

# Hierarchical Overlapping Group Lasso for GMANOVA Model

Mineaki Ohishi<sup>1\*</sup>, Isamu Nagai<sup>2</sup>, Ryoya Oda<sup>3</sup>  
and Hirokazu Yanagihara<sup>4,5</sup>

<sup>1</sup>Center for Data-driven Science and Artificial Intelligence, Tohoku University  
Kawauchi 41, Aoba-ku, Sendai 980-8576, Japan

<sup>2</sup>Faculty of Liberal Arts and Science, Chukyo University  
101-2 Yagoto Honmachi, Showa-ku, Nagoya 466-8666, Japan

<sup>3</sup>Graduate School of Advanced Science and Engineering, Hiroshima University  
1-3-1 Kagamiyama, Higashi-Hiroshima 739-8526, Japan

<sup>4</sup>Osaka Central Advanced Mathematical Institute, Osaka Metropolitan University  
3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan

<sup>5</sup>Research & Development Center, Osaka Medical and Pharmaceutical University  
2-7 Daigaku-machi, Takatsuki, Osaka 569-8686, Japan

## Abstract

This paper deals with the GMANOVA model with a matrix of polynomial basis functions as a within-individual design matrix. The model involves two model selection problems: the selection of explanatory variables and the selection of the degrees of the polynomials. The two problems can be uniformly addressed by hierarchically incorporating zeros into the vectors of regression coefficients. Based on this idea, we propose hierarchical overlapping group Lasso (HOGL) to perform the variable and degree selections simultaneously. Importantly, when using a polynomial basis, fitting a high-degree polynomial often causes problems in model selection. In the approach proposed here, these problems are handled by using a matrix of orthonormal basis functions obtained by transforming the matrix of polynomial basis functions. Algorithms are developed with optimality and convergence to optimize the method. The performance of the proposed method is evaluated using numerical simulation.

(Last Modified: October 23, 2025)

**Key words:** Block-wise coordinate descent method, GMANOVA model, Group Lasso, Growth curve model, MM algorithm, Model selection.

\*Corresponding author

E-mail address: mineaki.ohishi.a4@tohoku.ac.jp (Mineaki Ohishi)

## 1. Introduction

Suppose we have observations at  $p$  common time points for  $n$  ( $> p$ ) individuals and define  $\mathbf{Y}$  as an  $n \times p$  matrix of the observations. Let  $\mathbf{A}$  and  $\mathbf{X}$  be an  $n \times k$  ( $n > k$ ) between-individual design matrix and a  $p \times q$  ( $p \geq q$ ) within-individual design matrix, respectively. For these matrices, we consider the following GMANOVA model (Potthoff & Roy, 1964):

$$\mathbf{Y} = \mathbf{1}_n \boldsymbol{\mu}' + \mathbf{A} \boldsymbol{\Theta} \mathbf{X}' + \boldsymbol{\mathcal{E}}, \quad E[\boldsymbol{\mathcal{E}}] = \mathbf{O}_{n,p}, \quad \text{Cov}[\text{vec}(\boldsymbol{\mathcal{E}})] = \boldsymbol{\Sigma} \otimes \mathbf{I}_n, \quad (1.1)$$

where  $\mathbf{1}_m$  is an  $m$ -dimensional vector of ones,  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)'$  is a  $p$ -dimensional vector of location parameters,  $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_k)'$  is a  $k \times q$  matrix of regression coefficients,  $\mathbf{O}_{r,s}$  is an  $r \times s$  matrix of zeros,  $\boldsymbol{\Sigma}$  is a  $p \times p$  covariance matrix,  $\text{vec}(\cdot)$  represents a vec operator, and  $\otimes$  represents the Kronecker product. Furthermore, suppose that  $\mathbf{A}$  is centralized (i.e.,  $\mathbf{A}'\mathbf{1}_n = \mathbf{0}_k$ ) and  $\mathbf{A}$  and  $\mathbf{X}$  are column full rank (i.e.,  $\text{rank}(\mathbf{A}) = k$ ,  $\text{rank}(\mathbf{X}) = q$ ) and standardized such that a norm of each column vector is one, where  $\mathbf{0}_m$  is an  $m$ -dimensional vector of zeros. Note that the GMANOVA model is a generalized version of a multivariate linear regression model (e.g., Srivastava, 2002, Chap. 9; Timm, 2002, Chap. 4). When  $p = q$  and  $\mathbf{X} = \mathbf{I}_p$ , the GMANOVA model reduces to a multivariate linear regression model. Furthermore, when  $p = 1$ , the GMANOVA model reduces to a univariate linear regression model.

The GMANOVA model is also referred to as the growth curve model (e.g., von Rosen, 1991; Kshirsagar & Smith, 1995) and has aspects of a varying coefficient model (Satoh & Yanagihara, 2010). Let  $t_1, \dots, t_p$  be common time points of  $n$  observations and define  $\mathbf{X} = (\mathbf{x}(t_1), \dots, \mathbf{x}(t_p))'$ , where  $\mathbf{x}(t)$  is a  $q$ -dimensional vector of basis functions at time  $t$ . Then,  $y_{i,j}$ , the  $(i, j)$ th element of  $\mathbf{Y}$ , can be expressed as the following varying coefficient model:

$$y_{i,j} = \mu_j + \sum_{\ell=1}^k a_{i,\ell} \beta_{\ell}(t_j) + \varepsilon_{i,j}, \quad \beta_{\ell}(t) = \boldsymbol{\theta}'_{\ell} \mathbf{x}(t),$$

where  $a_{i,j}$  and  $\varepsilon_{i,j}$  are the  $(i, j)$ th elements of  $\mathbf{A}$  and  $\boldsymbol{\mathcal{E}}$ , respectively, and  $a_{i,\ell}$  is the  $\ell$ th explanatory variable of the  $i$ th individual. This paper adopts the following vector of polynomial basis functions:

$$\mathbf{x}(t) = (c_{q-1} t^{q-1}, \dots, c_1 t, p^{-1/2})', \quad c_j = \left( \sum_{\ell=1}^p t_{\ell}^{2j} \right)^{-1/2},$$

where  $c_j$  is a constant standardizing column vectors of  $\mathbf{X}$ . This implies that the time trend for each explanatory variable is modeled by the following polynomial of degree  $(q - 1)$ :

$$\beta_{\ell}(t) = \theta_{\ell,1} c_{q-1} t^{q-1} + \dots + \theta_{\ell,q-1} c_1 t + \theta_{\ell,q} p^{-1/2}, \quad (1.2)$$

where  $\theta_{\ell,j}$  is the  $j$ th element of  $\boldsymbol{\theta}_\ell$ . We consider model selection for the GMANOVA model (1.1) with the above settings.

As an example of model selection for a GMANOVA model, Fujikoshi & Rao (1991) dealt with selection of the explanatory variables, while Satoh *et al.* (1997) and Enomoto *et al.* (2015) dealt with selection of the degrees of the polynomial basis functions. Although they are typical approaches that select the best model from candidates based on a model selection criterion, sparse estimation methods offer a different approach and are popular for model selection. Sparse estimation methods can perform parameter estimation and model selection simultaneously by shrinking the parameters towards zero and allowing some estimates to be exactly zero. These methods are known to be useful for models with a large number of parameters. Following the proposal of Lasso by Tibshirani (1996), various specific methods have been offered, e.g., fused Lasso (Tibshirani *et al.*, 2005), group Lasso (Yuan & Lin, 2006), and sparse group Lasso (Simon *et al.*, 2013).

This paper proposes hierarchical overlapping group Lasso (HOGL) to perform the selection of the explanatory variables and the selection of the degrees of the polynomial basis functions simultaneously. Regarding variable selection, for multivariate models such as the GMANOVA model, the general approach is to select variables that affect at least one response variable (e.g., Obozinski *et al.*, 2008; Yanagihara & Oda, 2021). In our model, variable selection means the selection of column vectors of  $\mathbf{A}$  where  $\boldsymbol{\theta}_\ell = \mathbf{0}_q$  implies that the  $\ell$ th column vector (i.e., the  $\ell$ th explanatory variable) is removed from the model. Hence, variable selection can be performed by group Lasso for row vectors of  $\boldsymbol{\Theta}$ , which can be implemented by penalized estimation with  $\sum_{\ell=1}^k \|\boldsymbol{\theta}_\ell\|$ . On the other hand, regarding degree selection, by setting, for example, all the coefficients of the fourth degree and higher to zero, i.e.,  $\theta_{\ell,1} = \dots = \theta_{\ell,q-4} = 0$ ,  $\beta_\ell(t)$  reduces to a cubic polynomial. Hence, degree selection can be performed by hierarchically applying group Lasso to elements of  $\boldsymbol{\theta}_\ell$  beginning with higher-order coefficients, and we can consider employing  $\sum_{j=1}^q \|(\boldsymbol{\theta}_\ell)_{(j)}\|$  as a penalty term, where  $\boldsymbol{\gamma}_{(j)}$  is a sub-vector of an  $m$ -dimensional vector  $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_m)'$  defined by

$$\boldsymbol{\gamma}_{(j)} = (\gamma_1, \dots, \gamma_j)' \quad (j \in \{1, \dots, m\}), \quad (1.3)$$

and  $\boldsymbol{\gamma}_{(m)} = \boldsymbol{\gamma}$  holds. Based on the above, we define the HOGL penalty as

$$\begin{aligned} \Omega(\boldsymbol{\Theta} \mid \delta) &= \delta \sum_{\ell=1}^k \sum_{j=1}^q w_{\ell,j}^{(0)} \|(\boldsymbol{\theta}_\ell)_{(j)}\| + (1 - \delta) \sum_{\ell=1}^k w_{\ell,q}^{(0)} \|\boldsymbol{\theta}_\ell\| = \sum_{\ell=1}^k \sum_{j=1}^q w_{\ell,j}(\delta) \|(\boldsymbol{\theta}_\ell)_{(j)}\|, \quad (1.4) \\ w_{\ell,j}(\delta) &= \begin{cases} \delta w_{\ell,j}^{(0)} & (j = 1, \dots, q-1) \\ w_{\ell,q}^{(0)} & (j = q) \end{cases}, \quad \delta \in [0, 1], \end{aligned}$$

where  $\delta$  is a tuning parameter adjusting the balance of the penalties for the variable and de-

gree selections and  $w_{\ell,j}^{(0)} (> 0)$  is a penalty weight. With the HOGL penalty, estimations of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Theta}$  and the variable and degree selections are simultaneously performed by minimizing the following function:

$$\frac{1}{2} \text{tr} \left\{ (\mathbf{Y} - \mathbf{1}_n \boldsymbol{\mu}' - \mathbf{A} \boldsymbol{\Theta} \mathbf{X}')' (\mathbf{Y} - \mathbf{1}_n \boldsymbol{\mu}' - \mathbf{A} \boldsymbol{\Theta} \mathbf{X}') \mathbf{S}^{-1} \right\} + \lambda \Omega(\boldsymbol{\Theta} \mid \delta), \quad (1.5)$$

where  $\lambda (\geq 0)$  is a tuning parameter adjusting the strength of the penalty against model fitting and  $\mathbf{S}$  is an unbiased estimator of  $\boldsymbol{\Sigma}$  defined as

$$\mathbf{S} = \frac{\mathbf{Y}' \{ \mathbf{I}_n - \mathbf{1}_n \mathbf{1}_n' / n - \mathbf{A} (\mathbf{A}' \mathbf{A})^{-1} \mathbf{A}' \} \mathbf{Y}}{n - k - 1}.$$

As a model selection method based on sparse estimation for the GMANOVA model, weighted least squares estimation with group SCAD penalty (we call it wSCAD here) was proposed by Hu *et al.* (2014). wSCAD performs variable selection and degree selection simultaneously based on SCAD (Fan & Li, 2001) and is guaranteed to have oracle properties (Fan & Li, 2001) under specific conditions. One of the main differences between HOGL and wSCAD is their approach to degree selection. As described above, we can express  $\beta_\ell(t)$  as a cubic polynomial by setting all the coefficients of fourth degree and higher to zero. Whereas HOGL can set the coefficients to zero simultaneously, wSCAD sets the coefficients to zero individually. From this, we can expect that HOGL mitigates the risks of both missing the truly zero coefficients and mistakenly shrinking non-zero coefficients to zero.

To implement HOGL, an algorithm for minimizing (1.5) is important. Since  $\mathbf{A}$  is centralized, the estimator of  $\boldsymbol{\mu}$  is given by  $\hat{\boldsymbol{\mu}} = \mathbf{Y}' \mathbf{1}_n / n$ . Hence, it is sufficient to minimize (1.5) with respect to  $\boldsymbol{\Theta}$ . To achieve this, we propose an algorithm with optimality and convergence based on the MM philosophy (Hunter & Lange, 2004). Importantly, we construct a practical algorithm by deriving the update equation of a solution in closed form. Recognizing that using a polynomial basis can result in model unstableness in the case of high-degree polynomials, which, in turn, can lead to model misspecification in the model selection process, we transform the matrix of polynomial basis functions  $\mathbf{X}$  to a matrix of orthonormal basis functions. To choose the number of orthonormal basis functions for selecting the degrees of the original polynomial basis functions, we apply HOGL. To use HOGL with the matrix of orthonormal basis functions, we construct an algorithm based on the block-wise coordinate descent method. Moreover, we discuss an extension of HOGL. As shown in (1.4), a polynomial basis requires hierarchical overlapping by adding parameters one by one to select the degree of the polynomial. However, this is not necessarily the case for all basis types. For example, a Fourier basis has pairs sin and cos, and their selection requires hierarchical overlapping by adding parameters two by two. To select various types of basis functions, we extend HOGL to a flexible version of hierarchical overlapping.

The remainder of the paper is organized as follows. In Section 2, we establish the foundation of our study. In Section 3, we describe the study's main results. By deriving the update equation of a solution in closed form, we construct an algorithm to optimize HOGL. We also describe the transformation of the matrix of basis functions and our extension of HOGL. In Section 4, we numerically evaluate the performance of the proposed HOGL method. Section 5 concludes the paper. Technical details are provided in the Appendices.

## 2. Preliminaries

Since  $\mathbf{A}$  is centralized, the first term of (1.5) can be separated with respect to  $\boldsymbol{\mu}$  and  $\boldsymbol{\Theta}$  as

$$\frac{1}{2} \operatorname{tr} \left\{ (n\boldsymbol{\mu}\boldsymbol{\mu}' - \mathbf{Y}'\mathbf{1}_n\boldsymbol{\mu}' - \boldsymbol{\mu}\mathbf{1}_n'\mathbf{Y})\mathbf{S}^{-1} \right\} + \frac{1}{2} \operatorname{tr} \left\{ (\mathbf{Y} - \mathbf{A}\boldsymbol{\Theta}\mathbf{X}')'(\mathbf{Y} - \mathbf{A}\boldsymbol{\Theta}\mathbf{X}')\mathbf{S}^{-1} \right\}.$$

Let  $\mathbf{U} = \mathbf{Y}\mathbf{S}^{-1/2}$  and  $\mathbf{V} = \mathbf{S}^{-1/2}\mathbf{X}$ . Then, the second term of the above equation can be expressed as

$$\operatorname{RSS}(\boldsymbol{\theta}) = \frac{1}{2} \|\mathbf{u} - \mathbf{Z}\boldsymbol{\theta}\|^2 = \frac{1}{2} \operatorname{tr} \{ (\mathbf{U} - \mathbf{A}\boldsymbol{\Theta}\mathbf{V}')'(\mathbf{U} - \mathbf{A}\boldsymbol{\Theta}\mathbf{V}') \}, \quad (2.1)$$

where  $\mathbf{u} = \operatorname{vec}(\mathbf{U}')$ ,  $\boldsymbol{\theta} = \operatorname{vec}(\boldsymbol{\Theta}') = (\boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_k)'$ , and  $\mathbf{Z} = \mathbf{A} \otimes \mathbf{V}$ . Hence, the estimation problem of  $\boldsymbol{\Theta}$  based on minimizing (1.5) is equal to the minimization problem of the following penalized residual sum of squares (PRSS).

$$\operatorname{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda) = \operatorname{RSS}(\boldsymbol{\theta}) + \lambda\Omega(\boldsymbol{\Theta} \mid \delta). \quad (2.2)$$

To construct an algorithm to minimize (2.2), we define a surrogate function of RSS. Let  $L (> 0)$  be the maximum eigenvalue of  $\mathbf{Z}'\mathbf{Z}$  and  $\mathbf{r}(\boldsymbol{\theta})$  be a gradient vector of  $\operatorname{RSS}(\boldsymbol{\theta})$  as

$$\mathbf{r}(\boldsymbol{\theta}) = (\mathbf{r}_1(\boldsymbol{\theta})', \dots, \mathbf{r}_k(\boldsymbol{\theta})')' = \frac{\partial \operatorname{RSS}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \mathbf{Z}'\mathbf{Z}\boldsymbol{\theta} - \mathbf{Z}'\mathbf{u}.$$

Using them, we define

$$\begin{aligned} \operatorname{RSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}) &= L \sum_{\ell=1}^k \left[ \frac{1}{2} \|\boldsymbol{\theta}_\ell\|^2 - \left\{ \hat{\boldsymbol{\theta}}_\ell - \frac{\mathbf{r}_\ell(\hat{\boldsymbol{\theta}})}{L} \right\}' \boldsymbol{\theta}_\ell \right] + \operatorname{RSS}(\hat{\boldsymbol{\theta}}) - \mathbf{r}(\hat{\boldsymbol{\theta}})' \hat{\boldsymbol{\theta}} + \frac{L}{2} \|\hat{\boldsymbol{\theta}}\|^2, \\ \hat{\boldsymbol{\theta}} &= (\hat{\boldsymbol{\theta}}'_1, \dots, \hat{\boldsymbol{\theta}}'_k)' \in \mathbb{R}^{kq}. \end{aligned}$$

Then, RSS and  $\operatorname{RSS}^+$  have the following relationships:

$$\operatorname{RSS}^+(\hat{\boldsymbol{\theta}} \mid \hat{\boldsymbol{\theta}}) = \operatorname{RSS}(\hat{\boldsymbol{\theta}}), \quad \operatorname{RSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}) \geq \operatorname{RSS}(\boldsymbol{\theta}). \quad (2.3)$$

These can be obtained by Taylor expansion of  $\operatorname{RSS}(\boldsymbol{\theta})$  around  $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$  and  $\operatorname{RSS}^+$  is a surrogate

function of RSS.

To solve the optimization problem for HOGL, the essential task is to minimize the following function:

$$f(\gamma) = \frac{1}{2} \|\gamma\|^2 - \mathbf{b}'\gamma + \sum_{j=1}^q \lambda_j \|\gamma_{(j)}\|, \quad \gamma = (\gamma_1, \dots, \gamma_q)', \quad (2.4)$$

where  $\mathbf{b} = (b_1, \dots, b_q)'$  is a  $q$ -dimensional vector of constants,  $\lambda_j$  is a non-negative constant, and  $\gamma_{(j)}$  is a sub-vector of  $\gamma$  given by (1.3). A key to minimizing  $f(\gamma)$  is  $d_{\alpha,j}$  defined by

$$d_{\alpha,j} = \begin{cases} 0 & (j < \alpha) \\ (|b_j| - \lambda_j)_+ & (j = \alpha) \\ \left( \sqrt{d_{\alpha,j-1}^2 + b_j^2} - \lambda_j \right)_+ & (\alpha < j) \end{cases} \quad (\alpha, j \in \{1, \dots, q\}), \quad (2.5)$$

where  $(x)_+ = \max\{x, 0\}$ . Regarding  $d_{\alpha,j}$ , we have the following proposition (the proof is given in Appendix A.1).

**Proposition 1.** *The  $d_{\alpha,j}$  in (2.5) satisfies  $d_{\alpha,j} \geq d_{\alpha+1,j}$ .*

By putting  $d_{\alpha,j}$  into the  $(\alpha, j)$ th element, we have the following upper triangular matrix.

$$\begin{pmatrix} (|b_1| - \lambda_1)_+ & \left( \sqrt{d_{1,1}^2 + b_2^2} - \lambda_2 \right)_+ & \cdots & \left( \sqrt{d_{1,q-1}^2 + b_q^2} - \lambda_q \right)_+ \\ 0 & (|b_2| - \lambda_2)_+ & \ddots & \vdots \\ \vdots & \ddots & \ddots & \left( \sqrt{d_{q-1,q-1}^2 + b_q^2} - \lambda_q \right)_+ \\ 0 & \cdots & 0 & (|b_q| - \lambda_q)_+ \end{pmatrix}. \quad (2.6)$$

Proposition 1 means the elements of the above matrix are monotonically decreasing in each column.

### 3. Main results

The minimizer of  $f(\gamma)$  in (2.4) is given by the following theorem (the proof is given in Appendix A.2).

**Theorem 1.** *Define  $\alpha_*$  as*

$$\alpha_* = \begin{cases} \min \mathcal{A} & (\mathcal{A} \neq \emptyset) \\ q + 1 & (\mathcal{A} = \emptyset) \end{cases}, \quad \mathcal{A} = \left\{ \alpha \in \{1, \dots, q\} \mid \forall j \in \{\alpha, \alpha + 1, \dots, q\}, d_{\alpha,j} > 0 \right\},$$

where  $d_{\alpha,j}$  is given by (2.5). Let  $\gamma^* = (\gamma_1^*, \dots, \gamma_q^*)'$  be the minimizer of  $f(\gamma)$ , i.e.,  $\gamma^* = \arg \min_{\gamma \in \mathbb{R}^q} f(\gamma)$ . Then,  $\gamma_j^*$  is given by

$$\gamma_j^* = \begin{cases} 0 & (j < \alpha_*) \\ b_j \prod_{\ell=j}^q \frac{d_{\alpha_*,\ell}}{d_{\alpha_*,\ell} + \lambda_\ell} & (\alpha_* \leq j) \end{cases}.$$

Applying Theorem 1 to obtain  $\gamma^*$  requires searching for  $\alpha_*$ . We can obtain  $\alpha_*$  directly by searching for a row for which all elements are positive in the upper triangle part of the matrix (2.6). This requires the calculation of  $O(q^2)$ . Fortunately, Proposition 1 makes the search of  $\alpha_*$  efficient. Since Proposition 1,  $d_{\alpha+1,j} = d_{\alpha+2,j} = \dots = d_{q,j} = 0$  holds when  $d_{\alpha,j} = 0$ . This fact provides Algorithm 1. If  $d_{\alpha,j}$  is positive, move to the next column in the same row ( $d_{\alpha,j+1}$ ). If

---

**Algorithm 1** Efficient method for searching  $\alpha_*$

---

```

set  $\alpha \leftarrow 1$ 
for  $j = 1, \dots, q$  do
  calculate  $d_{\alpha,j}$ 
  if  $d_{\alpha,j} = 0$  then
    set  $\alpha \leftarrow j + 1$ 
  end if
end for
define  $\alpha_* = \alpha$ 

```

---

$d_{\alpha,j}$  is zero, since other elements in the same column are all zero, move to the diagonal element in the next column ( $d_{j+1,j+1}$ ). By starting from  $d_{1,1}$  and repeating the above procedure, we can obtain  $\alpha_*$  with the calculation of  $O(q)$ . With Theorem 1 and Algorithm 1, we construct the algorithm to solve the optimization problem for HOGL.

### 3.1. Optimization of hierarchical overlapping group Lasso

To perform the selection of the explanatory variables and the degrees of the polynomial basis functions simultaneously for the GMANOVA model (1.1), we estimate  $\theta$  ( $= \text{vec}(\Theta')$ ) based on minimizing  $\text{PRSS}(\theta \mid \delta, \lambda)$  in (2.2). For this minimization, we construct an algorithm based on the MM philosophy. Since (2.3) and strict convexity of PRSS, Razaviyayn *et al.* (2013) gives the following theorem.

**Theorem 2.** Let  $\theta^{(0)}$  be an initial vector and consider the following update of a solution.

$$\begin{aligned} \theta^{(i)} &= \arg \min_{\theta} \text{PRSS}^+(\theta \mid \theta^{(i-1)}, \delta, \lambda) \quad (i = 1, 2, \dots), \quad (3.1) \\ \text{PRSS}^+(\theta \mid \hat{\theta}, \delta, \lambda) &= \text{RSS}^+(\theta \mid \hat{\theta}) + \lambda \Omega(\Theta \mid \delta). \end{aligned}$$

Then,  $\boldsymbol{\theta}^{(i)}$  converges to the minimizer of  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$ .

From the theorem, we can obtain the minimizer of  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$  by repeatedly solving the minimization problem of  $\text{PRSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}, \delta, \lambda)$ . Notice that  $\text{PRSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}, \delta, \lambda)$  is separable with respect to  $\boldsymbol{\theta}_\ell$  ( $\ell = 1, \dots, k$ ). Hence, the minimization problem of  $\text{PRSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}, \delta, \lambda)$  reduces to that of the following function for each  $\boldsymbol{\theta}_\ell$ :

$$\frac{1}{2}\|\boldsymbol{\theta}_\ell\|^2 - \left\{ \hat{\boldsymbol{\theta}}_\ell - \frac{\mathbf{r}_\ell(\hat{\boldsymbol{\theta}})}{L} \right\}' \boldsymbol{\theta}_\ell + \sum_{j=1}^q \lambda_{\ell,j} \|(\boldsymbol{\theta}_\ell)_{(j)}\|, \quad \lambda_{\ell,j} = \frac{\lambda w_{\ell,j}(\delta)}. \quad (3.2)$$

It is obvious that this function is essentially equal to (2.4). Thus, we can obtain the minimizer in closed form by Theorem 1 and Algorithm 1. This is summarized as follows.

**Corollary 1.** *The minimizer of  $\text{PRSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}, \delta, \lambda)$  is given in closed form following Theorem 1.*

The details of the algorithm for minimizing  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$  are summarized in Algorithm 2.

---

**Algorithm 2** MM algorithm for minimizing  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$

---

**Require:**  $\delta, \lambda$  and an initial vector  $\boldsymbol{\theta}^{(0)}$

set  $i \leftarrow 0$

**repeat**

set  $i \leftarrow i + 1$

calculate  $\boldsymbol{\theta}^{(i)}$  by minimizing  $\text{PRSS}^+(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(i-1)}, \delta, \lambda)$  which is minimized by the following procedure

**for**  $\ell = 1, \dots, k$  **do**

calculate  $\boldsymbol{\theta}_\ell^{(i)}$  by applying Theorem 1 with Algorithm 1 to the minimization of (3.2) with  $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(i-1)}$

**end for**

**until**  $\boldsymbol{\theta}^{(i)}$  converges

---

From Theorem 2 and Corollary 1, it is guaranteed that the solution obtained by Algorithm 2 converges to the optimal solution. Note that although a for loop is used in Algorithm 2 to minimize  $\text{PRSS}^+(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}, \delta, \lambda)$ , this procedure is parallelizable.

In the implementation of HOGL, since PRSS includes two tuning parameters,  $\delta$  and  $\lambda$ , selection of the two parameters is required. The search range of  $\delta$  is  $[0, 1]$ , while that of  $\lambda$  is all positive real values. To limit the search range of  $\lambda$ , it is desirable to find  $\lambda$  such that  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$  is minimized at  $\boldsymbol{\theta} = \mathbf{0}_{kq}$ . A sufficient condition for such  $\lambda$  is given by the following theorem (the proof is given in Appendix A.4).

**Theorem 3.** *Define  $\lambda_{\max}$  as*

$$\lambda_{\max}(\delta) = \max_{\ell \in \{1, \dots, k\}, j \in \{1, \dots, q\}} \frac{|\mathbf{u}' \mathbf{z}_{\ell j}|}{w_{\ell,j}(\delta)},$$

where  $\mathbf{z}_{\ell j} = (a_{1,\ell} \mathbf{v}'_j, \dots, a_{n,\ell} \mathbf{v}'_j)'$ ,  $a_{i,\ell}$  is the  $(i, \ell)$ th element of  $\mathbf{A}$ , and  $\mathbf{v}_j$  is the  $j$ th column vector of  $\mathbf{V}$ . Then, we have

$$\lambda \geq \lambda_{\max}(\delta) \implies \text{PRSS}(\mathbf{0}_{kq} \mid \delta, \lambda) < \text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda) \quad (\forall \boldsymbol{\theta} \in \mathbb{R}^{kq} \setminus \{\mathbf{0}_{kq}\}).$$

From the theorem, we should select the optimal  $\lambda$  in  $[0, \lambda_{\max}]$ .

### 3.2. Transformation of the matrix of basis functions

In the previous section, we proposed HOGL to select the explanatory variables and the degrees of the polynomial basis functions simultaneously. However, we need to consider the potential problems that can be caused by the degrees of the polynomials. When  $q$ , the dimension of the polynomial basis, increases,  $t_j^{q-1}$  diverges to infinity if  $|t_j| > 1$  and  $t_j^{q-1}$  converges to zero if  $|t_j| < 1$ . Hence, fitting high-degree polynomials makes the model unstable and renders model selection difficult. To address this problem, we consider transformation of the matrix of basis functions.

Let  $\mathbf{v}_1, \dots, \mathbf{v}_q$  be column vectors of  $\mathbf{V}$  in (2.1). Gram-Schmidt orthogonalization provides orthonormal basis vectors  $\mathbf{h}_q, \mathbf{h}_{q-1}, \dots, \mathbf{h}_1$  from  $\mathbf{v}_q, \mathbf{v}_{q-1}, \dots, \mathbf{v}_1$  and define  $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_q)$ . Then, we have  $\mathbf{V} = \mathbf{H}\mathbf{Q}$ , where  $\mathbf{Q}$  is a  $q \times q$  lower triangular matrix (the details are given in Appendix A.5). This decomposition implies  $\mathbf{A}\boldsymbol{\Theta}\mathbf{V}' = \mathbf{A}\boldsymbol{\Xi}\mathbf{H}'$  ( $\boldsymbol{\Xi} = \boldsymbol{\Theta}\mathbf{Q}'$ ) and RSS in (2.1) is rewritten as

$$\text{RSS}^\dagger(\boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{u} - \mathbf{Z}^\dagger \boldsymbol{\xi}\|^2,$$

where  $\boldsymbol{\xi} = \text{vec}(\boldsymbol{\Xi}')$  and  $\mathbf{Z}^\dagger = \mathbf{A} \otimes \mathbf{H}$ . Hence, we consider the estimation of  $\boldsymbol{\Xi}$  instead of  $\boldsymbol{\Theta}$ . This transformation means that a model with a polynomial basis is transformed to one with an orthonormal basis. Then, to select degrees of polynomial basis functions in the original model, we need to select the number of orthonormal basis vectors. Fortunately, HOGL can be applied to the selection of the orthonormal basis vectors, and hence we can estimate  $\boldsymbol{\Xi}$  by minimizing the following PRSS:

$$\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda) = \text{RSS}^\dagger(\boldsymbol{\xi}) + \lambda \sum_{\ell=1}^k \sum_{j=1}^q w_{\ell,j}(\delta) \|(\boldsymbol{\xi}_\ell)_{(j)}\|, \quad \boldsymbol{\Xi} = (\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_k)'$$

By applying HOGL for  $\boldsymbol{\xi}_\ell$ , we can perform the selections of the explanatory variables and the number of orthonormal basis vectors simultaneously. Assume that we have the estimator  $\hat{\boldsymbol{\Xi}}$  of  $\boldsymbol{\Xi}$  by minimizing  $\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda)$ . Then, the estimator of  $\boldsymbol{\Theta}$  is given by  $\hat{\boldsymbol{\Theta}} = \hat{\boldsymbol{\Xi}}(\mathbf{Q}')^{-1}$  and the hierarchical structure in  $\hat{\boldsymbol{\Xi}}$  is inherited by  $\hat{\boldsymbol{\Theta}}$  because  $(\mathbf{Q}')^{-1}$  is an upper triangular matrix. Hence, we can select degrees of the original polynomials via the matrix of orthonormal basis.

Although Algorithm 2 can be applied to the minimization of  $\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda)$  because it is

essentially equal to  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$ , we can directly apply the block-wise coordinate descent method to the minimization problem by the orthogonality of  $\mathbf{H}$ . We divide  $\mathbf{Z}^\dagger$  into the blocks as

$$\mathbf{Z}^\dagger = (\mathbf{Z}_1^\dagger, \dots, \mathbf{Z}_k^\dagger), \quad \mathbf{Z}_\ell^\dagger = \mathbf{a}_\ell \otimes \mathbf{H},$$

where  $\mathbf{a}_\ell$  is the  $\ell$ th column vector of  $\mathbf{A}$ . Since  $\|\mathbf{a}_\ell\| = 1$  and  $\mathbf{H}'\mathbf{H} = \mathbf{I}_q$ , we can separate  $\text{RSS}^\dagger(\boldsymbol{\xi})$  with respect to  $\boldsymbol{\xi}_\ell$  as

$$\begin{aligned} \text{RSS}^\dagger(\boldsymbol{\xi}) &= \frac{1}{2} \|\tilde{\mathbf{u}}_\ell - \mathbf{Z}_\ell^\dagger \boldsymbol{\xi}_\ell\|^2 = \frac{1}{2} \|\boldsymbol{\xi}_\ell\|^2 - \tilde{\mathbf{u}}_\ell' \mathbf{Z}_\ell^\dagger \boldsymbol{\xi}_\ell + \frac{1}{2} \|\tilde{\mathbf{u}}_\ell\|^2, \\ \tilde{\mathbf{u}}_\ell &= \mathbf{u} - \sum_{j \neq \ell} \mathbf{Z}_j^\dagger \boldsymbol{\xi}_j. \end{aligned}$$

Hence,  $\boldsymbol{\xi}_\ell$  is updated by minimizing the following function:

$$\frac{1}{2} \|\boldsymbol{\xi}_\ell\|^2 - \tilde{\mathbf{u}}_\ell' \mathbf{Z}_\ell^\dagger \boldsymbol{\xi}_\ell + \sum_{j=1}^q \lambda_{\ell,j}^\dagger \|(\boldsymbol{\xi}_\ell)_{(j)}\|, \quad \lambda_{\ell,j}^\dagger = \lambda w_{\ell,j}(\delta). \quad (3.3)$$

This function is essentially equal to (2.4), and hence its minimizer is obtained in closed form by Theorem 1. The details of the algorithm for minimizing  $\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda)$  are summarized in Algorithm 3. Notice that the penalty term of  $\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda)$  is separable with respect to

---

**Algorithm 3** Blockwise coordinate descent method for minimizing  $\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda)$

---

**Require:**  $\delta, \lambda$  and an initial vector  $\boldsymbol{\xi}^{(0)}$

set  $i \leftarrow 0$

**repeat**

set  $i \leftarrow i + 1$

calculate  $\boldsymbol{\xi}^{(i)} = (\boldsymbol{\xi}_1^{(i)'}, \dots, \boldsymbol{\xi}_k^{(i)'})'$  by the following procedure

**for**  $\ell = 1, \dots, k$  **do**

calculate  $\boldsymbol{\xi}_\ell^{(i)}$  by applying Theorem 1 with Algorithm 1 to the minimization of (3.3) with  $\boldsymbol{\xi}_j = \boldsymbol{\xi}_j^{(i)}$  ( $j < \ell$ ) and

$\boldsymbol{\xi}_j = \boldsymbol{\xi}_j^{(i-1)}$  ( $j > \ell$ )

**end for**

**until**  $\boldsymbol{\xi}^{(i)}$  converges

---

$\boldsymbol{\xi}_\ell$  and (3.3) is strictly convex. Hence, Tseng (2001) guarantees that the solution obtained by Algorithm 3 converges to the optimal solution. Furthermore, similar to Theorem 3,  $\lambda_{\max}$  for  $\text{PRSS}^\dagger(\boldsymbol{\xi} \mid \delta, \lambda)$  is given by

$$\lambda_{\max}(\delta) = \max_{\ell \in \{1, \dots, k\}, j \in \{1, \dots, q\}} \frac{|\mathbf{u}' \mathbf{z}_{\ell j}^\dagger|}{w_{\ell,j}(\delta)},$$

where  $\mathbf{z}_{\ell j}^\dagger = (a_{1,\ell} \mathbf{h}'_j, \dots, a_{n,\ell} \mathbf{h}'_j)'$ .

### 3.3. Extended Hierarchical Overlapping Group Lasso

Thus far, we have discussed hierarchical overlapping by adding parameters one by one to select the degrees of the polynomials. Here, we extend HOGL to flexible hierarchical overlapping. As an example, consider using the following Fourier basis instead of the polynomial basis (e.g., von Rosen, 2018, Chap. 1):

$$\boldsymbol{x}(t) = (\cos(q-1)t, \sin(q-1)t, \cos(q-2)t, \sin(q-2)t, \dots, \cos t, \sin t, 1)'$$

For simplicity, the constants used to standardize the column vectors of  $\boldsymbol{X}$  are omitted from the expression. Then, the varying coefficients in (1.2) are given by

$$\beta_\ell(t) = \sum_{j=1}^{q-1} (\cos(q-j)t, \sin(q-j)t) \boldsymbol{\theta}_{\ell,j} + \theta_{\ell,q},$$

where  $\boldsymbol{\theta}_{\ell,j}$  is a two-dimensional vector. To select a pair of basis functions  $(\cos(q-j)t, \sin(q-j)t)$ , we need HOGL with hierarchical overlapping by adding parameters two at a time. To address the selection of basis functions with group structure, we extend HOGL.

Here, we divide the parameter vector  $\boldsymbol{\theta}_\ell$  ( $\ell \in \{1, \dots, k\}$ ) as

$$\boldsymbol{\theta}_\ell = (\boldsymbol{\theta}'_{\ell,1}, \dots, \boldsymbol{\theta}'_{\ell,q})',$$

where  $\boldsymbol{\theta}_{\ell,j}$  is an  $m_j$ -dimensional vector and  $\boldsymbol{\theta}_\ell$  is an  $m = \sum_{j=1}^q m_j$ -dimensional vector. Furthermore, for  $m$ -dimensional block vector  $\boldsymbol{\gamma} = (\boldsymbol{\gamma}'_1, \dots, \boldsymbol{\gamma}'_q)'$ , we extend the sub-vector in (1.3) to

$$\boldsymbol{\gamma}_{(j)} = (\boldsymbol{\gamma}'_1, \dots, \boldsymbol{\gamma}'_j)', \quad (3.4)$$

where  $\boldsymbol{\gamma}_j$  is an  $m_j$ -dimensional vector. Since this notation naturally extends the HOGL penalty in (1.4), we call the approach extended HOGL (EHOGL). When  $m_1 = \dots = m_q = 1$ , EHOGL reduces to HOGL. To solve the optimization problem for EHOGL, the essential task is to minimize the following function:

$$f(\boldsymbol{\gamma}) = \frac{1}{2} \|\boldsymbol{\gamma}\|^2 - \boldsymbol{b}'\boldsymbol{\gamma} + \sum_{j=1}^q \lambda_j \|\boldsymbol{\gamma}_{(j)}\|, \quad \boldsymbol{\gamma} = (\boldsymbol{\gamma}'_1, \dots, \boldsymbol{\gamma}'_q)', \quad (3.5)$$

where  $\boldsymbol{b} = (\boldsymbol{b}'_1, \dots, \boldsymbol{b}'_q)'$  is an  $m$ -dimensional block vector and  $\boldsymbol{b}_j$  is an  $m_j$ -dimensional vector of constants. To minimize  $f(\boldsymbol{\gamma})$ , we extend  $d_{\alpha,j}$  in (2.5) to

$$d_{\alpha,j} = \begin{cases} 0 & (j < \alpha) \\ (\|\boldsymbol{b}_j\| - \lambda_j)_+ & (j = \alpha) \\ \left( \sqrt{d_{\alpha,j-1}^2 + \|\boldsymbol{b}_j\|^2} - \lambda_j \right)_+ & (\alpha < j) \end{cases} \quad (\alpha, j \in \{1, \dots, q\}). \quad (3.6)$$

Then, the minimizer of  $f(\gamma)$  is given by the following theorem (the proof is given in Appendix A.2).

**Theorem 4.** Define  $\alpha_*$  as

$$\alpha_* = \begin{cases} \min \mathcal{A} & (\mathcal{A} \neq \emptyset) \\ q+1 & (\mathcal{A} = \emptyset) \end{cases}, \quad \mathcal{A} = \{\alpha \in \{1, \dots, q\} \mid \forall j \in \{\alpha, \alpha+1, \dots, q\}, d_{\alpha,j} > 0\}.$$

Let  $\gamma^* = (\gamma_1^{*'}, \dots, \gamma_q^{*'})'$  be the minimizer of  $f(\gamma)$  in (3.5), i.e.,  $\gamma^* = \arg \min_{\gamma \in \mathbb{R}^m} f(\gamma)$ . Then,  $\gamma_j^*$  is given by

$$\gamma_j^* = \begin{cases} \mathbf{0}_{m_j} & (j < \alpha_*) \\ \prod_{\ell=j}^q \frac{d_{\alpha_*,\ell}}{d_{\alpha_*,\ell} + \lambda_\ell} \mathbf{b}_j & (\alpha_* \leq j) \end{cases}.$$

Similar to Proposition 1,  $d_{\alpha,j}$  in (3.6) satisfies  $d_{\alpha,j} \geq d_{\alpha+1,j}$ . Hence,  $\alpha_*$  in Theorem 4 can be obtained by Algorithm 1.

#### 4. Numerical study

In this section, we evaluate the performance of the proposed HOGL using Monte Carlo simulation with 1000 iterations, and compare the performance of HOGL with that of wSCAD and a third method using sparse group Lasso. The numerical calculation programs are executed in R (ver. 4.5.0; R Core Team, 2025) on a computer running the Windows 11 Pro operating system with an AMD EPYC TM 7763 processor and 128 GB of RAM. HOGL is available via R package HOGLgmanova (Ohishi, 2025).

We first describe the setting of the simulation. The simulation model is defined by

$$\begin{aligned} \mathbf{Y} &\sim N_{n \times p}(\mathbf{A}\Theta\mathbf{X}', \Sigma \otimes \mathbf{I}_n), \\ \mathbf{A} &= \mathbf{A}_0\Psi^{1/2}, \quad \Psi = \mathbf{R}_k^{1/2}\Omega_k(0.5)\mathbf{R}_k^{1/2}, \quad \Sigma = \mathbf{R}_p^{1/2}\Omega_p(0.5)\mathbf{R}_p^{1/2}, \end{aligned}$$

where  $\mathbf{A}_0$  is an  $n \times k$  matrix with elements identically and independently distributed according to  $U(-1, 1)$ ,  $\mathbf{R}_k = \text{diag}(1, \dots, k)$ , and  $\Omega_k(\rho)$  is a  $k \times k$  autocorrelation matrix with the  $(i, j)$ th element  $\rho^{|i-j|}$ . Furthermore,  $\mathbf{X}$  is a  $p \times q$  matrix of polynomial basis functions of degree  $(q-1)$  and the  $j$  ( $\in \{1, \dots, p\}$ )th time point is given by  $t_j = 2(j-1)/(p-1) - 1$ , where  $t_j$  is the  $j$ th of the points obtained by uniformly dividing  $[-1, 1]$ . The  $\Theta$  is defined by

$$\Theta = \begin{pmatrix} \mathbf{O}_{k_*,q-6} & \Theta_* \\ \mathbf{O}_{k-k_*,q-6} & \mathbf{O}_{k-k_*,6} \end{pmatrix}, \quad \Theta_* = \nu \begin{pmatrix} 0 & 0 & 0 & 0 & -3 & 0.5 \\ 0 & 0 & 0 & 4 & 1 & -2 \\ 0 & 0 & 6 & -2 & -4 & 2 \\ 0 & 12 & 3 & -12 & -3 & 1.5 \\ -12 & -1 & 15 & 1 & -1 & -0.5 \end{pmatrix},$$

where  $\nu$  is a parameter adjusting the signal-to-noise ratio (SNR). The number of true explanatory variables is  $k_* = 5$ , and the true degree of varying coefficient  $\beta_\ell(t)$  ( $\ell \in \{1, \dots, k_*\}$ ) in (1.2) is  $\ell$ . Figure 1 shows the shapes of  $\beta_\ell(t)$ . Furthermore, SNR is defined by

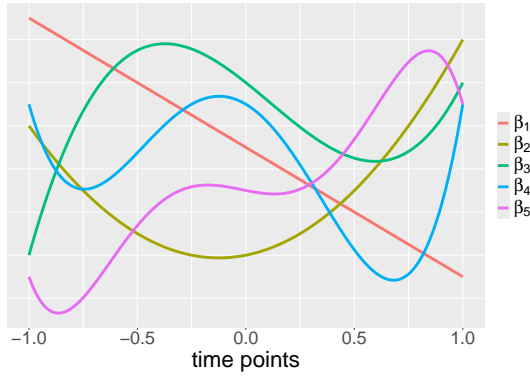


Figure 1. Shapes of varying coefficients  $\beta_\ell(t)$

$$\text{SNR} = \frac{1}{p} \sum_{j=1}^p \frac{\text{Var}[\mathbf{a}'\Theta\mathbf{x}_j]}{\sigma_j^2} = \frac{1}{3p} \sum_{j=1}^p \frac{\mathbf{x}'_j\Theta'\Psi\Theta\mathbf{x}_j}{\sigma_j^2},$$

and  $\nu$  is defined such that  $\text{SNR} = 1$ , where  $\mathbf{a} = \Psi^{1/2}\mathbf{a}_0$ ,  $\mathbf{a}_0$  is a  $k$ -dimensional vector with elements identically and independently distributed according to  $U(-1, 1)$ ,  $\mathbf{x}_j$  is the  $j$ th row vector of  $\mathbf{X}$  and  $\sigma_j^2$  is the  $j$ th diagonal element.

Under the setting described above, we determine the selection probabilities of the true variables and degrees, as well as the mean squared errors (MSEs) and the runtime. Selection probabilities are based on the percentage (%) of the time that the true variables or degrees are selected over 1000 iterations. MSEs are defined for the matrices of the fitted values  $\hat{\mathbf{Y}}$  and the estimators of the regression coefficients  $\hat{\Theta}$ , as shown below:

$$\text{MSE}_f[\hat{\mathbf{Y}}] = \frac{1}{np} \text{E} \left[ \text{tr} \left\{ (\hat{\mathbf{Y}} - \mathbf{A}\Theta\mathbf{X}')' (\hat{\mathbf{Y}} - \mathbf{A}\Theta\mathbf{X}') \Sigma^{-1} \right\} \right],$$

$$\text{MSE}_c[\hat{\Theta}] = \frac{1}{kq} \text{E} \left[ \text{tr} \left\{ (\hat{\Theta} - \Theta)' (\hat{\Theta} - \Theta) \right\} \right],$$

where expectation is evaluated by Monte Carlo simulation with 1000 iterations. The methods used in this simulation are as follows.

- HOGL1: proposed method with  $V$  using Algorithm 2.
- HOGL2: proposed method with  $H$  using Algorithm 3, where  $H$  is obtained by transforming  $V$  as described in Section 3.2.
- SGL: penalized estimation method with sparse group Lasso penalty instead of HOGL penalty in (1.4).
- wSCAD: method proposed by Hu *et al.* (2014).

Based on adaptive Lasso (Zou, 2006), penalty weights in HOGL penalty are defined by

$$w_{\ell,j}^{(0)} = \begin{cases} \left\| (\tilde{\theta}_\ell)_{(j)} \right\|^{-1} & \text{(HOGL1)} \\ \left\| (\tilde{\xi}_\ell)_{(j)} \right\|^{-1} & \text{(HOGL2)} \end{cases}, \quad \tilde{\theta}_\ell = \tilde{\Theta}' e_\ell, \quad \tilde{\xi}_\ell = \tilde{\Xi}' e_\ell,$$

where  $\tilde{\Theta}$  and  $\tilde{\Xi}$  are the ordinary least squares estimators of  $\Theta$  and  $\Xi$ , respectively, defined by

$$\tilde{\Theta} = (A'A)^{-1} A'UV(V'V)^{-1}, \quad \tilde{\Xi} = \tilde{\Theta}Q' = (A'A)^{-1} A'UH,$$

and  $e_\ell$  is a  $k$ -dimensional unit vector with the  $\ell$ th element one. Regarding the two tuning parameters  $\delta$  and  $\lambda$ , we set 10 and 100 candidates for  $\delta$  and  $\lambda$ , respectively, and select the best pair using a grid search based on minimizing the extended GCV (EGCV) criterion (Ohishi *et al.*, 2020) defined by

$$\text{EGCV}(\delta, \lambda \mid \alpha) = \frac{\text{tr}\{(\mathbf{Y} - \hat{\mathbf{Y}}_{\delta,\lambda})'(\mathbf{Y} - \hat{\mathbf{Y}}_{\delta,\lambda})\mathbf{S}^{-1}\}}{(1 - \text{df}_{\delta,\lambda}/np)^\alpha},$$

where  $\hat{\mathbf{Y}}_{\delta,\lambda}$  is a matrix of fitted values under given  $\delta$  and  $\lambda$ ,  $\text{df}_{\delta,\lambda}$  is the number of corresponding non-zero parameters, and we set  $\alpha = \log(np)$ . Similar to HOGL, we incorporate penalty weights into SGL and select two tuning parameters. Regarding wSCAD, we define 10 candidates for each of the three tuning parameters and select the best pair by grid search based on minimizing the BIC as proposed by Hu *et al.* (2014). Then, wSCAD is guaranteed to have the oracle properties under a large sample asymptotic framework (i.e., only  $n$  diverges to infinity). Note that  $\alpha = \log(np)$  in the EGCV criterion corresponds to the BIC in Hu *et al.* (2014). This simulation considers the following four cases as  $n$  increases: (1)  $p = k = 10$ , (2)  $p/n = 0.4$ ,  $k = 10$ , (3)  $p = 10$ ,  $k/n = 0.4$ , and (4)  $p/n = 0.2$ ,  $k/n = 0.4$ .

Table 1 summarizes the selection probabilities of the true variables and degrees when  $q = 6$ , i.e., fitting a polynomial of degree five. We can see that HOGL2 performed very well in both

Table 1. Selection probability (%) when  $q = 6$

$n$	$p$	$k$	Variable				Degree			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	14.4	<b>79.2</b>	15.4	2.8	4.5	<b>51.1</b>	3.2	0.0
300	10	10	36.9	<b>93.0</b>	31.2	23.2	17.5	<b>78.8</b>	7.0	12.6
500	10	10	54.1	<b>97.3</b>	39.6	81.8	25.7	<b>88.0</b>	9.8	68.4
100	40	10	2.8	<b>58.6</b>	2.8	2.4	1.9	<b>37.1</b>	1.0	0.0
300	120	10	35.0	<b>86.9</b>	32.7	49.1	16.5	<b>73.0</b>	3.3	36.8
500	200	10	61.4	<b>92.5</b>	58.4	90.8	26.2	<b>80.3</b>	5.2	79.1
100	10	40	8.4	<b>60.3</b>	2.1	0.0	0.9	<b>42.7</b>	3.7	0.0
300	10	120	21.5	<b>89.8</b>	9.0	0.0	3.7	<b>70.6</b>	9.0	0.0
500	10	200	27.9	<b>94.3</b>	18.3	0.4	5.5	<b>82.6</b>	7.5	27.4
100	20	40	3.7	<b>47.8</b>	0.7	0.0	1.3	<b>39.3</b>	1.8	0.0
300	60	120	11.7	<b>87.5</b>	4.3	2.2	2.6	<b>71.3</b>	6.5	0.3
500	100	200	28.8	<b>90.9</b>	10.9	78.1	5.8	84.3	5.2	<b>86.9</b>

variable selection and degree selection. Although wSCAD also performed well, the performance declined when  $k$  increased. It can be considered that the theoretical properties of wSCAD appear in the results. The performances of HOGL1 and SGL were not good; in particular, degree selection was quite poor.

Table 2. MSE when  $q = 6$

$n$	$p$	$k$	MSE <sub>f</sub>				MSE <sub>c</sub>			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	0.070	<b>0.043</b>	0.071	0.125	0.625	<b>0.253</b>	0.548	1.210
300	10	10	0.020	<b>0.012</b>	0.021	0.044	0.153	<b>0.059</b>	0.176	0.368
500	10	10	0.011	<b>0.007</b>	0.013	0.029	0.074	<b>0.030</b>	0.108	0.231
100	40	10	0.028	0.022	0.031	<b>0.022</b>	1.738	<b>0.840</b>	2.720	3.652
300	120	10	0.005	0.004	0.005	<b>0.003</b>	0.492	<b>0.219</b>	1.097	1.167
500	200	10	0.003	0.002	0.003	<b>0.001</b>	0.285	<b>0.124</b>	0.707	0.344
100	10	40	0.291	<b>0.063</b>	0.154	0.616	0.634	<b>0.099</b>	0.232	0.523
300	10	120	0.102	<b>0.015</b>	0.075	0.270	0.087	<b>0.007</b>	0.035	0.042
500	10	200	0.063	<b>0.008</b>	0.052	0.118	0.030	<b>0.002</b>	0.015	0.017
100	20	40	0.159	<b>0.041</b>	0.089	0.299	0.995	<b>0.184</b>	0.384	0.708
300	60	120	0.026	<b>0.006</b>	0.020	0.123	0.227	<b>0.020</b>	0.092	0.138
500	100	200	0.009	<b>0.003</b>	0.008	0.012	0.062	<b>0.006</b>	0.041	0.048

Table 2 summarizes the MSEs of the fitted values and coefficients when  $q = 6$ . Regarding the MSE of the coefficients, HOGL2 was always superior. We can consider that this was caused by the high performances of variable selection and degree selection. HOGL2 also performed well with respect to the MSE of the fitted values. However, in the case in which  $n$  and  $p$  increase and  $k$  is fixed, wSCAD performed best, although differences among the four methods became small. It can be guessed that this setting represents a desirable situation in which sample size and number of time points are sufficiently large for the number of explanatory variables and the dimension of the polynomial basis. In addition, since the tuning parameters are selected according to prediction accuracy, the MSE of the fitted values may be good even when the performances of variable selection and degree selection are poor.

Table 3. Selection probability (%) when  $q = 10$

$n$	$p$	$k$	Variable				Degree			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	0.9	<b>74.8</b>	20.7	0.8	0.0	<b>31.6</b>	0.0	0.0
300	10	10	1.1	<b>91.6</b>	24.7	0.4	0.0	<b>65.1</b>	0.0	0.0
500	10	10	0.4	<b>95.8</b>	28.5	0.3	0.0	<b>80.2</b>	0.0	0.0
100	40	10	0.9	<b>53.2</b>	16.5	1.0	0.0	<b>19.7</b>	0.0	0.0
300	120	10	0.1	<b>87.0</b>	42.1	1.0	0.0	<b>61.1</b>	0.0	0.0
500	200	10	0.0	<b>92.4</b>	50.6	1.3	0.0	<b>76.1</b>	0.0	0.1
100	10	40	0.0	<b>52.4</b>	5.1	0.0	0.0	<b>20.6</b>	0.0	0.0
300	10	120	0.2	<b>85.7</b>	13.3	0.0	0.0	<b>49.6</b>	0.0	0.0
500	10	200	0.1	<b>89.8</b>	11.8	0.0	0.0	<b>67.2</b>	0.0	0.0
100	20	40	0.1	<b>41.3</b>	4.6	0.0	0.0	<b>20.6</b>	0.0	0.0
300	60	120	0.0	<b>83.5</b>	9.7	0.0	0.0	<b>53.7</b>	0.0	0.0
500	100	200	0.1	<b>89.4</b>	13.0	4.5	0.0	<b>74.3</b>	0.0	0.0

Table 3 summarizes the probabilities of selecting the true variables and degrees when  $q = 10$ , i.e., fitting a polynomial of degree nine. Comparing results with the case in which  $q = 6$ , HOGL2 maintained its high performance; however, the performances of the other methods declined. It can be considered that fitting a high-degree polynomial rendered the model unstable, which made identification of the basis vectors difficult. Figure 2 shows the basis functions when  $n = 500$ ,  $p = 10$ ,  $k = 10$ , and  $q = 10$ ; the left and right panels show the column vectors of  $\mathbf{V}$  and  $\mathbf{H}$ , respectively. From the figure, we can see that although identification of the basis functions of the original polynomial basis is difficult, basis function differences became very clear when transforming the polynomial basis to an orthonormal basis. Hence, we can

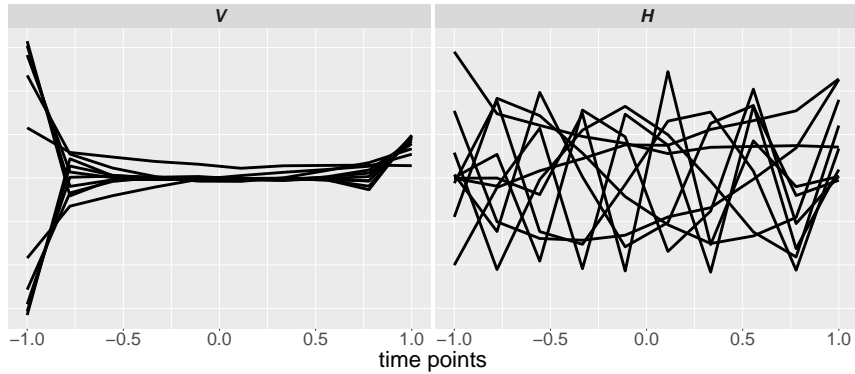


Figure 2. Standardized polynomial basis functions (left) and transformed basis functions (right)

conclude that the performances of HOGL1, SGL, and wSCAD, which use  $V$ , declined, while HOGL2, which uses  $H$ , was not affected by high-degree polynomials.

Table 4. MSE when  $q = 10$

$n$	$p$	$k$	MSE <sub>f</sub>				MSE <sub>c</sub>			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	0.334	<b>0.049</b>	0.216	0.670	2616.288	<b>0.857</b>	3518.354	884.144
300	10	10	0.043	<b>0.013</b>	0.053	0.071	1245.160	<b>0.056</b>	1142.321	1267.022
500	10	10	0.026	<b>0.007</b>	0.030	0.036	733.753	<b>0.027</b>	672.788	753.098
100	40	10	0.050	<b>0.024</b>	0.049	0.076	1696.813	<b>2.101</b>	1569.357	1271.156
300	120	10	0.008	<b>0.004</b>	0.007	0.005	858.507	<b>0.425</b>	743.454	646.727
500	200	10	0.004	0.002	0.003	<b>0.002</b>	576.993	<b>0.134</b>	489.744	431.723
100	10	40	0.987	<b>0.074</b>	0.511	0.790	34.732	<b>0.106</b>	737.015	0.703
300	10	120	0.734	<b>0.018</b>	0.366	0.774	1.505	<b>0.008</b>	106.320	2.961
500	10	200	0.627	<b>0.009</b>	0.306	0.743	0.734	<b>0.002</b>	41.426	2.731
100	20	40	0.397	<b>0.047</b>	0.274	0.523	5.060	<b>0.164</b>	163.205	12.746
300	60	120	0.126	<b>0.006</b>	0.061	0.365	2.409	<b>0.021</b>	58.196	10.226
500	100	200	0.076	<b>0.003</b>	0.033	0.242	1.322	<b>0.005</b>	27.600	10.187

Table 4 summarizes the MSEs for the fitted values and coefficients when  $q = 10$ . As can be seen here, the MSE values for the coefficients clearly reflect the performances of variable and degree selections, while the MSEs for the fitted values are barely affected. Table 5 and Figure 3 show one example of estimation results when  $n = 500$ ,  $p = 200$ ,  $k = 10$ , and  $q = 10$ , where the estimation errors of the matrices of fitted values  $\hat{Y}$  and the estimates of regression

Table 5. Example of estimation results

	Estimation error		Estimated degrees									
	Fitted values	Coefficients	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$\beta_6$	$\beta_7$	$\beta_8$	$\beta_9$	$\beta_{10}$
HOGL1	0.003	580.863	9	9	9	9	9	0	9	9	9	9
HOGL2	0.002	0.050	1	2	3	4	5	0	0	0	0	0
SGL	0.003	398.871	9	9	9	9	9	0	0	0	0	0
wSCAD	0.001	359.984	1	9	9	9	9	0	8	9	9	0

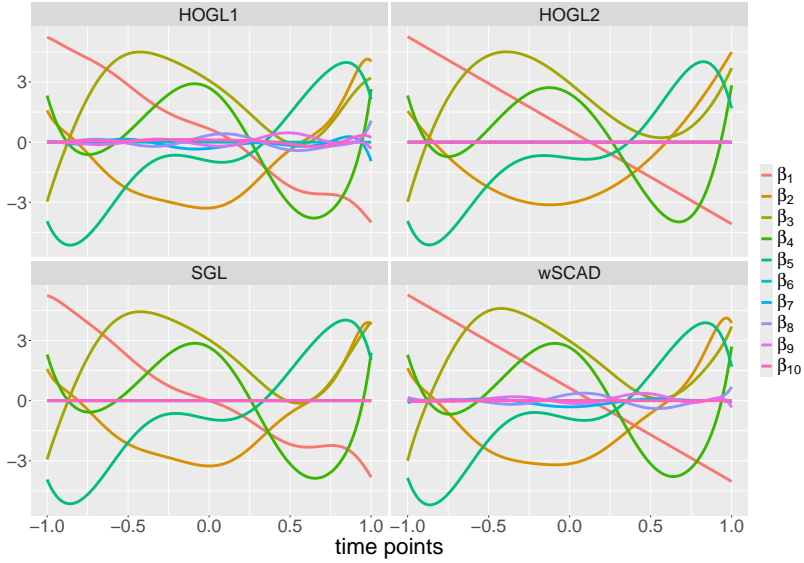


Figure 3. Example of estimated varying coefficients

coefficients  $\hat{\Theta}$  are given by  $\text{tr}\{(\hat{Y} - \mathbf{A}\Theta X')'(\hat{Y} - \mathbf{A}\Theta X')\Sigma^{-1}\}/np$  and  $\text{tr}\{(\hat{\Theta} - \Theta)'(\hat{\Theta} - \Theta)\}/kq$ , respectively, and  $\beta_\ell$  is given by (1.2). In this example, only HOGL2 was able to perfectly select the true variables and degrees, and the estimation error for the coefficients was very small. On the other hand, since the other methods were unable to make similarly accurate selections, the estimation errors of the coefficients were much larger. However, the differences among the estimation errors for the fitted values were relatively small, as can be seen in Figure 3. The figure shows that the shapes of the varying coefficients do not exhibit large differences even when the true degrees were not selected, which would explain why the incorrect selection of the true degrees did not materially affect the estimation error of the fitted values. Nevertheless, correct selection of the true degrees is important to properly and concisely interpret a model.

Table 6 summarizes runtime. The table shows that HOGL2 has advantages not only in terms

Table 6. Runtime (sec.)

$n$	$p$	$k$	$q = 6$				$q = 10$			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	135.6	<b>5.0</b>	10.5	15.8	127.0	7.5	<b>6.8</b>	18.3
300	10	10	127.5	<b>4.6</b>	10.1	15.1	140.6	<b>6.8</b>	7.4	18.1
500	10	10	116.7	<b>4.4</b>	10.5	14.6	150.4	<b>6.6</b>	7.9	18.5
100	40	10	148.2	<b>5.3</b>	9.5	14.4	165.8	7.3	<b>6.3</b>	18.6
300	120	10	119.7	<b>5.0</b>	16.7	14.5	171.5	<b>6.8</b>	24.1	19.6
500	200	10	95.4	<b>6.3</b>	43.1	17.1	183.9	<b>7.6</b>	66.9	20.5
100	10	40	1272.5	<b>35.5</b>	137.6	67.9	1638.8	<b>56.2</b>	76.6	87.6
300	10	120	5485.4	<b>102.8</b>	909.2	405.1	15193.4	<b>165.7</b>	662.8	1064.8
500	10	200	8617.4	<b>166.6</b>	2656.5	1339.6	44570.6	<b>273.2</b>	2064.3	4726.5
100	20	40	1325.8	<b>36.3</b>	97.0	60.4	1996.5	53.8	<b>40.5</b>	79.3
300	60	120	5413.2	<b>85.0</b>	946.5	361.5	17647.3	<b>130.9</b>	530.2	917.0
500	100	200	5899.0	<b>121.4</b>	3610.8	1212.9	49388.0	<b>180.8</b>	2715.4	4149.9

of model selection and MSE but also in calculation speed. In contrast, HOGL1 is particularly slow. Thus, we can say that transformation of the matrix of basis functions affects not only the performance of model selection but also the calculation speed.

## 5. Concluding remarks

This paper proposed hierarchical overlapping group Lasso (HOGL) to perform the selections of the explanatory variables and the degrees of the polynomial basis functions simultaneously in the GMANOVA model. By hierarchically applying group Lasso to coefficients, beginning with higher-order coefficients, HOGL can uniformly address both selection problems. Algorithms with optimality and convergence that can be used to solve the optimization problem for HOGL were also proposed, based on the MM philosophy and the block-wise coordinate descent method in which the update equation of a solution is given in closed form. In a numerical simulation comparing the proposed method with several existing methods, HOGL was shown to have advantages in terms of both variable section and degree selection, as well as MSE and calculation speed. Notably, HOGL was able to maintain high performance even when fitting high-degree polynomials by transforming the matrix of the polynomial basis functions. It should be noted, however, that all the methods examined have the potential for improvement.

While wSCAD is guaranteed to have oracle properties under specific conditions, appropriate candidates and search ranges for its three tuning parameters are not provided. In the simulation,

we selected the three tuning parameters from 1000 sets of candidates. Although we would expect improvements in performance if more candidates are included, runtime requirements may render this impractical. The two tuning parameters of HOGL were selected from 1000 pairs of candidates. Although HOGL showed good numerical performance by selecting the optimal pair of tuning parameters using the EGCV criterion with  $\alpha = \log(np)$ , theoretical reasons were not offered. The two tuning parameters of SGL were also selected from 1000 pairs of candidates using the EGCV criterion with  $\alpha = \log(np)$ . However, the choice of  $\alpha$  produced a large difference in performance, particularly for SGL. The choice of  $\alpha$  involved a clear trade-off between model selection performance and MSE. For example,  $\alpha = \sqrt{np}$  improved model selection but worsened MSE, while  $\alpha = \log(np)$  was good for MSE.

Moreover, the true model in the simulation depended on SNR. SNR can be interpreted as the difficulty of model selection: a larger SNR value implies that model selection is easier. In fact, we found that the performances of all the methods improved and that the differences among the methods became small by setting  $\text{SNR} = 3$  (the results are given in Appendix A.6). At the same time, it is a great advantage that HOGL performed well in the setting where  $\text{SNR} = 1$ , which corresponds to the more difficult setting. As future work, we plan to investigate theoretical characteristics such as oracle properties and develop methods using other penalty versions (e.g., SCAD) of HOGL.

**Acknowledgment** The authors thank FORTE Science Communications (<https://www.forte-science.co.jp/>) for English language editing. This work was partially supported by JSPS KAKENHI Grant Numbers 23H00809, 25K17296, and 25K21159.

## References

- Enomoto, R., Sakurai, T. & Fujikoshi, Y. (2015). Consistency properties of AIC, BIC,  $C_p$  and their modifications in the growth curve model under a large- $(q, n)$  framework. *SUT J. Math.*, **51**, 59–81.
- Fan, J. & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, **96**, 1348–1360.
- Fujikoshi, Y. & Rao, C. R. (1991). Selection of covariables in the growth curve model. *Biometrika*, **78**, 779–785.
- Hu, J., Xin, X. & You, J. (2014). Model determination and estimation for the growth curve model via group SCAD penalty. *J. Multivariate Anal.*, **124**, 199–213.
- Hunter, D. R. & Lange, K. (2004). A tutorial on MM algorithms. *Amer. Statist.*, **58**, 30–37.

- Kshirsagar, A. M. & Smith, W. B. (1995). *Growth Curves*. Marcel Dekker, Inc. New York, Basel, Hong Kong.
- Obozinski, G. R., Wainwright, M. J. & Jordan, M. (2008). High-dimensional support union recovery in multivariate regression. D. Koller, D. Schuurmans, Y. Bengio & L. Bottou, eds, *Advances in Neural Information Processing Systems*. Vol. 21. Curran Associates, Inc.
- Ohishi, M. (2025). *HOGlgmanova: Variable and basis function selections in GMANOVA model*. R package version 0.1.0. **URL:** <https://github.com/ohishim/HOGlgmanova>
- Ohishi, M., Yanagihara, H. & Fujikoshi, Y. (2020). A fast algorithm for optimizing ridge parameters in a generalized ridge regression by minimizing a model selection criterion. *J. Statist. Plann. Inference*, **204**, 187–205.
- Potthoff, R. F. & Roy, S. N. (1964). A generalized multivariate analysis of variance model useful especially for growth curve problems. *Biometrika*, **51**, 313–325.
- Razaviyayn, M., Hong, M. & Luo, Z.-Q. (2013). A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM J. Optim.*, **23**, 1126–1153.
- Satoh, K., Kobayashi, M. & Fujikoshi, Y. (1997). Variable selection for the growth curve model. *J. Multivariate Anal.*, **60**, 277–292.
- Satoh, K. & Yanagihara, H. (2010). Estimation of varying coefficients for a growth curve model. *Amer. J. Math. Management Sci.*, **30**, 243–256.
- Simon, N., Friedman, J., Hastie, T. & Tibshirani, R. (2013). A sparse-group Lasso. *J. Comput. Graph. Statist.*, **22**, 231–245.
- Srivastava, M. S. (2002). *Methods of Multivariate Statistics*. John Wiley & Sons, Inc. New York.
- Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B*, **58**, 267–288.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005). Sparsity and smoothness via the fused Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **67**, 91–108.
- Timm, N. H. (2002). *Applied Multivariate Analysis*. Springer-Verlag. New York.
- Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, **109**, 475–494.

von Rosen, D. (1991). The growth curve model: A review. *Comm. Statist. Theory Methods*, **20**, 2791–2822.

von Rosen, D. (2018). *Bilinear Regression Analysis: An introduction*. Springer. Cham.

Yanagihara, H. & Oda, R. (2021). Coordinate descent algorithm for normal-likelihood-based group Lasso in multivariate linear regression. I. Czarnowski, R. J. Howlett & L. C. Jain, eds, *Intelligent Decision Technologies*. Springer Singapore. Singapore. 429–439.

Yuan, M. & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **68**, 49–67.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *J. Amer. Statist. Assoc.*, **101**, 1418–1429.

## Appendix

### A.1. Proof of Proposition 1

Notice that it is obvious when  $j \leq q - 2 \wedge j + 1 \leq \alpha \leq q - 1$  and  $1 \leq j = \alpha \leq q - 1$ . Hence, we prove the following two cases:

$$d_{\alpha,j} - d_{\alpha+1,j} = \begin{cases} \left( \sqrt{d_{\alpha,\alpha}^2 + b_{\alpha+1}^2} - \lambda_{\alpha+1} \right)_+ - (|b_{\alpha+1}| - \lambda_{\alpha+1})_+ & \text{(C1: } 2 \leq \alpha + 1 = j \leq q) \\ \left( \sqrt{d_{\alpha,j-1}^2 + b_j^2} - \lambda_j \right)_+ - \left( \sqrt{d_{\alpha+1,j-1}^2 + b_j^2} - \lambda_j \right)_+ & \text{(C2: } 3 \leq j \wedge \alpha \leq j - 2) \end{cases}.$$

When C1, since  $(d_{\alpha,\alpha}^2 + b_{\alpha+1}^2)^{1/2} \geq |b_{\alpha+1}|$ , we have  $d_{\alpha,\alpha+1} \geq d_{\alpha+1,\alpha+1}$ . Regarding C2, given the result for C1, it follows for  $j = \alpha + 2$  that

$$d_{\alpha,\alpha+2} = \left( \sqrt{d_{\alpha,\alpha+1}^2 + b_{\alpha+2}^2} - \lambda_{\alpha+2} \right)_+ \geq \left( \sqrt{d_{\alpha+1,\alpha+1}^2 + b_{\alpha+2}^2} - \lambda_{\alpha+2} \right)_+ = d_{\alpha+1,\alpha+2}.$$

Suppose that  $d_{\alpha,j_0} \geq d_{\alpha+1,j_0}$  holds for  $j = j_0 \in \{\alpha + 2, \alpha + 3, \dots, q - 1\}$ . Then, we have

$$d_{\alpha,j_0+1} = \left( \sqrt{d_{\alpha,j_0}^2 + b_{j_0+1}^2} - \lambda_{j_0+1} \right)_+ \geq \left( \sqrt{d_{\alpha+1,j_0}^2 + b_{j_0+1}^2} - \lambda_{j_0+1} \right)_+ = d_{\alpha+1,j_0+1}.$$

Hence, mathematical induction leads to  $d_{\alpha,j} \geq d_{\alpha+1,j}$  for C2; consequently, Proposition 1 is proved.

## A.2. Proofs of Theorems 1 and 4

We prove Theorem 4. The proof of Theorem 1 follows by setting  $m_1 = \dots = m_q = 1$ . We define a set of block vectors as

$$\mathbb{R}^{m_1+\dots+m_q} = \left\{ \mathbf{x} = (\mathbf{x}'_1, \dots, \mathbf{x}'_q)' \in \mathbb{R}^m \mid \mathbf{x}_j \in \mathbb{R}^{m_j} (j = 1, \dots, q) \right\}.$$

Note that although  $\mathbb{R}^m = \mathbb{R}^{m_1+\dots+m_q}$  holds, we use the notation  $\mathbb{R}^{m_1+\dots+m_q}$  to emphasize the sizes of each block. For a block vector  $\mathbf{x} \in \mathbb{R}^{m_1+\dots+m_q}$ , we define sub-vector  $\mathbf{x}_{[a:b]} \in \mathbb{R}^{m_a+\dots+m_b}$  as

$$\mathbf{x}_{[a:b]} = (\mathbf{x}'_a, \mathbf{x}'_{a+1}, \dots, \mathbf{x}'_b)' \quad (1 \leq a \leq b \leq q).$$

Note that  $\mathbf{x}_{(j)} = \mathbf{x}_{[1,j]}$  holds, where  $\mathbf{x}_{(j)}$  is a sub-vector of  $\mathbf{x}$  given by (3.4). The following lemma is a key to proving the theorem (the proof is given in Appendix A.3).

**Lemma A.1.** *Consider the following system of equations for  $\gamma \in \mathbb{R}^{m_1+\dots+m_q}$ .*

$$\left( 1 + \sum_{\ell=j}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|} \right) \gamma_j = \mathbf{b}_j \quad (\gamma_1 \neq \mathbf{0}_{m_1}; j = 1, \dots, q). \quad (\text{A.1})$$

A necessary and sufficient condition such that the system of equations has a real root is given by

$$\forall j \in \{1, \dots, q\}, d_j > 0,$$

and the real root is given by

$$\gamma_j = \prod_{\ell=j}^q \frac{d_\ell}{d_\ell + \lambda_\ell} \mathbf{b}_j, \quad d_j = \begin{cases} \|\mathbf{b}_1\| - \lambda_1 & (j = 1) \\ \sqrt{d_{j-1}^2 + \|\mathbf{b}_j\|^2} - \lambda_j & (j = 2, \dots, q) \end{cases}.$$

We redefine  $f$  in (3.5) as

$$f(\gamma \mid \mathbf{b}, \boldsymbol{\lambda}) = \frac{1}{2} \|\gamma\|^2 - \mathbf{b}'\gamma + \sum_{j=1}^q \lambda_j \|\gamma_{(j)}\|, \quad \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_q)',$$

and define two sets of block vectors as

$$\mathcal{B}_q^{m_1+\dots+m_q} = \{ \mathbf{x} \in \mathbb{R}^{m_1+\dots+m_q} \mid \mathbf{x}_1 \neq \mathbf{0}_{m_1} \}, \quad \mathcal{R}_{q,\alpha}^{m_1+\dots+m_q} = \{ \mathbf{x} \in \mathbb{R}^{m_1+\dots+m_q} \mid \mathbf{x}_{[1:\alpha]} = \mathbf{0}_{m_1+\dots+m_\alpha} \}.$$

For all  $\gamma \in \mathcal{R}_{q,\alpha}^{m_1+\dots+m_q}$ , we have

$$\begin{aligned} f(\gamma \mid \mathbf{b}, \boldsymbol{\lambda}) &= \frac{1}{2} \|\gamma_{[\alpha+1:q]}\|^2 - \mathbf{b}'_{[\alpha+1:q]} \gamma_{[\alpha+1:q]} + \sum_{j=1}^{q-\alpha} \lambda_{\alpha+j} \|(\gamma_{[\alpha+1:q]})_{(j)}\| \\ &= f(\gamma_{[\alpha+1:q]} \mid \mathbf{b}_{[\alpha+1:q]}, \boldsymbol{\lambda}_{[\alpha+1:q]}). \end{aligned}$$

Furthermore,  $f$  is differentiable for  $\gamma \in \mathcal{B}_q^{m_1+\dots+m_q}$  and its partial derivative is given by

$$\nabla_j f(\gamma \mid \mathbf{b}, \boldsymbol{\lambda}) = \frac{\partial f}{\partial \gamma_j}(\gamma \mid \mathbf{b}, \boldsymbol{\lambda}) = \gamma_j - \mathbf{b}_j + \sum_{\ell=j}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|} \gamma_j = \left(1 + \sum_{\ell=j}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \gamma_j - \mathbf{b}_j.$$

For all  $\alpha \in \{1, \dots, q\}$ ,  $f(\gamma_{[\alpha:q]} \mid \mathbf{b}_{[\alpha:q]}, \boldsymbol{\lambda}_{[\alpha:q]})$  is differentiable for  $\gamma_{[\alpha:q]} \in \mathcal{B}_{q-\alpha+1}^{m_\alpha+\dots+m_q}$ . Let  $\mathbf{s}_\alpha$  be a root of the following system of equations when  $\gamma_{[\alpha:q]} \in \mathcal{B}_{q-\alpha+1}^{m_\alpha+\dots+m_q}$ :

$$\nabla_j f(\gamma_{[\alpha:q]} \mid \mathbf{b}_{[\alpha:q]}, \boldsymbol{\lambda}_{[\alpha:q]}) = \mathbf{0}_{m_j} \quad (j = \alpha, \alpha + 1, \dots, q).$$

Lemma A.1 tells us whether  $\mathbf{s}_\alpha$  exists and  $\mathbf{s}_\alpha$  is given in closed form if it exists. Notice that  $f$  is a strictly convex function. Hence, we have

$$\begin{cases} \mathbf{s}_\alpha \in \mathcal{B}_{q-\alpha+1}^{m_\alpha+\dots+m_q} \text{ exists} & \implies \mathbf{s}_\alpha = \arg \min_{\gamma_{[\alpha:q]}} f(\gamma_{[\alpha:q]} \mid \mathbf{b}_{[\alpha:q]}, \boldsymbol{\lambda}_{[\alpha:q]}) \\ \mathbf{s}_\alpha \in \mathcal{B}_{q-\alpha+1}^{m_\alpha+\dots+m_q} \text{ does not exist} & \implies \arg \min_{\gamma_{[\alpha:q]}} f(\gamma_{[\alpha:q]} \mid \mathbf{b}_{[\alpha:q]}, \boldsymbol{\lambda}_{[\alpha:q]}) \in \mathcal{R}_{q-\alpha+1,1}^{m_\alpha+\dots+m_q}. \end{cases}$$

Thus, the condition such that  $\mathbf{s}_\alpha$  exists is given by

$$\forall j \in \{\alpha, \alpha + 1, \dots, q\}, d_{\alpha,j} > 0,$$

where  $d_{\alpha,j}$  is given in (3.6). If  $\mathbf{s}_\alpha$  does not exist, we check the existence of  $\mathbf{s}_{\alpha+1}$ , which is the minimizer of  $f(\gamma_{[\alpha+1:q]} \mid \mathbf{b}_{[\alpha+1:q]}, \boldsymbol{\lambda}_{[\alpha+1:q]})$ , by the same procedure. By repeating the procedure for  $\alpha = 1, 2, \dots$ , we have

$$\begin{cases} \mathbf{s}_1 \in \mathcal{B}_q^{m_1+\dots+m_q} \text{ exists} & \implies \boldsymbol{\gamma}^* = \mathbf{s}_1 \\ \mathbf{s}_1 \in \mathcal{B}_q^{m_1+\dots+m_q} \text{ does not exist} & \implies \boldsymbol{\gamma}^* \in \mathcal{R}_{q,1}^{m_1+\dots+m_q}, \end{cases}$$

$$\boldsymbol{\gamma}^* \in \mathcal{R}_{q,\alpha-1}^{m_1+\dots+m_q} \text{ and } \begin{cases} \mathbf{s}_\alpha \in \mathcal{B}_{q-\alpha+1}^{m_\alpha+\dots+m_q} \text{ exists} & \implies \boldsymbol{\gamma}^* = (\mathbf{0}'_{m_1+\dots+m_{\alpha-1}}, \mathbf{s}'_\alpha)' \\ \mathbf{s}_\alpha \in \mathcal{B}_{q-\alpha+1}^{m_\alpha+\dots+m_q} \text{ does not exist} & \implies \boldsymbol{\gamma}^* \in \mathcal{R}_{q,\alpha}^{m_1+\dots+m_q} \end{cases}.$$

Consequently, Theorems 1 and 4 are proved.

### A.3. Proof of Lemma A.1

Since  $\gamma_j \neq \mathbf{0}_{m_j} \Leftrightarrow \|\gamma_{(j)}\| \neq 0$  holds for all  $j \in \{1, \dots, q\}$ , (A.1) exists. We first derive  $\gamma$  satisfying (A.1). When  $j = 1$ , from  $\|\gamma_{(1)}\| = \|\gamma_1\|$ , (A.1) implies

$$\left(1 + \sum_{\ell=2}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|} + \frac{\lambda_1}{\|\gamma_1\|}\right) \|\gamma_1\| = \|\mathbf{b}_1\| \iff \left(1 + \sum_{\ell=2}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \|\gamma_1\| = d_1 > 0.$$

When  $j = j_0$  ( $\in \{1, \dots, q-2\}$ ), we suppose

$$\left(1 + \sum_{\ell=j_0+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \|\gamma_{(j_0)}\| = d_{j_0} > 0. \quad (\text{A.2})$$

Notice that  $\|\gamma_{j+1}\|^2 + \|\gamma_{(j)}\|^2 = \|\gamma_{(j+1)}\|^2$ . Hence, squaring (A.1) for  $j = j_0 + 1$  and (A.2) and adding the results yields

$$\left(1 + \sum_{\ell=j_0+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right)^2 \|\gamma_{(j_0+1)}\|^2 = d_{j_0}^2 + \|\mathbf{b}_{j_0+1}\|^2.$$

By taking the square root of both sides in the above equation, it follows from the same procedure when  $j = 1$  that

$$\begin{aligned} \left(1 + \sum_{\ell=j_0+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \|\gamma_{(j_0+1)}\| &= \left(1 + \sum_{\ell=j_0+2}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \|\gamma_{(j_0+1)}\| + \lambda_{j_0+1} = \sqrt{d_{j_0}^2 + \|\mathbf{b}_{j_0+1}\|^2} \\ \iff \left(1 + \sum_{\ell=j_0+2}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \|\gamma_{(j_0+1)}\| &= d_{j_0+1} > 0. \end{aligned}$$

Hence, mathematical induction gives

$$\left(1 + \sum_{\ell=j+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) \|\gamma_{(j)}\| = d_j > 0, \quad (\text{A.3})$$

for  $j = 1, \dots, q-1$ . Furthermore, the same procedure with (A.1) for  $j = q$  and (A.2) for  $j_0 = q-1$  yields

$$\left(1 + \frac{\lambda_q}{\|\gamma\|}\right) \|\gamma\| = \sqrt{d_{q-1}^2 + \|\mathbf{b}_q\|^2}.$$

From the results above, we have

$$\|\gamma_{(j)}\| = \begin{cases} \left(1 + \sum_{\ell=j+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right)^{-1} d_j & (j = 1, \dots, q-1) \\ d_q & (j = q) \end{cases}.$$

Furthermore, this result implies

$$\begin{aligned} \|\gamma_{(j)}\| &= \left(1 + \sum_{\ell=j+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right)^{-1} d_j = \left(1 + \sum_{\ell=j+2}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|} + \frac{\lambda_{j+1}}{\|\gamma_{(j+1)}\|}\right)^{-1} d_j, \\ \left(1 + \sum_{\ell=j+1}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|}\right) &= \frac{d_j}{\|\gamma_{(j)}\|}, \end{aligned}$$

for  $j = 1, \dots, q-2$  and  $\|\gamma_{(q)}\| = d_q$ . Hence, for  $j = 1, \dots, q-1$ , we have

$$\|\gamma_{(j)}\| = \left( \frac{d_{j+1}}{\|\gamma_{(j+1)}\|} + \frac{\lambda_{j+1}}{\|\gamma_{(j+1)}\|} \right)^{-1} d_j = \frac{d_j}{d_{j+1} + \lambda_{j+1}} \|\gamma_{(j+1)}\|.$$

Since  $\|\gamma_{[1:q]}\| = d_q$ , the above equation implies

$$\begin{aligned} \frac{\|\gamma_{(j)}\|}{d_j} &= \frac{d_{j+1}}{d_{j+1} + \lambda_{j+1}} \frac{\|\gamma_{(j+1)}\|}{d_{j+1}} = \frac{d_{j+1}}{d_{j+1} + \lambda_{j+1}} \frac{d_{j+2}}{d_{j+2} + \lambda_{j+2}} \frac{\|\gamma_{[1:j+2]}\|}{d_{j+2}} \\ &= \dots = \prod_{\ell=j+1}^q \frac{d_\ell}{d_\ell + \lambda_\ell} \frac{\|\gamma_{[1:q]}\|}{d_q} = \prod_{\ell=j+1}^q \frac{d_\ell}{d_\ell + \lambda_\ell}. \end{aligned}$$

Hence, we have

$$\begin{aligned} \|\gamma_1\| &= d_1 \prod_{\ell=2}^q \frac{d_\ell}{d_\ell + \lambda_\ell} = (d_1 + \lambda_1) \prod_{\ell=1}^q \frac{d_\ell}{d_\ell + \lambda_\ell} = \|\mathbf{b}_1\| \prod_{\ell=1}^q \frac{d_\ell}{d_\ell + \lambda_\ell}, \\ \|\gamma_j\| &= \sqrt{\|\gamma_{(j)}\|^2 - \|\gamma_{(j-1)}\|^2} = \sqrt{\|\gamma_{(j)}\|^2 - \left( \frac{d_{j-1}}{d_j + \lambda_j} \right)^2 \|\gamma_{(j)}\|^2} = \frac{\|\mathbf{b}_j\|}{d_j + \lambda_j} \|\gamma_{(j)}\| \\ &= \|\mathbf{b}_j\| \frac{d_j}{d_j + \lambda_j} \frac{\|\gamma_{(j)}\|}{d_j} = \|\mathbf{b}_j\| \prod_{\ell=j}^q \frac{d_\ell}{d_\ell + \lambda_\ell} \quad (j = 2, \dots, q). \end{aligned}$$

Moreover, it follows from (A.1) that

$$\|\gamma_j\| = \left( 1 + \sum_{\ell=j}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|} \right)^{-1} \|\mathbf{b}_j\|,$$

and we have

$$\left( 1 + \sum_{\ell=j}^q \frac{\lambda_\ell}{\|\gamma_{(\ell)}\|} \right)^{-1} = \prod_{\ell=j}^q \frac{d_\ell}{d_\ell + \lambda_\ell}.$$

Thus,  $\gamma$  satisfying (A.1) is given by

$$\gamma_j = \prod_{\ell=j}^q \frac{d_\ell}{d_\ell + \lambda_\ell} \mathbf{b}_j \quad (j = 1, \dots, q).$$

Next, we give a necessary and sufficient condition for a root of (A.1) existing. As described above, we have the following sufficient condition:

$$\forall j \in \{1, \dots, q\}, d_j > 0 \implies \exists \gamma \in \mathbb{R}^{m_1 + \dots + m_q} \text{ s.t. (A.1) holds.}$$

On the other hand, if  $j$  exists such that  $d_j \leq 0$  holds, (A.3) does not hold, and hence a real root

of (A.1) does not exist. This implies

$$\exists j \in \{1, \dots, q\} \text{ s.t. } d_j \leq 0 \implies \forall \gamma \in \mathbb{R}^{m_1 + \dots + m_q}, \text{ (A.1) does not hold.}$$

The contrapositive of this leads to the following necessary condition:

$$\exists \gamma \in \mathbb{R}^{m_1 + \dots + m_q} \text{ s.t. (A.1) holds} \implies \forall j \in \{1, \dots, q\}, d_j > 0.$$

Thus, we have the necessary and sufficient condition. Consequently, Lemma A.1 is proved.

#### A.4. Proof of Theorem 3

We set  $\lambda_j = \lambda w_j / L$  for  $f$  in (2.4). Theorem 1 implies a sufficient condition such that  $f$  is minimized at  $\gamma = \mathbf{0}_q$  as

$$\forall j \in \{1, \dots, q\}, d_{j,j} = 0 \iff \lambda \geq \max_{j \in \{1, \dots, q\}} \frac{L|b_j|}{w_j}. \quad (\text{A.4})$$

With this condition, we derive the condition for  $\lambda$  such that  $\text{PRSS}(\boldsymbol{\theta} \mid \delta, \lambda)$  is minimized at  $\boldsymbol{\theta} = \mathbf{0}_{kq}$ . Notice that the optimality of Algorithm 2 means that we can obtain the optimal solution for an arbitrary initial solution. Hence, it is sufficient to consider the condition such that  $\text{PRSS}^+(\boldsymbol{\theta} \mid \mathbf{0}_{kq}, \delta, \lambda)$  is minimized at  $\boldsymbol{\theta} = \mathbf{0}_{kq}$ . At the minimization of (3.2), a sufficient condition for  $\lambda$  such that (3.2) is minimized at  $\boldsymbol{\theta}_\ell = \mathbf{0}_q$  when  $\hat{\boldsymbol{\theta}} = \mathbf{0}_{kq}$  follows (A.4). Consequently, Theorem 3 is proved.

#### A.5. Details of the transformation of the matrix of basis functions

Let  $F_q$  be a  $q \times q$  anti-diagonal matrix with anti-diagonal elements ones. Then, the QR decomposition yields  $(\mathbf{v}_q, \mathbf{v}_{q-1}, \dots, \mathbf{v}_1) = \mathbf{V} F_q = \mathbf{H}_0 \mathbf{Q}_0$ , where  $\mathbf{H}_0$  is a  $p \times q$  matrix satisfying  $\mathbf{H}_0' \mathbf{H}_0 = \mathbf{I}_q$  and  $\mathbf{Q}_0$  is a  $q \times q$  upper triangular matrix. From  $F_q F_q = \mathbf{I}_q$ , this decomposition implies  $\mathbf{V} = \mathbf{H}_0 F_q F_q \mathbf{Q}_0 F_q$  and we have  $\mathbf{H} = \mathbf{H}_0 F_q$  and  $\mathbf{Q} = F_q \mathbf{Q}_0 F_q$ .

#### A.6. Simulation results for SNR = 3

This section shows the simulation results described in Section 4 when SNR = 3. In comparison with SNR = 1, we can see that all the methods performed better.

HOGL for GMANOVA

Table A.1. Selection probability (%) when  $q = 6$

$n$	$p$	$k$	Variable				Degree			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	28.2	<b>91.6</b>	19.5	6.9	12.9	<b>72.7</b>	4.7	0.9
300	10	10	70.8	<b>97.7</b>	55.9	85.4	40.8	<b>92.4</b>	14.2	75.2
500	10	10	82.9	<b>98.2</b>	69.5	95.2	52.6	<b>95.1</b>	23.0	85.2
100	40	10	9.5	<b>70.1</b>	9.5	4.4	8.1	<b>54.9</b>	2.0	0.1
300	120	10	64.7	89.1	59.2	<b>91.2</b>	36.3	82.3	7.4	<b>83.5</b>
500	200	10	84.0	93.1	81.6	<b>98.8</b>	51.7	85.6	15.8	<b>96.2</b>
100	10	40	17.8	<b>82.7</b>	6.0	0.1	7.1	<b>68.9</b>	5.5	0.1
300	10	120	29.0	<b>94.6</b>	12.5	16.5	16.8	<b>89.7</b>	10.3	83.4
500	10	200	62.8	<b>97.5</b>	21.7	14.3	22.2	<b>94.2</b>	6.7	92.7
100	20	40	7.3	<b>69.7</b>	2.0	0.5	5.3	<b>63.2</b>	4.0	0.0
300	60	120	36.4	<b>91.5</b>	11.2	82.6	9.9	86.2	4.0	<b>93.9</b>
500	100	200	76.8	<b>92.5</b>	34.8	87.9	26.1	93.0	5.6	<b>97.3</b>

Table A.2. MSE when  $q = 6$

$n$	$p$	$k$	MSE <sub>f</sub>				MSE <sub>c</sub>			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	0.061	<b>0.037</b>	0.064	0.086	0.438	<b>0.176</b>	0.619	1.273
300	10	10	0.017	<b>0.011</b>	0.020	0.035	0.096	<b>0.044</b>	0.171	0.248
500	10	10	0.009	<b>0.006</b>	0.011	0.013	0.048	<b>0.023</b>	0.095	0.091
100	40	10	0.027	0.020	0.030	<b>0.018</b>	1.317	<b>0.644</b>	2.828	3.623
300	120	10	0.005	0.004	0.005	<b>0.001</b>	0.409	<b>0.184</b>	1.055	0.462
500	200	10	0.003	0.002	0.003	<b>0.000</b>	0.247	<b>0.113</b>	0.632	0.131
100	10	40	0.197	<b>0.046</b>	0.147	0.622	0.469	<b>0.060</b>	0.222	0.439
300	10	120	0.061	<b>0.012</b>	0.059	0.126	0.042	<b>0.004</b>	0.032	0.052
500	10	200	0.031	<b>0.007</b>	0.035	0.036	0.011	<b>0.001</b>	0.012	0.010
100	20	40	0.118	<b>0.032</b>	0.087	0.219	0.812	<b>0.121</b>	0.389	0.805
300	60	120	0.016	<b>0.005</b>	0.015	0.030	0.105	<b>0.014</b>	0.087	0.127
500	100	200	0.005	<b>0.003</b>	0.005	0.003	0.025	<b>0.005</b>	0.030	0.017

Table A.3. Selection probability (%) when  $q = 10$

$n$	$p$	$k$	Variable				Degree			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	0.8	<b>88.0</b>	18.3	0.4	0.0	<b>57.6</b>	0.0	0.0
300	10	10	0.4	<b>97.6</b>	32.8	0.4	0.0	<b>88.3</b>	0.0	0.0
500	10	10	0.1	<b>98.1</b>	38.2	0.3	0.0	<b>92.7</b>	0.0	0.0
100	40	10	0.4	<b>67.6</b>	22.4	0.6	0.0	<b>40.1</b>	0.0	0.0
300	120	10	0.1	<b>89.5</b>	47.1	0.9	0.0	<b>78.0</b>	0.0	0.1
500	200	10	0.0	<b>93.5</b>	60.5	1.8	0.0	<b>84.0</b>	0.0	0.2
100	10	40	0.4	<b>75.3</b>	13.8	0.0	0.0	<b>50.6</b>	0.0	0.0
300	10	120	0.3	<b>92.8</b>	9.0	0.0	0.0	<b>81.5</b>	0.0	0.0
500	10	200	0.1	<b>95.7</b>	7.4	0.0	0.0	<b>90.1</b>	0.0	0.0
100	20	40	0.3	<b>65.0</b>	5.7	0.0	0.0	<b>46.9</b>	0.0	0.0
300	60	120	0.0	<b>90.1</b>	13.6	1.5	0.0	<b>79.4</b>	0.0	0.0
500	100	200	0.0	<b>92.7</b>	17.1	0.3	0.0	<b>90.3</b>	0.0	0.0

Table A.4. MSE when  $q = 10$

$n$	$p$	$k$	MSE <sub>f</sub>				MSE <sub>c</sub>			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	0.127	<b>0.040</b>	0.162	0.266	4276.068	<b>0.826</b>	4004.717	4310.401
300	10	10	0.041	<b>0.011</b>	0.049	0.065	1262.396	<b>0.034</b>	1129.817	1271.469
500	10	10	0.025	<b>0.007</b>	0.029	0.030	741.832	<b>0.018</b>	661.475	751.059
100	40	10	0.049	<b>0.021</b>	0.048	0.048	1737.444	<b>1.019</b>	1579.390	1339.786
300	120	10	0.008	0.004	0.007	<b>0.004</b>	861.548	<b>0.266</b>	739.860	632.382
500	200	10	0.004	0.002	0.003	<b>0.001</b>	577.077	<b>0.101</b>	483.212	427.223
100	10	40	2.155	<b>0.052</b>	0.915	2.352	31.219	<b>0.071</b>	1056.697	98.678
300	10	120	1.431	<b>0.013</b>	0.550	2.309	6.919	<b>0.004</b>	133.269	7.512
500	10	200	1.022	<b>0.007</b>	0.410	2.011	2.401	<b>0.001</b>	49.203	3.439
100	20	40	0.648	<b>0.035</b>	0.339	1.028	20.777	<b>0.113</b>	259.949	148.508
300	60	120	0.163	<b>0.005</b>	0.064	0.099	6.517	<b>0.011</b>	65.318	94.937
500	100	200	0.083	<b>0.003</b>	0.030	0.045	3.109	<b>0.004</b>	28.790	48.901

Table A.5. Runtime (sec.)

$n$	$p$	$k$	$q = 6$				$q = 10$			
			HOGL1	HOGL2	SGL	wSCAD	HOGL1	HOGL2	SGL	wSCAD
100	10	10	138.8	<b>4.9</b>	10.9	17.6	148.3	<b>7.0</b>	7.2	20.0
300	10	10	112.2	<b>4.3</b>	10.4	15.8	167.4	<b>6.2</b>	7.7	19.3
500	10	10	94.5	<b>4.1</b>	10.7	14.3	175.3	<b>5.9</b>	8.2	20.0
100	40	10	137.2	<b>4.9</b>	9.9	16.4	168.2	6.8	<b>6.8</b>	19.7
300	120	10	95.3	<b>4.4</b>	16.8	14.4	189.8	<b>5.8</b>	23.7	18.4
500	200	10	77.9	<b>5.8</b>	43.5	16.1	194.1	<b>7.3</b>	66.9	26.2
100	10	40	1203.6	<b>33.0</b>	133.0	76.8	1678.3	<b>45.8</b>	83.4	94.3
300	10	120	3533.1	<b>85.9</b>	872.9	455.0	14674.1	<b>122.9</b>	680.3	1090.2
500	10	200	4775.6	<b>132.1</b>	2538.0	1561.1	42591.9	<b>193.5</b>	2091.8	4737.2
100	20	40	1183.6	<b>30.6</b>	93.0	72.0	1954.3	<b>43.1</b>	44.5	83.0
300	60	120	3011.6	<b>62.2</b>	936.3	412.1	17813.8	<b>104.8</b>	534.7	1044.1
500	100	200	2874.1	<b>87.4</b>	3410.7	1277.9	47408.6	<b>125.6</b>	2769.9	4189.2