), \frac{s_1^0}{n})$$

as $n \to \infty$, $|G| \to \infty$.

Theorem 3 states that the oracle estimator $\hat{\boldsymbol{x}}^o$ can be accurately reconstructed, which in turn yields feature selection consistency as well as the recovery of the performance of the oracle estimator in parameter estimation. Moreover, as indicated in Assumption 1, the result holds when $s_1^0|G|$ grows in the order of $\exp(c_1^{-1} n C_{\min})$. This is in contrast to existing results on consistent feature selection, where the number of candidate features should be no greater than $\exp(c^* n)$ for some c^* (Zhao & Yu, 2006; Wang et al., 2007). In this sense, the number of candidate features is allowed to be much larger when an additional group structure is incorporated, particularly when each group contains considerable redundant features. It is not clear whether such a result also holds for other bi-level[3] variable selection methods, such as the composite MCP (Huang et al., 2009) and group bridge (Breheny & Huang, 2009).

To our knowledge, our theory for the grouped selection is the first of this kind. However, it has a root in feature selection. The large deviation approach used here is applicable to derive bounds for feature selection consistency. In such a situation, the result agrees with the necessary condition for feature selection consistency for any method, except for the constants independent of the sample size (Shen et al., 2012). In other words, the required conditions are weaker than those for L_1-regularization commonly used in the literature (Van De Geer & Bühlmann, 2009). The use of the Hellinger-distance is to avoid specifying a sub-Gaussian tail of the random error. This means that the result continues to hold even when the error does not have a sub-Gaussian tail. Although we require $\hat{\boldsymbol{x}}$ to be a global minimizer of (2), a weaker version of the theory can be derived for a local minimizer obtained from the DC programming by following similar derivations in (Shen et al., 2012). We leave such discussions in a longer version of the paper.

[3]The by-level here means simultaneous group-level and feature-level analysis. This term is first introduced in (Breheny & Huang, 2009).

5. Significance

This section is devoted to a brief discussion of advantages of our work statistically and computationally. Moreover, it explains why the proposed method is useful to perform efficient and interpretable feature selection given a natural group structure.

Interpretability. The parameters in formulation (2) are highly interpretable in that s_1 and s_2 are upper bounds of the number of nonzero elements as well as that of groups. This is advantageous, especially in the presence of certain prior knowledge regarding the number of features and/or that of groups. However, such an interpretation vanishes with other (convex & nonconvex) methods such as lasso, sparse group lasso, composite MCP or group bridge, in which incorporating such prior knowledge often requires repeated trials of different parameters.

Parameter tuning. Typically, tuning parameters for good generalization usually requires considerable amount work due to a large number of choices of parameters. However, parameter tuning in model (1) may search through integer values in a bounded range, and can be further simplified when certain prior knowledge is available. This permits more efficient tuning than its regularization counterpart. Based on our limited experience, we note that τ does not need to be tuned precisely as we may fix at some small values.

Performance and Computation. Although our model (2) is proposed as a computational surrogate of the ideal L_0-method, its performance can also be theoretically guaranteed, i.e., consistent feature selection can be achieved. Moreover, the computation of our model is much more efficient and applicable to large-scale applications.

6. Empirical Evaluation

6.1. Evaluation of Projection Algorithms

Since DC programming and the accelerated gradient methods are both standard, the efficiency of the proposed nonconvex formulation (2) depends on the projection step in (8). Therefore, we focus on evaluating the projection algorithms and comparing with two popular projection algorithms: Alternating Direction Method of Multiplier (ADMM) (Boyd et al., 2011) and Dykstra's projection algorithm (Combettes & Pesquet, 2010). We give a detailed derivation of adapting these two algorithms to our formulation in the supplement.

To evaluate the efficiency, we first generate the vector $\boldsymbol{v}$ whose entries are uniformly distributed in $[-50, 50]$ and the dimension of $\boldsymbol{v}$, denoted as p, is chosen from

the set $\{10^2, 10^3, 10^4, 10^5, 10^6\}$. Next we partition the vector into 10 groups of equal size. Finally, s_2 is set to $5\log(p)$ and s_1, the radius of the L_1-ball, is computed by $\frac{\sqrt{10}}{2}s_2$ (motivated by the fact that $s_1 \leq \sqrt{10}s_2$).

For a fair comparison, we run our projection algorithm until converge and record the minimal objective value as f^*. Then we run ADMM and Dykstra's algorithm until their objective values become close to ours. More specifically, we terminate their iterations as soon as $f_{\text{ADMM}} - f^* \leq 10^{-3}$ and $f_{\text{Dykstra}} - f^* \leq 10^{-3}$, where f_{ADMM} and f_{Dykstra} stand for the objective value of ADMM and Dykstra's algorithm respectively. Table 1 summarizes the average running time of all three algorithms over 100 replications.

Table 1. Running time (in seconds) of Dykstra's, ADMM and our projection algorithm. All three algorithms are averaged over 100 replications.

Methods	10^2	10^3	10^4	10^5	10^6
Dykstra	0.1944	0.5894	4.8702	51.756	642.60
ADMM	0.0519	0.1098	1.2000	26.240	633.00
ours	$< 10^{-7}$	0.0002	0.0051	0.0440	0.5827

Next we demonstrate the accuracy of our projection algorithm. Toward this end, the general convex optimization toolbox CVX (Grant & Boyd, 2011) is chosen as the baseline. Following the same strategy of generating data, we report the distance (computed from the Euclidean norm $\|\cdot\|_2$) between optimal solution of the three projection algorithms and that of the CVX as well as the running time. Note that the projection is strictly convex with a unique global optimal solution.

For ADMM and Dykstra's algorithm, the termination criterion is that the relative difference of the objective values between consecutive iterations is less than a threshold value. Specifically, we terminate the iteration if $|f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k)| \leq 10^{-7} f(\boldsymbol{x}_{k-1})$. For our projection algorithm, we set the *tol* in Algorithm 2 to be 10^{-7}. The results are summarized in Table 2 and Figure 1. Powered by second-order optimization algorithms, CVX can provide fast and accurate solutions for medium-size problems but would suffer from great computational burden for large-scale ones. Therefore we only report the results up to $5,000$ dimensions.

From the above results we can observe that for projections of a moderate size, all three algorithms perform well. However, for large-scale ones, the advantage of the proposed algorithm is evident as our method provides more accurate solution with less time.

Table 2. Distance between the optimal solution of projection algorithms and that of the CVX. All the results are averaged over 100 replications.

Methods	50	100	500	1000	5000
Dykstra	9.00	9.81	11.40	11.90	12.42
ADMM	0.64	0.08	3.6e-3	6.3e-3	1.3e-2
ours	1.4e-3	1.1e-3	1.2e-3	1.7e-3	7.3e-3

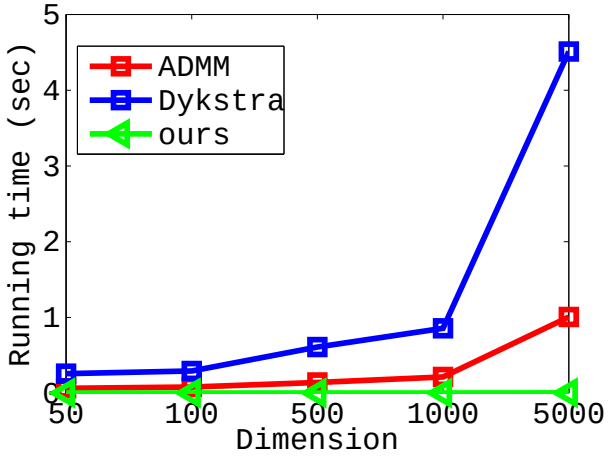

Figure 1. The average running time for different algorithms to achieve the precision level listed in Table 2.

6.2. Performance on Synthetic Data

We generate a 60×100 matrix $\boldsymbol{A}$, whose entries follow i.i.d standard normal distribution. The 100 features (columns) are partitioned into 10 groups of equal size. The ground truth vector $\boldsymbol{x}_0$ possesses nonzero elements only in 4 of the 10 groups. In addition, only 4 elements in each nonzero group are nonzero. Finally $\boldsymbol{y}$ is generated according to $\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{z}$ with $\boldsymbol{z}$ following distribution $\mathcal{N}(0, 0.5^2)$. The data are divided into training and testing set of equal size.

We fit our method to the training set and compare with both convex methods (lasso, group lasso and sparse group lasso) and methods based on nonconvex bi-level penalties (group bridge and composite MCP). Since the data are intentionally generated to be sparse in both group-level and feature-level, approaches that only perform group selection, such as group lasso, group SCAD and ordinary group MCP, are not included due to their suboptimal results.

The tuning parameters of the convex methods are selected from $\{0.01, 0.1, 1, 10\}$, whereas for our method, the number of nonzero groups is selected from the set $\{2, 4, 6, 8\}$ and the number of features is chosen from $\{2s_2, 4s_2, 6s_2, 8s_2\}$. 10-fold cross validation is taken for

parameter tuning. Group bridge and composite MCP are carried out using their original R-package **grpreg** and the tuning parameters are set to the default values (100 parameters with 10-fold cross-validation).

Following similar settings in (Breheny & Huang, 2009), we list the number of selected groups and features by each method. In addition, the number of false positive or false negative groups/features are also reported in Table 3. We can observe that our model correctly identifies the underlying groups and features. Moreover, our method effectively excludes redundant features and groups compared to other methods, which is illustrated by our low false positive numbers and relatively high false negative numbers. Such a phenomenon also appears in the evaluations in (Breheny & Huang, 2009).

Table 3. Comparison of performance on synthetic data. All the results are averaged for 100 replications.

Methods	Groups			Features		
	NO.	FP	FN	NO.	FP	FN
lasso	7.56	3.85	0.29	17.37	9.84	8.47
sgl	7.29	3.68	0.39	17.68	10.13	8.45
ours	3.37	0.81	1.44	11.70	5.97	10.27
cMCP	9.5	5.7	0.2	8.02	3.4	11.38
gBrdg	10	6	0	72.8	57.92	1.12

6.3. Performance on Real-world Application

Our method is further evaluated on the application of examining Electroencephalography (EEG) correlates of genetic predisposition to alcoholism (Frank & Asuncion, 2010). EEG records the brain's spontaneous electrical activity by measuring the voltage fluctuations over multiple electrodes placed on the scalp. This technology has been widely used in clinical diagnosis, such as coma, brain death and genetic predisposition to alcoholism. In fact, encoded in the EEG data is a certain group structure, since each electrode records the electrical activity of a certain region of the scalp. Identifying and utilizing such spatial information has the potential of increasing stability of a prediction.

The training set contains 200 samples of 16384 dimensions, sampled from 64 electrodes placed on subject's scalps at 256 Hz (3.9-msec epoch) for 1 second. Therefore, the data can naturally be divided into 64 groups of size 256. We apply the lasso, group lasso, sparse group lasso, group SCAD, group MCP, group bridge, composite MCP and our proposed method on the training set and adapt the 5-fold cross-validation for selecting tuning parameters. More specifically, for lasso and group lasso, the candidate tuning parameters are specified by 10 parameters[4] sampled using the logarithmic scale from the parameter spaces, while for the sparse group lasso, the parameters form a 10×10 grid[5], sampled from the parameter space in logarithmic scale. For our method, the number of groups is selected from the set: $s_2 = \{30, 40, 50\}$ and s_1, the number of features is chosen from the set $\{50s_2, 100s_2, 150s_2\}$. Default settings in the R package **grpreg** (100 parameters, 10-fold cross validation) are applied to other nonconvex methods. The accuracy of classification together with the number of selected features and groups over a test set, which also contains 200 samples, are reported in Table 4. Clearly our method achieves the best performance of classification. Note that, although lasso's performance is almost as good as ours with even less features, however, it fails to identify the underlying group structure in the data, as revealed by the fact all 64 groups are selected. Moreover, other nonconvex approaches such as the group SCAD, group MCP and group bridge seem to over-penalized the group penalty, which results in very few selected groups and suboptimal performance.

Table 4. Comparison of performance on EEG data.

Methods	Accuracy	# Feature	# Group
lasso	67.0	2068	64
glasso	62.5	8704	34
sglasso	65.5	4834	61
ours	68.0	3890	25
gSCAD	63.0	1792	7
gMCP	55.0	256	1
cMCP	65.5	62	35
gBrdg	51.5	80	2

7. Conclusion and Future Work

This paper expands a nonconvex paradigm into sparse group feature selection. In particular, an efficient optimization scheme is developed based on the DC programming, accelerated gradient method and efficient projection. In addition, theoretical properties on the accuracy of selection and parameter estimation are analyzed. The efficiency and efficacy of the proposed method are validated on both synthetic data and real-world applications. The proposed method will be further investigated on real-world applications involving the group structure. Moreover, extending our approach to multi-modal multi-task learning (Zhang & Shen, 2011) is another promising direction.

[4] $\lambda_{\text{lasso}} = \text{logspace}(10^{-3}, 1)$, $\lambda_{\text{glasso}} = \text{logspace}(10^{-2}, 1)$

[5] The product space of $\lambda_{\text{lasso}} \times \lambda_{\text{glasso}}$

References

Bach, Francis, Jenatton, Rodolphe, Mairal, Julien, and Obozinski, Guillaume. *Convex Optimization with Sparsity-Inducing Norms*. 2010.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Bickel, P.J., Ritov, Y., and Tsybakov, A.B. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers, 2011.

Breheny, P. and Huang, J. Penalized methods for bi-level variable selection. *Statistics and its interface*, 2(3):369, 2009.

Combettes, P.L. and Pesquet, J.C. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 2010.

Donoho, D.L. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 2002. ISSN 0018-9448.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 272–279. ACM, 2008.

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Frank, A. and Asuncion, A. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

Friedman, J., Hastie, T., and Tibshirani, R. A note on the group lasso and a sparse group lasso. *Arxiv preprint arXiv:1001.0736*, 2010.

Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, April 2011.

Huang, J. and Zhang, T. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

Huang, J., Ma, S., Xie, H., and Zhang, C.H. A group bridge approach for variable selection. *Biometrika*, 96 (2):339–355, 2009.

Huang, J., Breheny, P., and Ma, S. A selective review of group selection in high dimensional models. *arXiv preprint arXiv:1204.6491*, 2012.

Kolmogorov, A.N. and Tihomirov, V.M. *e-Entropy and e-capacity of sets in functional spaces*. American Mathematical Society, 1961.

Mazumder, R., Friedman, J.H., and Hastie, T. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

Meier, L., Van De Geer, S., and Bühlmann, P. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1): 53–71, 2008.

Natarajan, B. K. Sparse approximation solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995.

Nesterov, Y. Gradient methods for minimizing composite objective function. *CORE Discussion Papers*, 2007.

Schmidt, M., Le Roux, N., Bach, F., et al. Convergence rates of inexact proximal-gradient methods for convex optimization. In *NIPS'11-25 th Annual Conference on Neural Information Processing Systems*, 2011.

Shen, X., Pan, W., and Zhu, Y. Likelihood-based selection and sharp parameter estimation. *Journal of American Statistical Association*, 107:223–232, 2012.

Tao, P.D. and An, LTH. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Math. Vietnam*, 22(1):289–355, 1997.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 267–288, 1996.

Van De Geer, S.A. and Bühlmann, P. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

Wang, L., Chen, G., and Li, H. Group scad regression analysis for microarray time course gene expression data. *Bioinformatics*, 23(12):1486–1494, 2007.

Yang, H., Xu, Z., King, I., and Lyu, M. Online learning for group lasso. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 1191–1198. ACM, 2010.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Zhang, D. and Shen, D. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease, 2011.

Zhang, T. Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, 11:1081–1107, 2010.

Zhang, T. Multi-stage convex relaxation for feature selection. *arXiv:1106.0565v2*, 2011.

Zhao, P. and Yu, B. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7: 2541–2563, 2006.