Supplementary Material for paper: Efficient Sparse Group Feature Selection via Nonconvex Optimization

A. Proof of Theorem 3

The proof uses a large deviation probability inequality of (Wong & Shen, 1995) to treat one-sided log-likelihood ratios with constraints.

Let $\mathcal{S} = \{\boldsymbol{x}^\tau : \|\boldsymbol{x}^\tau\|_0 \le s_1^0, \|\boldsymbol{x}^\tau\|_{0,G} \le s_2^0\}$, $\|\boldsymbol{x}\|_0 = \sum_{j=1}^p I(|x_j| \ne 0)$ is the L_0-norm of $\boldsymbol{x}$, and $\|\boldsymbol{x}\|_{0,G} = \sum_{j=1}^{|G|} I(\|\boldsymbol{x}_j\|_2 \ne 0)$ is the L_0-norm over the groups. Now we partition $\mathcal{S}$. Note that for $C \subset (G_1, \cdots, G_{|G|})$, it can be partitioned into $C = (C \setminus C^0) \cup (C \cap C^0)$. Then

$$\mathcal{S} = \bigcup_{i=0}^{s_2^0} \bigcup_{C \in \mathcal{B}_i} \mathcal{S}_{A_C, C},$$

where $S_{A_C,C} = \{\boldsymbol{x}^\tau \in \mathcal{S} : C(\boldsymbol{x}) = C = (G_{i_1}, \cdots, G_{i_k}), \sum_j |A_{G_j}| \le s_1^0\}$, and $\mathcal{B}_i = \{C \ne C_0 : |C^0 \setminus C| = i, |C| \le s_2^0\}$, with $|\mathcal{B}_i| = \binom{s_2^0}{s_2^0 - i} \sum_{j=0}^{i} \binom{|G|-s_2^0}{j}$; $i = 0, \cdots, s_2^0$.

To bound the error probability, let $L(\boldsymbol{x}) = -\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$ be the likelihood. Note that

$$\{\hat{\boldsymbol{x}} \ne \hat{\boldsymbol{x}}^o\} \subseteq \{L(\hat{\boldsymbol{x}}) - L(\hat{\boldsymbol{x}}^o) \ge 0\} \subseteq \{L(\hat{\boldsymbol{x}}) - L(\boldsymbol{x}^0) \ge 0\}.$$

This together with $\{\hat{\boldsymbol{x}} \ne \hat{\boldsymbol{x}}^o\} \subseteq \{\hat{\boldsymbol{x}} \in \mathcal{S}\}$ implies that

$$\{\hat{\boldsymbol{x}} \ne \hat{\boldsymbol{x}}^o\} \subseteq \{L(\hat{\boldsymbol{x}}) - L(\boldsymbol{x}^0) \ge 0\} \cap \{\hat{\boldsymbol{x}} \in \mathcal{S}\}.$$

Consequently,

$$\begin{aligned}
&I \equiv P\big(\hat{\boldsymbol{x}} \ne \hat{\boldsymbol{x}}^o\big) \\
&\le P\Big(L(\hat{\boldsymbol{x}}) - L(\boldsymbol{x}^0) \ge 0; \hat{\boldsymbol{x}} \in \mathcal{S}\Big) \\
&\le \sum_{i=1}^{s_2^0} \sum_{C \in \mathcal{B}_i} \sum_{S_{A_C,C}} P^*\Big(\sup_{\boldsymbol{x} \in \mathcal{S}_{A_C,C}} \big(L(\boldsymbol{x}) - L(\boldsymbol{x}^0)\big) \ge 0\Big) \\
&\le \sum_{i=1}^{s_2^0} \sum_{j=1}^{s_1^0} \sum_{|C|=i, |A_G|=j} P^*\Big(\sup_{\{-\log(1-h^2(\boldsymbol{x},\boldsymbol{x}^0)) \ge \max(i,1) C_{\min}(\boldsymbol{x}^0) - d_3 \tau^{d_2} p, \boldsymbol{x} \in \mathcal{S}_{A_C,C}\}} \big(L(\boldsymbol{x}) - L(\boldsymbol{x}^0)\big) \ge 0\Big),
\end{aligned}$$

where P^* is the outer measure and the last two inequalities use the fact that $\mathcal{S}_{A_C,C} \subseteq \{\boldsymbol{x} \in \mathcal{S}_{A_C,C} : \max(|C^0 \setminus C|, 1) C_{\min}(\boldsymbol{x}^0) \le -\log(1 - h^2(\boldsymbol{x}, \boldsymbol{x}^0))\} \subseteq \{-\log(1 - h^2(\boldsymbol{x}, \boldsymbol{x}^0)) \ge d_1 \max(i, 1) C_{\min}(\boldsymbol{x}^0) - d_3 \tau^{d_2} p\}$, under Assumption 3.

For I, we apply Theorem 1 of (Wong & Shen, 1995) to bound each term. Towards this end, we verify their entropy condition (3.1) for the local entropy over $\mathcal{S}_{A_C,C}$ for $|C| = 1, \cdots, s_2^0$ and $|A| = 1, \cdots, s_1^0$. Under Assumption 2 $\varepsilon = \varepsilon_{n,p} = (2c_0)^{1/2} c_4^{-1} \log(2^{1/2}/c_3) \log p(\frac{s_1^0}{n})^{1/2}$ satisfies there with respect to $\varepsilon > 0$, that is,

$$\sup_{\{0 \le |A| \le p_0\}} \int_{2^{-8}\varepsilon^2}^{2^{1/2}\varepsilon} H^{1/2}(t/c_3, \mathcal{F}_{ji}) dt \le p_0^{1/2} 2^{1/2} \varepsilon \log(2/2^{1/2} c_3) \le c_4 n^{1/2} \varepsilon^2. \quad (16)$$

for some constant $c_3 > 0$ and $c_4 > 0$, say $c_3 = 10$ and $c_4 = \frac{(2/3)^{5/2}}{512}$. By Assumption 2, $C_{\min}(\boldsymbol{x}^0) \ge \varepsilon^2_{n,p_0,p}$ implies (16), provided that $s_1^0 \ge (2c_0)^{1/2} c_4^{-1} \log(2^{1/2}/c_3)$.

Note that $|\mathcal{B}_i| = \binom{s_2^0}{s_2^0 - i} \sum_{j=0}^{i} \binom{|G|-s_2^0}{j} \le (|G|(|G| - s_2^0)^i \le (|G|^2/4)^i$ by the binomial coefficients formula. Moreover, $\sum_{j=1}^{s_1^0} 2^j i^j \le i^{s_1^0}$, and $\sum_{j_1+\cdots+j_i=j} \binom{j}{j_1, \cdots j_i} 2^j = (2i)^j$ using the Multinomial Theorem. By Theorem 1 of (Wong & Shen, 1995), there exists a constant $c_2 > 0$, say $c_2 = \frac{4}{27}\frac{1}{1926}$,

$$\begin{aligned} I &\leq \sum_{i=1}^{s_2^0} |\mathcal{B}_i| \sum_{j=1}^{s_1^0} \sum_{(j_1,\cdots j_i)} \binom{j}{j_1,\cdots j_i} 2^{j_1}\cdots 2^{j_i} \exp\big(-c_2 n i C_{\min}(\boldsymbol{x}^0)\big) \\ &\leq \sum_{i=1}^{s_2^0} \exp\big(-c_2 n i C_{\min}(\boldsymbol{x}^0) + 2i(\log|G| + \log s_1^0)\big) \\ &\leq \exp\big(-c_2 n C_{\min}(\boldsymbol{x}^0) + 2(\log|G| + \log s_1^0)\big). \end{aligned}$$

Let $G = \{\hat{\boldsymbol{x}} \neq \hat{\boldsymbol{x}}^0\}$. For the risk property, $Eh^2(\hat{\boldsymbol{x}}, \boldsymbol{x}^0) = Eh^2(\hat{\boldsymbol{x}}^0, \boldsymbol{x}^0) + Eh^2(\hat{\boldsymbol{x}}, \boldsymbol{x}^0)I(G)$ is upper bounded by

$$Eh^2(\hat{\boldsymbol{x}}, \boldsymbol{x}^0) + \exp\big(-c_2 n C_{\min}(\boldsymbol{x}^0) + 2(\log|G| + \log s_1^0)\big) = (1 + o(1))Eh^2(\hat{\boldsymbol{x}}^0, \boldsymbol{x}^0),$$

using the fact that $h(\hat{\boldsymbol{x}}, \boldsymbol{x}^0) \leq 1$. This completes the proof.

B. Accelerated Gradient Method

The AGM procedure is listed in Algorithms 3, in which $f(\boldsymbol{x})$ is the objective function $\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ with $\nabla f(\boldsymbol{x})$ denotes its gradient at $\boldsymbol{x}$. In addition, $f_{L,\boldsymbol{u}}(\boldsymbol{x})$ is the linearization of $f(\boldsymbol{x})$ at $\boldsymbol{u}$ defined as follows:

$$f_{L,\boldsymbol{u}}(\boldsymbol{x}) = f(\boldsymbol{u}) + \nabla f(\boldsymbol{u})^T(\boldsymbol{x} - \boldsymbol{u}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2.$$

Algorithm 3 Accelerated Gradient Method (Nesterov, 2007; Beck & Teboulle, 2009) for (7)

Input: $\boldsymbol{A}$, $\boldsymbol{y}$, s_1, s_2, L_0, $\boldsymbol{x}_0$,
Output: solution $\boldsymbol{x}$ to (7)

1: **Initialize**: L_0, $\boldsymbol{x}_1 = \boldsymbol{x}_0$, $\alpha_{-1} = 0$, $\alpha_0 = 1$, $t = 0$.
2: **repeat**
3: $t = t + 1$, $\beta_t = \frac{\alpha_{t-2} - 1}{\alpha_{t-1}}$, $\boldsymbol{u}_t = \boldsymbol{x}_t + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$
4: **Line search**: Find the smallest $L = 2^j L_{t-1}$ such that

$$f(\boldsymbol{x}_{t+1}) \leq f_{L,\boldsymbol{u}_t}(\boldsymbol{x}_{t+1}),$$

where $\boldsymbol{x}_{t+1} = \mathrm{SGLP}(\boldsymbol{u}_t - \frac{1}{L}\nabla f(\boldsymbol{u}_t), s_1, s_2)$
5: $\alpha_{t+1} = \frac{1 + \sqrt{1 + 4\alpha_t^2}}{2}$, $L_t = L$.
6: **until** Converge
7: **return** $\boldsymbol{x}_t$

C. Proof of Theorem 2

We utilize an intermediate lemma from (Bonnans & Shapiro, 1998):

Lemma 2. *Let X be a metric space and U be a normed space. Suppose that for all $x \in X$, the function $\psi(x, \cdot)$ is differentiable and that $\psi(x, Y)$ and $D_Y\psi(x, Y)$ (the partial derivative of $\psi(x, Y)$ with respect to Y) are continuous on $X \times U$. Let Φ be a compact subset of X. Define the optimal value function as $\phi(Y) = \inf_{x\in\Phi} \psi(x, Y)$. The optimal value function $\phi(Y)$ is directionally differentiable. In addition, if for any $Y \in U$, $\psi(\cdot, Y)$ has a unique minimizer $x(Y)$ over Φ, then $\phi(Y)$ is differentiable at Y and the gradient of $\phi(Y)$ is given by $\phi'(Y) = D_Y\psi(x(Y), Y)$.*

Proof of Theorem 2. Since both constraints are active, if $(x, \lambda, \eta) = \mathrm{SGLP}(v, s_1, s_2)$, then x and λ are also the optimal solutions to the following problem:

$$\underset{\lambda}{\text{maximize}}\ \underset{x\in X}{\text{minimize}} \quad \psi(x, \lambda) = \frac{1}{2}\|x - v\|_2^2 + \lambda(\|x\|_1 - s_1),$$

where $X = \{x : \|x\|_G \leq s_2\}$. By Lemma 2, $\phi(\lambda) = \inf_{x\in X} \psi(x, \lambda)$ is differentiable with the derivative given by $\|x\|_1$. In addition, as a pointwise infimum of a concave function, so does $\phi(\lambda)$ (Boyd & Vandenberghe, 2004) and its derivative, $\|x\|_1$, is non-increasing. Therefore $s_1 = \|x\|_1$ is non-decreasing as λ becomes smaller. This completes the proof. □

D. Algorithm for Solving (8)

Based on the analysis in Section 3.2, we give a detailed description of the sparse group lasso projection algorithm in Algorithm 4:

Algorithm 4 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 $\widehat{\eta}$ 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 $\boldsymbol{x}^*$

E. Algorithm for Solving (13)

We give a detailed description of algorithm for solving the restricted projection (13) in Algorithm 5.

F. The ADMM Projection algorithm

Alternating Direction Method of Multipliers (ADMM) is widely chosen for its capability of decomposing coupled variables/constraints, which is exactly the case in our projection problem. Before applying ADMM, we transform (8) into an equivalent form as follows:

$$\begin{aligned}\underset{x}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{u}\|_1 \le s_1 \\ & \|\boldsymbol{w}\|_G \le s_2 \\ & \boldsymbol{u}=\boldsymbol{x}, \boldsymbol{w}=\boldsymbol{x}.\end{aligned}$$

The augmented Lagrangian is:

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\eta}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 + \boldsymbol{\lambda}^T(\boldsymbol{u}-\boldsymbol{x}) + \boldsymbol{\eta}^T(\boldsymbol{w}-\boldsymbol{x}) + \frac{\rho}{2}(\|\boldsymbol{u}-\boldsymbol{x}\|_2^2 + \|\boldsymbol{w}-\boldsymbol{x}\|_2^2).$$

Utilize the scaled form (Boyd et al., 2011), i.e., let $\boldsymbol{\lambda} = \frac{\boldsymbol{\lambda}}{\rho}$, $\boldsymbol{\eta} = \frac{\boldsymbol{\eta}}{\rho}$, we can obtain an equivalent augmented Lagrangian:

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\eta}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 + \frac{\rho}{2}(\|\boldsymbol{x}-\boldsymbol{u}-\boldsymbol{\lambda}\|_2^2 + \|\boldsymbol{x}-\boldsymbol{w}-\boldsymbol{\eta}\|_2^2) - \frac{\rho}{2}(\|\boldsymbol{\lambda}\|_2^2 + \|\boldsymbol{\eta}\|_2^2).$$

Now we calculate the optimal $\boldsymbol{x}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\eta}$ through alternating minimization. For fixed $\boldsymbol{u}$ and $\boldsymbol{w}$, the optimal $\boldsymbol{x}$ possesses a closed-form solution:

$$\boldsymbol{x} = \frac{1}{1+2\rho}\left(\boldsymbol{v} + \rho(\boldsymbol{u}+\boldsymbol{\lambda}+\boldsymbol{w}+\boldsymbol{\eta})\right).$$

For fixed $\boldsymbol{x}$ and $\boldsymbol{u}$, finding the optimal $\boldsymbol{w}$ is a group lasso projection:

$$\begin{aligned}\underset{\boldsymbol{w}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}-(\boldsymbol{x}-\boldsymbol{\eta})\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{w}\|_G \le s_2\end{aligned} \tag{17}$$

For fixed $\boldsymbol{x}$ and $\boldsymbol{w}$, finding the optimal $\boldsymbol{u}$ amounts to solve an L_1-ball projection:

$$\begin{aligned}\underset{\boldsymbol{u}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{u}-(\boldsymbol{x}-\boldsymbol{\lambda})\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{u}\|_1 \le s_1.\end{aligned} \tag{18}$$

The update of multipliers is standard as follows:

$$\begin{aligned}\boldsymbol{\lambda} &= \boldsymbol{\lambda} + \boldsymbol{u} - \boldsymbol{x} \\ \boldsymbol{\eta} &= \boldsymbol{\eta} + \boldsymbol{w} - \boldsymbol{x}\end{aligned} \tag{19}$$

Algorithm 6 summarizes the above procedure. Note that, the value of the penalty term ρ is fixed in Algorithm 6. However, in our implementation, we increase ρ whenever necessary to obtain faster convergence.

G. The Dykstra's Algorithm

The Dykstra's algorithm is a general scheme to compute the projection onto intersections of convex sets. It is carried out by taking Euclidean projections onto each convex set alternatively in a smart way and is guaranteed to converge for least squares objective function (Combettes & Pesquet, 2010). The details of applying Dykstra's Algorithm to our projection problem are listed in Algorithm 7.

Algorithm 5 Restricted Sparse Group Lasso Projection Algorithm

Input: $\boldsymbol{v}$, s_1, s_2, T_1, T_3
Output: an optimal solution $\boldsymbol{x}$ to the Restricted Sparse Group Projection Problem (13)
Function RSGLP($\boldsymbol{v}$, s_1, s_2, T_1, T_3)

if $\|\boldsymbol{x}^{T_1}\|_1 \leq s_1$ **and** $\|\boldsymbol{x}^{T_3}\|_G \leq s_2$ **then**
  return $\boldsymbol{v}$
end if
$\boldsymbol{x}_{C_1}^{(T_1)^c} = \boldsymbol{v}^{(T_1)^c}$, $\boldsymbol{x}_{C_1}^{T_1} = \mathcal{P}_1^{s_1}(\boldsymbol{v}^{T_1})$
$\boldsymbol{x}_{C_2}^{(T_3)^c} = \boldsymbol{v}^{(T_3)^c}$, $\boldsymbol{x}_{C_2}^{T_3} = \mathcal{P}_G^{s_2}(\boldsymbol{v}^{T_3})$
$\boldsymbol{x}_{C_{12}}^{(T_1)^c} = \boldsymbol{v}^{(T_1)^c}$, $\boldsymbol{x}_{C_{12}}^{T_1}=$ bisec($\boldsymbol{v}$, s_1, s_2, T_1, T_3)
if $\|\boldsymbol{x}_{C_1}^{T_3}\|_G \leq s_2$ **then**
  return $\boldsymbol{x}_{C_1}$
else if $\|\boldsymbol{x}_{C_2}^{T_1}\|_1 \leq s_1$ **then**
  return $\boldsymbol{x}_{C_2}$
else
  return $\boldsymbol{x}_{C_{12}}$
end if

Function bisec($\boldsymbol{v}$, s_1, s_2, T_1, T_3)

Initialize up, low and tol
while $up - low > tol$ **do**
  $\hat{\lambda} = (low + up)/2$
  if (15) has a solution $\hat{\eta}$ given $v^{\hat{\lambda}}$ **then**
    calculate $\widehat{s_1}$ using $\widehat{\eta}$ and $\hat{\lambda}$.
    if $\hat{s_1} \leq s_1$ **then**
      $up = \hat{\lambda}$
    else
      $low = \hat{\lambda}$
    end if
  else
    $up = \hat{\lambda}$
  end if
end while
$\lambda^* = up$
Solve (15) to get η^*
Calculate $(\boldsymbol{x}^*)^{T_1}$ from λ^* and η^*.
return $(\boldsymbol{x}^*)^{T_1}$

Algorithm 6 ADMM (Boyd et al., 2011) for (8)

Input: $\boldsymbol{v}$, s_1, s_2
Output: an optimal solution x to (8)
  Initialize: $\boldsymbol{x}_0$, $\boldsymbol{u}_0$, $\boldsymbol{w}_0$, $\boldsymbol{\lambda}_0$, $\boldsymbol{\eta}_0$, $t = 0$, $\rho > 0$
  repeat
    $t = t + 1$
    $\boldsymbol{x}_t = \frac{1}{1+2\rho}\left(\boldsymbol{v} + \rho(\boldsymbol{u}_{t-1} + \boldsymbol{\lambda}_{t-1} + \boldsymbol{w}_{t-1} + \boldsymbol{\eta}_{t-1})\right)$
    $\boldsymbol{w}_t = \mathcal{P}_G^{s_2}(\boldsymbol{x}_t - \boldsymbol{\eta}_{t-1})$
    $\boldsymbol{u}_t = \mathcal{P}_1^{s_1}(\boldsymbol{x}_t - \boldsymbol{\lambda}_{t-1})$
    $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \boldsymbol{u}_t - \boldsymbol{x}_t$, $\boldsymbol{\eta}_t = \boldsymbol{\eta}_{t-1} + \boldsymbol{w}_t - \boldsymbol{x}_t$.
  until Converge
  return $\boldsymbol{x}_t$

Algorithm 7 Dykstra's Algorithm (Combettes & Pesquet, 2010) for (8)

Input: $\boldsymbol{v}$, s_1, s_2
Output: an optimal solution x to (8)
  Initialize: $\boldsymbol{x}_0 = \boldsymbol{v}$, $\boldsymbol{p}_0 = \boldsymbol{0}$, $\boldsymbol{q}_0 = \boldsymbol{0}$, $t = 0$
  repeat
    $t = t + 1$
    $\boldsymbol{y}_{t-1} = \mathcal{P}_G^{s_2}(\boldsymbol{x}_{t-1} + \boldsymbol{p}_{t-1})$
    $\boldsymbol{p}_t = \boldsymbol{x}_{t-1} + \boldsymbol{p}_{t-1} - \boldsymbol{y}_{t-1}$
    $\boldsymbol{x}_t = \mathcal{P}_1^{s_1}(\boldsymbol{y}_{t-1} + \boldsymbol{q}_{t-1})$
    $\boldsymbol{q}_t = \boldsymbol{y}_{t-1} + \boldsymbol{q}_{t-1} - \boldsymbol{x}_t$
  until Converge
  return $\boldsymbol{x}_t$