Functional Sparse Estimation of Time Varying Graphical Model

Meilei Jiang[†]

[†]Department of Statistics and Operations Research
University of North Carolina at Chapel Hill

April 30, 2016

Overview

Outline

1. Introduction
2. Sparse Gaussian Graphical Model Estimation
   - Neighborhood Selection Approach
   - Penalized Likelihood Estimation Approach
   - Constrained ℓ_1 Minimization Approach
3. Heterogeneous Data And Joint Estimation Of Multiple Graphs
4. Time Varying Graphical Model
   - Varying Coefficient Model
   - Functional Coefficient Model
5. Simulation Study

Background

Precision matrix plays a fundamental role in many high dimensional inference problems.

- Classification and discriminant analysis.

Background

Precision matrix plays a fundamental role in many high dimensional inference problems.

- Classification and discriminant analysis.
- Applications: portfolio optimization, speech recognition and genomics.

Background

Precision matrix plays a fundamental role in many high dimensional inference problems.

- Classification and discriminant analysis.
- Applications: portfolio optimization, speech recognition and genomics.
- Graphical Model: recovering the structure of undirected Gaussian graph is equivalent to the support of the precision matrix.

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.
- Graph $\mathcal{G} = (V, E)$ (Lauritzen, 1996).

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.
- Graph $\mathcal{G} = (V, E)$ (Lauritzen, 1996).
  - $V = \{1, \cdots, p\}$ is the set of nodes.

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \boldsymbol{\Sigma})$
  - Precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.
- Graph $\mathcal{G} = (V, E)$ (Lauritzen, 1996).
  - $V = \{1, \cdots, p\}$ is the set of nodes.
  - $E = \left\{(j, l) | X_j \text{ is conditionally dependent with } X_l, \text{given } X_{V/\{j,l\}}\right\}$

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.
- Graph $\mathcal{G} = (V, E)$ (Lauritzen, 1996).
  - $V = \{1, \cdots, p\}$ is the set of nodes.
  - $E = \left\{(j, l) | X_j \text{ is conditionally dependent with } X_l, \text{given } X_{V/\{j,l\}}\right\}$
  - $e_{jl} \in E$ if and only if $\rho^{jl} \neq 0$

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.
- Graph $\mathcal{G} = (V, E)$ (Lauritzen, 1996).
  - $V = \{1, \cdots, p\}$ is the set of nodes.
  - $E = \left\{(j, l) | X_j \text{ is conditionally dependent with } X_l, \text{given } X_{V/\{j,l\}}\right\}$
  - $e_{jl} \in E$ if and only if $\rho^{jl} \neq 0$
  - $e_{jl} \in E$ if and only if $\omega_{jl} \neq 0$

Gaussian Graphical Model

- $\mathbf{X} = (X_1, \cdots, X_p) \sim \mathbb{N}(\mathbf{0}, \mathbf{\Sigma})$
  - Precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.
  - Partial correlation between X_j and X_l: $\rho^{jl} = -(\omega_{jl}/\sqrt{\omega_{jj}\omega_{ll}})$.
- Graph $\mathcal{G} = (V, E)$ (Lauritzen, 1996).
  - $V = \{1, \cdots, p\}$ is the set of nodes.
  - $E = \{(j, l) | X_j \text{ is conditionally dependent with } X_l, \text{given } X_{V/\{j,l\}}\}$
  - $e_{jl} \in E$ if and only if $\rho^{jl} \neq 0$
  - $e_{jl} \in E$ if and only if $\omega_{jl} \neq 0$
- Estimation of $\mathcal{G}$ is equivalent to recover the non-zero entry of the precision matrix $\mathbf{\Omega}$.

Outline

1. Introduction
2. Sparse Gaussian Graphical Model Estimation
   - Neighborhood Selection Approach
   - Penalized Likelihood Estimation Approach
   - Constrained ℓ_1 Minimization Approach
3. Heterogeneous Data And Joint Estimation Of Multiple Graphs
4. Time Varying Graphical Model
   - Varying Coefficient Model
   - Functional Coefficient Model
5. Simulation Study

Estimation of Ω

- A random sample $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(n)}$ of $\mathbf{X}$

Estimation of Ω

- A random sample $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(n)}$ of $\mathbf{X}$
- When $n > p$, a nature estimator of $\mathbf{\Omega}$: $\hat{\mathbf{\Omega}}_n = \hat{\mathbf{\Sigma}}_n^{-1}$

Estimation of Ω

- A random sample $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(n)}$ of $\mathbf{X}$
- When $n > p$, a nature estimator of $\mathbf{\Omega}$: $\hat{\mathbf{\Omega}}_n = \hat{\mathbf{\Sigma}}_n^{-1}$
  - $\hat{\mathbf{\Sigma}}_n$ is the sample covariance matrix

Estimation of Ω

- A random sample $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(n)}$ of $\mathbf{X}$
- When $n > p$, a nature estimator of $\mathbf{\Omega}$: $\hat{\mathbf{\Omega}}_n = \hat{\mathbf{\Sigma}}_n^{-1}$
  - $\hat{\mathbf{\Sigma}}_n$ is the sample covariance matrix
- When $n < p$, the estimation of $\mathbf{\Omega}$ becomes much more challenging due to singularity of $\hat{\mathbf{\Sigma}}_n$.

Neighborhood Selection And Linear Regression

Neighborhood Selection And Linear Regression

Connection between linear regression and prediction matrix $\boldsymbol{\Omega}$: For each $j \in \{1, \cdots, n\}$

$$\mathbf{X}_j = \mathbf{X}_{-j}\boldsymbol{\beta}_j + \varepsilon_j = \sum_{l \neq j} \mathbf{X}_l \beta_{jl} + \varepsilon_j \tag{1}$$

- $\beta_{jl} = \omega_{jl}/\omega_{jj}$

Neighborhood Selection And Linear Regression

Connection between linear regression and prediction matrix $\boldsymbol{\Omega}$: For each $j \in \{1, \cdots, n\}$

$$\mathbf{X}_j = \mathbf{X}_{-j}\boldsymbol{\beta}_j + \varepsilon_j = \sum_{l \neq j} \mathbf{X}_l \beta_{jl} + \varepsilon_j \tag{1}$$

- $\beta_{jl} = \omega_{jl}/\omega_{jj}$
- The support of $\boldsymbol{\beta}_j = \{\beta_{jl}; l \neq j\}$ (non-zero entries) is the neighborhood of node j.

Neighborhood Selection And Linear Regression

Connection between linear regression and prediction matrix $\boldsymbol{\Omega}$: For each $j \in \{1, \cdots, n\}$

$$\mathbf{X}_j = \mathbf{X}_{-j}\boldsymbol{\beta}_j + \varepsilon_j = \sum_{l \neq j} \mathbf{X}_l \beta_{jl} + \varepsilon_j \tag{1}$$

- $\beta_{jl} = \omega_{jl}/\omega_{jj}$
- The support of $\boldsymbol{\beta}_j = \{\beta_{jl}; l \neq j\}$ (non-zero entries) is the neighborhood of node j.
- Select the non-zero entry for jth row of $\boldsymbol{\Omega}$ is equivalent to the multivariate regression problem (1).

Neighborhood Selection Approach

- Meinshausen and Bühlmann (2006) applied Lasso least square regression on model (1) to estimate the support of $\mathbf{\Omega}$ row by row.

Neighborhood Selection Approach

- Meinshausen and Bühlmann (2006) applied Lasso least square regression on model (1) to estimate the support of $\boldsymbol{\Omega}$ row by row.
- Peng et al. (2012) considered a joint sparse regression to estimate the p rows of $\boldsymbol{\Omega}$ together.

Neighborhood Selection Approach

- Meinshausen and Bühlmann (2006) applied Lasso least square regression on model (1) to estimate the support of $\boldsymbol{\Omega}$ row by row.
- Peng et al. (2012) considered a joint sparse regression to estimate the p rows of $\boldsymbol{\Omega}$ together.
- Yuan (2010) applied Danzig selector on the multivariate regression problem (1) and estimated the entries of $\boldsymbol{\Omega}$ directly.

Neighborhood Selection Approach

- Meinshausen and Bühlmann (2006) applied Lasso least square regression on model (1) to estimate the support of $\boldsymbol{\Omega}$ row by row.
- Peng et al. (2012) considered a joint sparse regression to estimate the p rows of $\boldsymbol{\Omega}$ together.
- Yuan (2010) applied Danzig selector on the multivariate regression problem (1) and estimated the entries of $\boldsymbol{\Omega}$ directly.
- Graph structures can be recovered consistently in a high dimension settings. (Meinshausen and Bühlmann, 2006; Peng et al, 2012)

Log-likelihood Estimation

Another nature way is to estimate $\mathbf{\Omega}$ is the penalized likelihood approach.

- Log-likelihood function: $l(\mathbf{X}^(i), 1 \leq i \leq n|\mathbf{\Omega}) = -\mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) + \log|\mathbf{\Omega}|$

Log-likelihood Estimation

Another nature way is to estimate $\mathbf{\Omega}$ is the penalized likelihood approach.

- Log-likelihood function: $l(\mathbf{X}^(i), 1 \leq i \leq n|\mathbf{\Omega}) = -\mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) + \log|\mathbf{\Omega}|$
- Penalized likelihood estimator:

$$\hat{\mathbf{\Omega}}_L = \arg\min \mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) - \log|\mathbf{\Omega}| + \sum_{j \neq l} \lambda_{jl} p_{jl}(|\omega_{jl}|) \tag{2}$$

Log-likelihood Estimation

Another nature way is to estimate $\mathbf{\Omega}$ is the penalized likelihood approach.

- Log-likelihood function: $l(\mathbf{X}^(i), 1 \leq i \leq n|\mathbf{\Omega}) = -\mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) + \log|\mathbf{\Omega}|$
- Penalized likelihood estimator:

$$\hat{\mathbf{\Omega}}_L = \arg\min \mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) - \log|\mathbf{\Omega}| + \sum_{j \neq l} \lambda_{jl} p_{jl}(|\omega_{jl}|) \tag{2}$$

  - Yuan and Lin (2007), Banerjee et al. (2008), Friedman et al. (2008), Rothman et al. (2008) studied the Lasso type estimator .

Log-likelihood Estimation

Another nature way is to estimate $\mathbf{\Omega}$ is the penalized likelihood approach.

- Log-likelihood function: $l(\mathbf{X}^(i), 1 \leq i \leq n|\mathbf{\Omega}) = -\mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) + \log|\mathbf{\Omega}|$
- Penalized likelihood estimator:

$$\hat{\mathbf{\Omega}}_L = \arg\min \mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) - \log|\mathbf{\Omega}| + \sum_{j \neq l} \lambda_{jl} p_{jl}(|\omega_{jl}|) \qquad (2)$$

  - Yuan and Lin (2007), Banerjee et al. (2008), Friedman et al. (2008), Rothman et al. (2008) studied the Lasso type estimator .
  - Friedman et al. (2008) proposed a fast algorithm, *glasso*, for Lasso type estimator.

Log-likelihood Estimation

Another nature way is to estimate $\mathbf{\Omega}$ is the penalized likelihood approach.

- Log-likelihood function: $l(\mathbf{X}^{(}i), 1 \le i \le n|\mathbf{\Omega}) = -\text{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) + \log|\mathbf{\Omega}|$
- Penalized likelihood estimator:

$$\hat{\mathbf{\Omega}}_L = \arg\min \text{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) - \log|\mathbf{\Omega}| + \sum_{j \ne l} \lambda_{jl} p_{jl}(|\omega_{jl}|) \qquad (2)$$

  - Yuan and Lin (2007), Banerjee et al. (2008), Friedman et al. (2008), Rothman et al. (2008) studied the Lasso type estimator .
  - Friedman et al. (2008) proposed a fast algorithm, *glasso*, for Lasso type estimator.
  - Yuan and Lin (2007) studied the non-negative garrote estimator.

Log-likelihood Estimation

Another nature way is to estimate $\mathbf{\Omega}$ is the penalized likelihood approach.

- Log-likelihood function: $l(\mathbf{X}^(i), 1 \leq i \leq n|\mathbf{\Omega}) = -\text{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) + \log|\mathbf{\Omega}|$
- Penalized likelihood estimator:

$$\hat{\mathbf{\Omega}}_L = \arg\min \text{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}_n) - \log|\mathbf{\Omega}| + \sum_{j \neq l} \lambda_{jl} p_{jl}(|\omega_{jl}|) \tag{2}$$

  - Yuan and Lin (2007), Banerjee et al. (2008), Friedman et al. (2008), Rothman et al. (2008) studied the Lasso type estimator .
  - Friedman et al. (2008) proposed a fast algorithm, *glasso*, for Lasso type estimator.
  - Yuan and Lin (2007) studied the non-negative garrote estimator.
  - Fan et al. (2009), Lam et al. (2009) studied the penalized likelihood estimator with the smoothly clipped absolute deviation (SCAD) penalty and the adaptive Lasso penalty.

Constrained ℓ_1 Minimization Approach

- Cai et al. (2011) performed a constrained l1 minimization approach to estimate sparse precision matrix (CLIME).

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_1 &= \arg\min \|\boldsymbol{\Omega}\|_1, \text{subject to:} \\ |\boldsymbol{\Sigma}_n^* \boldsymbol{\Omega} - \mathbf{I}|_\infty &\leq \tau_n \end{aligned} \tag{3}$$

Constrained ℓ_1 Minimization Approach

- Cai et al. (2011) performed a constrained l1 minimization approach to estimate sparse precision matrix (CLIME).

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_1 = \arg\min \|\boldsymbol{\Omega}\|_1, \text{subject to:} \\ |\boldsymbol{\Sigma}_n^* \boldsymbol{\Omega} - \mathbf{I}|_\infty \leq \tau_n \end{aligned} \tag{3}$$

- Problem 3 can be decomposed to p vector-minimization problem:

$$\hat{\boldsymbol{\omega}}_j = \arg\min |\boldsymbol{\omega}_j|_1 \text{ subject to} |\boldsymbol{\Sigma}_n^* \boldsymbol{\omega}_j - \mathbf{e}_j|_\infty \leq \tau_n, 1 \leq j \leq p.$$

Constrained ℓ_1 Minimization Approach

- Cai et al. (2011) performed a constrained l1 minimization approach to estimate sparse precision matrix (CLIME).

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_1 = \arg\min \|\boldsymbol{\Omega}\|_1, &\text{subject to:} \\ |\boldsymbol{\Sigma}_n^* \boldsymbol{\Omega} - \mathbf{I}|_\infty &\leq \tau_n \end{aligned} \tag{3}$$

- Problem 3 can be decomposed to p vector-minimization problem:

$$\hat{\boldsymbol{\omega}}_j = \arg\min |\boldsymbol{\omega}_j|_1 \text{ subject to} |\boldsymbol{\Sigma}_n^* \boldsymbol{\omega}_j - \mathbf{e}_j|_\infty \leq \tau_n, 1 \leq j \leq p.$$

- Cai et al. (2016) proposed adaptive constrained l1 minimization estimator (ACLIME), which achieved the optimal minimax rate of convergence.

Outline

Joint Estimation of Multiple Graphical Models

- Data are heterogeneous with p variables and G groups.

Joint Estimation of Multiple Graphical Models

- Data are heterogeneous with p variables and G groups.
  - $\mathbf{X}_i^{(g)} = (X_{i,1}^{(g)}, X_{i,2}^{(g)}, \cdots, X_{i,p}^{(g)}) \sim \mathcal{N}(0, \mathbf{\Sigma}^{(g)}), i = 1, \ldots, n_g; g = 1, \ldots, G.$

Joint Estimation of Multiple Graphical Models

- Data are heterogeneous with p variables and G groups.
  - $\mathbf{X}_i^{(g)} = (X_{i,1}^{(g)}, X_{i,2}^{(g)}, \cdots, X_{i,p}^{(g)}) \sim \mathcal{N}(0, \boldsymbol{\Sigma}^{(g)}), i = 1, \ldots, n_g; g = 1, \ldots, G.$
  - $\{\boldsymbol{\Omega}\} = \left\{\boldsymbol{\Omega}^{(g)} = (\boldsymbol{\Sigma}^{(g)})^{-1} | g = 1 \cdots G\right\}$

Joint Estimation of Multiple Graphical Models

Joint penalized likelihood of multiple precision matrices

$$\left\{\hat{\boldsymbol{\Omega}}\right\} = \underset{\{\boldsymbol{\Omega}\}}{\arg\min} \sum_{g=1}^{G} n_g \left[\mathrm{tr}(\boldsymbol{\Omega}^{(g)}\hat{\boldsymbol{\Sigma}}^{g}) - \log\left|\boldsymbol{\Omega}^{(g)}\right|\right] + P(\{\boldsymbol{\Omega}\}) \tag{4}$$

- Guo et al. (2011) employs a non-convex penalty called *hierarchical group penalty*: $P(\{\boldsymbol{\Omega}\}) = \lambda \sum_{j \neq l} \left(\sum_{g=1}^{G} |\omega_{jl}^{(g)}|\right)^{1/2}$

Joint penalized likelihood of multiple precision matrices

$$\left\{\hat{\boldsymbol{\Omega}}\right\} = \underset{\{\boldsymbol{\Omega}\}}{\arg\min} \sum_{g=1}^{G} n_g \left[\mathrm{tr}(\boldsymbol{\Omega}^{(g)}\hat{\boldsymbol{\Sigma}}^{g}) - \log\left|\boldsymbol{\Omega}^{(g)}\right|\right] + P(\{\boldsymbol{\Omega}\}) \qquad (4)$$

- Guo et al. (2011) employs a non-convex penalty called *hierarchical group penalty*: $P(\{\boldsymbol{\Omega}\}) = \lambda \sum_{j \neq l} \left(\sum_{g=1}^{G} |\omega_{jl}^{(g)}|\right)^{1/2}$
- Honorio and Samaras (2012) adopts a convex penalty, $P(\{\boldsymbol{\Omega}\}) = \lambda \sum_{j \neq l} |\omega_{jl}^{(1)}, \cdots, \omega_{jl}^{(G)}|_q$.

Joint Estimation of Multiple Graphical Models

Joint penalized likelihood of multiple precision matrices

$$\left\{\hat{\mathbf{\Omega}}\right\} = \underset{\{\mathbf{\Omega}\}}{\arg\min} \sum_{g=1}^{G} n_g \left[\mathrm{tr}(\mathbf{\Omega}^{(g)} \hat{\mathbf{\Sigma}}^{g}) - \log |\mathbf{\Omega}^{(g)}| \right] + P(\{\mathbf{\Omega}\}) \qquad (4)$$

- Guo et al. (2011) employs a non-convex penalty called *hierarchical group penalty*: $P(\{\mathbf{\Omega}\}) = \lambda \sum_{j \neq l} \left(\sum_{g=1}^{G} |\omega_{jl}^{(g)}| \right)^{1/2}$
- Honorio and Samaras (2012) adopts a convex penalty, $P(\{\mathbf{\Omega}\}) = \lambda \sum_{j \neq l} |\omega_{jl}^{(1)}, \cdots, \omega_{jl}^{(G)}|_q$.
- Danaher et al. (2014) considered a *fused lasso penalty*, $P(\{\mathbf{\Omega}\}) = \lambda_1 \sum_{j \neq l} \sum_{g=1}^{G} |\omega_{jl}^{(g)}| + \lambda_2 \sum_{g < g'} \sum_{jl} |\omega_{jl}^{(g)} - \omega_{jl}^{(g')}|$.

Joint Estimation of Multiple Graphical Models

- Lee and Liu (2015) decomposed $\{\boldsymbol{\Omega}\}$ into the common structure $M = \frac{1}{G}\sum_g \boldsymbol{\Omega}^{(g)}$ and the individual structure $R^{(g)} = \boldsymbol{\Omega}^{(g)} - M$, and applied constrained ℓ_1 minimization to estimate the parameters.

Outline

Time Varying Graphical Model

It is not unusual that the index of groups of samples have a order. A common situation is that the data are collected by time order.

- $\mathbf{X}(t) \sim (0, \mathbf{\Sigma}(t)), t = t_1, \ldots, t_n$.
  - Denote $\mathbf{X}^i = \mathbf{X}(t_i), i = 1, \cdots, n$.
- Dynamic graph: $G(t) = (V, E(t))$
  - $V = \{1, \cdots, p\}$.
  - $E(t) = \left\{(j, l) \in V^2 : \mathrm{Cov}\left[X_j(t), X_l(t) | X_k(t), k \neq j, l\right] \neq 0, j \neq l\right\}$.

Time Varying Graphical Model: Penalized likelihood estimation

Zhou (2010) developed a nonparametric framework for estimating time varying graphical model by kernel smoothing and ℓ_1 penalty.

$$\hat{\mathbf{\Omega}}(\tau) = \arg\min_{\mathbf{\Omega}} \left\{ \mathrm{tr}(\mathbf{\Omega}\hat{\mathbf{\Sigma}}(\tau)) - \log|\mathbf{\Omega}| + \lambda \|\mathbf{\Sigma}^{-}\|_1 \right\}$$
$$\text{where } \hat{\mathbf{\Omega}}(\tau) = \sum_i \omega_i^{\tau} \mathbf{X}^i (\mathbf{X}^i)', \text{and } \omega_i^{\tau} = \frac{K_h(t_i - \tau)}{\sum_{i'} K_h(t_{i'} - \tau)} \tag{5}$$

Time Varying Graphical Model And Varying Coefficient Model

- Dynamic partial correlation coefficient $\rho_{jl}(t) = -\omega_{jl}(t)/\sqrt{\omega_{jj}(t)\omega_{ll}(t)}$
- Dynamic neighborhood selection:

$$X_j(t) = \mathbf{X}'_{-j}(t)\boldsymbol{\beta}_j(t) + \varepsilon_j(t) = \sum_{l \neq j} X_l(t)\beta_{jl}(t) + \varepsilon_j(t). \qquad (6)$$

  - Varying coefficient model (Hastie and Tibshirani, 1993)
- Kolar et al. (2009, 2010); Kolar and Xing (2011) proposed a local linear regression approach with kernel ℓ_1 penalty to estimate the smoothly varying graph,

$$\hat{\boldsymbol{\beta}}_j(\tau) = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p-1}}{\arg\min} \sum_i (X_j^i - \sum_{l \neq j} X_l^i \beta_l)^2 \omega_i^\tau + \lambda |\boldsymbol{\beta}|_1 \qquad (7)$$

Time Varying Graphical Model And Varying Coefficient Model

- Kolar et al. (2009, 2010); Kolar and Xing (2011, 2012) proposed a combination of ℓ_1 and fused lasso penalty to estimate graph with jump.

$$
\begin{aligned}
&\left\{\hat{\boldsymbol{\beta}}_j(t_1), \cdots, \hat{\boldsymbol{\beta}}_j(t_n)\right\} = \\
&\underset{\boldsymbol{\beta}(t_i), i \le n}{\arg\min} \sum_i (X_j^i - \sum_{l \neq j} X_l^i \beta_l(t_i))^2 + \lambda_1 \sum_i |\boldsymbol{\beta}(t_i)|_1 \\
&+ \lambda_2 \sum_{i=2}^{n} |\boldsymbol{\beta}(t_i) - \boldsymbol{\beta}(t_{i-1})|_1
\end{aligned}
\tag{8}
$$

Time Varying Graphical Model And Functional Coefficient Model: Motivation

Limitation of aforementioned method:

Time Varying Graphical Model And Functional Coefficient Model: Motivation

Limitation of aforementioned method:

- Apply local linear regression on the model (7) can only get the estimation on input time point.

Use basis expansion to estimate $\beta_j(t)$ directly.

Time Varying Graphical Model And Functional Coefficient Model: Motivation

Limitation of aforementioned method:

- Apply local linear regression on the model (7) can only get the estimation on input time point.
- The assumption in the model (8) of the coefficient is piecewise constant is usually not realistic.

Use basis expansion to estimate $\beta_j(t)$ directly.

Functional Coefficient Model

- Assuming data are collected at $t_1, \cdots, t_n$, and at each time point t, we have n_t replicates (Huang et al, 2004).

$$\begin{aligned} X_j^r(t) &= \mathbf{X}_{-j}^r(t)'\boldsymbol{\beta}_j(t) + \varepsilon_j^r(t) \\ &= \sum_{l \neq j} X_l^r(t)\beta_{jl}(t) + \varepsilon_j^r(t), \end{aligned} \tag{9}$$

Functional Coefficient Model

- Assuming data are collected at $t_1, \cdots, t_n$, and at each time point t, we have n_t replicates (Huang et al, 2004).

$$
\begin{aligned}
X_j^r(t) &= \mathbf{X}_{-j}^r(t)'\boldsymbol{\beta}_j(t) + \varepsilon_j^r(t) \\
&= \sum_{l \neq j} X_l^r(t)\beta_{jl}(t) + \varepsilon_j^r(t),
\end{aligned} \tag{9}
$$

  - $\mathbf{X}_{-j}^r(t) = (X_l^r(t))_{l \neq j} \in \mathbb{R}^{(p-1)\times 1}, r = 1, \cdots, n_t$.

Functional Coefficient Model

- Assuming data are collected at $t_1, \cdots, t_n$, and at each time point t, we have n_t replicates (Huang et al, 2004).

$$\begin{aligned} X_j^r(t) &= \mathbf{X}_{-j}^r(t)'\boldsymbol{\beta}_j(t) + \varepsilon_j^r(t) \\ &= \sum_{l \neq j} X_l^r(t)\beta_{jl}(t) + \varepsilon_j^r(t), \end{aligned} \tag{9}$$

  - $\mathbf{X}_{-j}^r(t) = (X_l^r(t))_{l \neq j} \in \mathbb{R}^{(p-1)\times 1}, r = 1, \cdots, n_t$.
  - $t = t_1, \cdots, t_n, j = 1, \cdots, p$.

Functional Coefficient Model

- Assuming data are collected at $t_1, \cdots, t_n$, and at each time point t, we have n_t replicates (Huang et al, 2004).

$$\begin{aligned} X_j^r(t) &= \mathbf{X}_{-j}^r(t)'\boldsymbol{\beta}_j(t) + \varepsilon_j^r(t) \\ &= \sum_{l \neq j} X_l^r(t)\beta_{jl}(t) + \varepsilon_j^r(t), \end{aligned} \tag{9}$$

  - $\mathbf{X}_{-j}^r(t) = (X_l^r(t))_{l \neq j} \in \mathbb{R}^{(p-1)\times 1}, r = 1, \cdots, n_t$.
  - $t = t_1, \cdots, t_n, j = 1, \cdots, p$.

- For each t, let $\mathbf{X}_j(t) = (X_j^1(t), \cdots, X_j^{n_t}(t))^T \in \mathbb{R}^{n_t \times 1}$, then

$$\begin{aligned} \mathbf{X}_j(t) &= \mathbf{X}_{-j}(t)'\boldsymbol{\beta}_j(t) + \boldsymbol{\varepsilon}_j(t) \\ &= \sum_{l \neq j} \mathbf{X}_l(t)\beta_{jl}(t) + \boldsymbol{\varepsilon}_j(t), \end{aligned} \tag{10}$$

Functional Coefficient Model

- For each functional coefficient $\beta_{jl}(t)$, we consider the basis expansion $\mathbf{B}_{jl}(t) = (B_{jl1}(t), \cdots, B_{jlk_{jl}}(t))$:

$$\beta_{jl}(t) = \sum_{s=1}^{k_{jl}} B_{jls}(t)\gamma_{jls} + e_{jl}(t) = \mathbf{B}_{jl}(t)\gamma_{jl} + e_{jl}(t)$$

- $\boldsymbol{\beta}_j(t) = \mathbf{B}(t)\boldsymbol{\gamma}_j + \mathbf{e}_j(t) = (\beta_{jl}|j \neq l) \in \mathbb{R}^{p-1}$
  - $\mathbf{B}(t) = \text{diag}\{\mathbf{B}_{jl}(t)\} \in \mathbb{R}^{(p-1)\times\sum_{l\neq j} k_{jl}}$
  - $\boldsymbol{\gamma}_j = (\gamma_{jl})_{l\neq j} \in \mathbb{R}^{\sum_{l\neq j} k_{jl}\times 1}$
  - $\mathbf{e}(t) = (\mathbf{e}_{jl}|j \neq l) \in \mathbb{R}^{p-1}$

Functional Coefficient Model

Functional Coefficient Model

Thus Equation (9) can be represented as

Functional Coefficient Model

Thus Equation (9) can be represented as

$$\begin{aligned}
\mathbf{X}_j(t) &= \mathbf{X}_{-j}(t)\mathbf{B}_j(t)\gamma_j + \tilde{\varepsilon}(t) \\
&= \mathbf{U}(t)\gamma_j + \tilde{\varepsilon}(t), \\
\text{where } \mathbf{U}(t) &= \mathbf{X}_{-j}(t)\mathbf{B}_j(t) \in \mathbb{R}^{\sum_{t=1}^{n} n_t \times \sum_{j\neq i} k_{jl}}, \\
\tilde{\varepsilon}(t) &= \mathbf{X}_{-j}(t)\mathbf{e}_j(t) + \varepsilon(t) \\
&= \sum_{l\neq j} \mathbf{X}_l(t) e_{jl}(t) + \varepsilon(t), \\
t &= t_1, \cdots, t_n, j = 1, \cdots, p.
\end{aligned} \tag{11}$$

Functional Coefficient Model

Thus Equation (9) can be represented as

$$
\begin{aligned}
\mathbf{X}_j(t) &= \mathbf{X}_{-j}(t)\mathbf{B}_j(t)\gamma_j + \tilde{\varepsilon}(t) \\
&= \mathbf{U}(t)\gamma_j + \tilde{\varepsilon}(t), \\
\text{where } \mathbf{U}(t) &= \mathbf{X}_{-j}(t)\mathbf{B}_j(t) \in \mathbb{R}^{\sum_{t=1}^{n} n_t \times \sum_{j \neq i} k_{jl}}, \\
\tilde{\varepsilon}(t) &= \mathbf{X}_{-j}(t)\mathbf{e}_j(t) + \varepsilon(t) \\
&= \sum_{l \neq j} \mathbf{X}_l(t) e_{jl}(t) + \varepsilon(t), \\
t &= t_1, \cdots, t_n, j = 1, \cdots, p.
\end{aligned}
\tag{11}
$$

As seen in Equation (11), our model is quite flexible since the basis of each functional coefficient can be different.

Functional Coefficient Model

Combing the data from $t_1, \cdots, t_n$, we can get

$$\begin{aligned} \mathbf{X}_j &= \mathbf{U}_j \gamma_j + \tilde{\varepsilon}_j, j = 1, \cdots, p. \\ \text{where } \mathbf{X}_j &= (X_j(t_1)', \cdots, X_j(t_n)')' \in \mathbb{R}^{\sum_{t=1}^n n_t \times 1} \\ \mathbf{U}_j &= (\mathbf{U}_j(t_1)', \cdots, \mathbf{U}_j(t_n)')' \in \mathbb{R}^{\sum_{t=1}^n n_t \times \sum_{j \neq i} k_{jl}} \\ \tilde{\varepsilon}_j &= (\tilde{\varepsilon}_j(t_1), \cdots, \tilde{\varepsilon}_j(t_n))^T \in \mathbb{R}^{\sum_{t=1}^n n_t \times 1} \end{aligned} \tag{12}$$

For each j, least square of γ_j in the model 12 is

$$\begin{aligned} l(\gamma_j) &= (\mathbf{X}_j - \mathbf{U}_j \gamma_j)' \mathbf{W} (\mathbf{X}_j - \mathbf{U}_j \gamma_j) \\ &= \sum_{i=1}^n (\mathbf{X}_j(t_i) - \mathbf{U}_j(t_i) \gamma_j)^2 w_i \\ &= \sum_{i=1}^n (\mathbf{X}_j(t_i) - \mathbf{X}_{-j}(t_i) \beta_j(t_i))^2 w_i \end{aligned}$$

Functional Coefficient Model: Control Derivatives Sparsity

In Model (12), we want to estimate a sparse graph and interpretable coefficient functions.

Functional Coefficient Model: Control Derivatives Sparsity

In Model (12), we want to estimate a sparse graph and interpretable coefficient functions.

- For each j and $l \neq j$, put ℓ_1 penalty on $\beta_{jl}^{(m)} = \frac{\mathrm{d}^{\mathrm{m}}}{\mathrm{d}t^m}\beta_{jl}(t) \approx \frac{\mathrm{d}^{\mathrm{m}}}{\mathrm{d}t^m}\mathbf{B}_{jl}(t)\boldsymbol{\gamma}_{jl}$ for some m.

Functional Coefficient Model: Control Derivatives Sparsity

In Model (12), we want to estimate a sparse graph and interpretable coefficient functions.

- For each j and $l \neq j$, put ℓ_1 penalty on $\beta_{jl}^{(m)} = \frac{\mathrm{d^m}}{\mathrm{d}t^m}\beta_{jl}(t) \approx \frac{\mathrm{d^m}}{\mathrm{d}t^m}\mathbf{B}_{jl}(t)\boldsymbol{\gamma}_{jl}$ for some m.
- Assume $\beta_{jl}^{(0)}(t) = 0$ and $\beta_{jl}^{(2)}(t) = 0$ in large area, then $\beta_{jl}(t)$ is zero in many region and linear in the left regions.

Functional Coefficient Model: Control Derivatives Sparsity

In Model (12), we want to estimate a sparse graph and interpretable coefficient functions.

- For each j and $l \neq j$, put ℓ_1 penalty on $\beta_{jl}^{(m)} = \frac{\mathrm{d^m}}{\mathrm{d}t^m}\beta_{jl}(t) \approx \frac{\mathrm{d^m}}{\mathrm{d}t^m}\mathbf{B}_{jl}(t)\boldsymbol{\gamma}_{jl}$ for some m.
- Assume $\beta_{jl}^{(0)}(t) = 0$ and $\beta_{jl}^{(2)}(t) = 0$ in large area, then $\beta_{jl}(t)$ is zero in many region and linear in the left regions.
- Let $\mathbf{A}^{(m)}(t) = \frac{\mathrm{d^m}}{\mathrm{d}t^m}\mathbf{B}(t) \in \mathbb{R}^{(p-1)\times\sum_{l\neq j} k_{jl}}$ and $\mathbf{A}^{(m)} = (\mathbf{A}^{(m)}(t_1)', \cdots, \mathbf{A}^{(m)}(t_n)')' \in \mathbb{R}^{n(p-1)\times\sum_{l\neq j} k_{jl}}$

Functional Coefficient Model: Control Derivatives Sparsity

In Model (12), we want to estimate a sparse graph and interpretable coefficient functions.

- For each j and $l \neq j$, put ℓ_1 penalty on $\beta_{jl}^{(m)} = \frac{\mathrm{d^m}}{\mathrm{d}t^m}\beta_{jl}(t) \approx \frac{\mathrm{d^m}}{\mathrm{d}t^m}\mathbf{B}_{jl}(t)\boldsymbol{\gamma}_{jl}$ for some m.
- Assume $\beta_{jl}^{(0)}(t) = 0$ and $\beta_{jl}^{(2)}(t) = 0$ in large area, then $\beta_{jl}(t)$ is zero in many region and linear in the left regions.
- Let $\mathbf{A}^{(m)}(t) = \frac{\mathrm{d^m}}{\mathrm{d}t^m}\mathbf{B}(t) \in \mathbb{R}^{(p-1)\times\sum_{l\neq j} k_{jl}}$ and $\mathbf{A}^{(m)} = (\mathbf{A}^{(m)}(t_1)', \cdots, \mathbf{A}^{(m)}(t_n)')' \in \mathbb{R}^{n(p-1)\times\sum_{l\neq j} k_{jl}}$

Functional Coefficient Model: Control Derivatives Sparsity

In Model (12), we want to estimate a sparse graph and interpretable coefficient functions.

- For each j and $l \neq j$, put ℓ_1 penalty on $\beta_{jl}^{(m)} = \frac{\mathrm{d}^{\mathrm{m}}}{\mathrm{d}t^m}\beta_{jl}(t) \approx \frac{\mathrm{d}^{\mathrm{m}}}{\mathrm{d}t^m}\mathbf{B}_{jl}(t)\boldsymbol{\gamma}_{jl}$ for some m.
- Assume $\beta_{jl}^{(0)}(t) = 0$ and $\beta_{jl}^{(2)}(t) = 0$ in large area, then $\beta_{jl}(t)$ is zero in many region and linear in the left regions.
- Let $\mathbf{A}^{(m)}(t) = \frac{\mathrm{d}^{\mathrm{m}}}{\mathrm{d}t^m}\mathbf{B}(t) \in \mathbb{R}^{(p-1)\times\sum_{l\neq j} k_{jl}}$ and $\mathbf{A}^{(m)} = (\mathbf{A}^{(m)}(t_1)', \cdots, \mathbf{A}^{(m)}(t_n)')' \in \mathbb{R}^{n(p-1)\times\sum_{l\neq j} k_{jl}}$

The penalty for Model (12) is

$$
\begin{aligned}
p(\gamma_j) &= \lambda_0|\mathbf{A}^{(0)}\gamma_j|_1 + \lambda_m|\mathbf{A}^{(m)}\gamma_j|_1 \\
&= \sum_{i=1}^{n}\lambda_0|\boldsymbol{\beta}_j^{(0)}(t_i)|_1 + \sum_{i=1}^{n}\lambda_2|\boldsymbol{\beta}_j^{(m)}(t_i)|_1
\end{aligned}
$$

Functional Coefficient Model: Optimization Problem

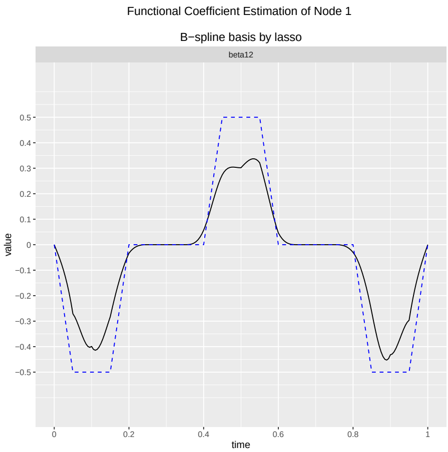
The optimization problem for Model (12) with sparse coefficient derivatives:

$$\begin{aligned}
\mathcal{J}(\gamma_j) &= l(\gamma_j) + p(\gamma_j) \\
&= (\mathbf{X}_j - \mathbf{U}_j\gamma_j)'\mathbf{W}(\mathbf{X}_j - \mathbf{U}_j\gamma_j) + \lambda_0|\mathbf{A}^{(0)}\gamma_j|_1 + \lambda_m|\mathbf{A}^{(m)}\gamma_j|_1 \\
&= \sum_{i=1}^{n}(\mathbf{X}_j(t_i) - \mathbf{X}_{-j}(t_i)\boldsymbol{\beta}_j(t_i))^2 w_i + \\
&\sum_{i=1}^{n}\lambda_0|\boldsymbol{\beta}_j^{(0)}(t_i)|_1 + \sum_{i=1}^{n}\lambda_m|\boldsymbol{\beta}_j^{(m)}(t_i)|_1
\end{aligned} \tag{13}$$

This is a generalized lasso optimization problem (Tibshirani, 2011).

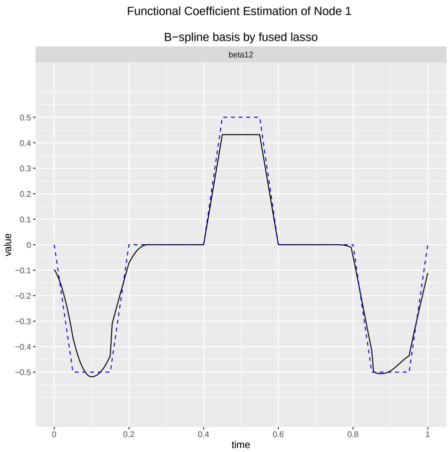
Outline

Simulation



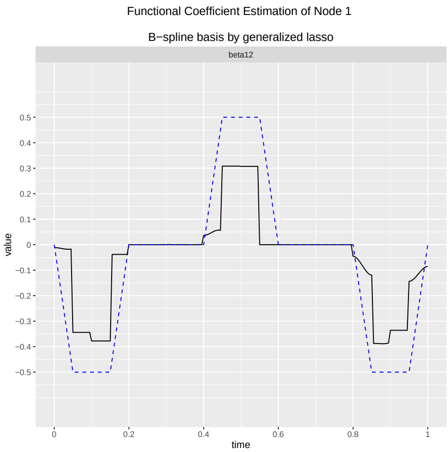
Functional Coefficient Estimation of Node 1
B−spline basis by lasso
beta12
value
time
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
0
0.2
0.4
0.6
0.8
1

Simulation



Functional Coefficient Estimation of Node 1
B−spline basis by fused lasso
beta12
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
value
0
0.2
0.4
0.6
0.8
1
time

Simulation



Functional Coefficient Estimation of Node 1
B−spline basis by generalized lasso
beta12
value
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
0
0.2
0.4
0.6
0.8
1
time