Statistical Learning and Data Science

Xinwei Deng
xdeng@vt.edu
Department of Statistics

Gaussian Graphical Model Estimation

- Graphical Model via Covariance Matrix Estimation
  - Modified Cholesky decomposition approach.
  - Structured graphical model estimation.
  - Nonparametric approach for graphical model estimation.
  - Multiple graphical models.

Why Estimating Σ^{-1}: Gaussian Graphical Model

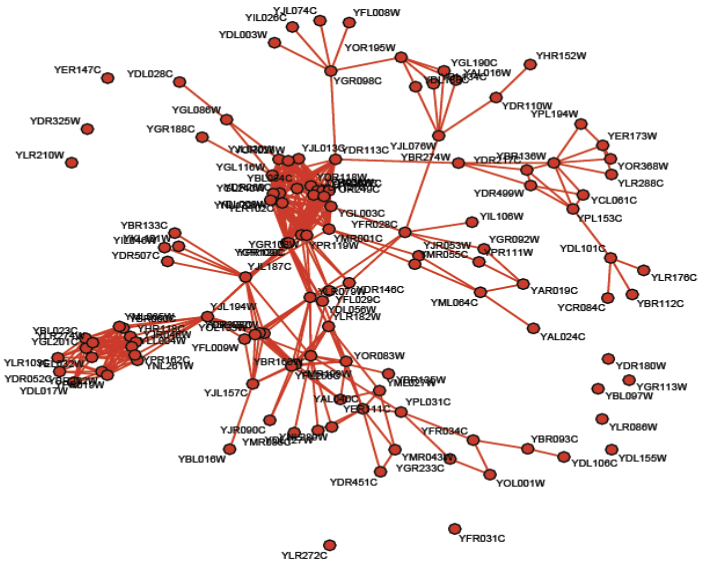
y_1	y_2	y_3	y_4	$\cdots$	y_q

$$\mathbf{\Omega} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \vdots & \ddots & \vdots & \vdots \\ c_{q1} & c_{q2} & \cdots & c_{qq} \end{pmatrix}$$

concentration matrix $\mathbf{\Sigma}^{-1} \equiv \mathbf{\Omega} = (c_{ij})$

Data $\mathbf{Y} = (y_{ij})$ is an $n \times q$ matrix

- describe the **conditional dependency** among variables: if $c_{ij} = 0$ zero, then variables i and j are conditionally independent given the other variables.



The Likelihood Viewpoint

- The multi-response model is simplified as

$$\boldsymbol{y} = \mu + \epsilon, \epsilon \sim N(0, \boldsymbol{\Sigma}).$$

- With data $\boldsymbol{y}_i, i = 1, \ldots, n$, it is equivalent to

$$\boldsymbol{y}_i \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), i = 1, \ldots, n.$$

- To estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the log-likelihood function becomes

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log \left\{ \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} |\boldsymbol{\Sigma}^{-1}|^{1/2} \exp(-\frac{(\boldsymbol{y}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu})}{2}) \right\} \\ &\propto \frac{n}{2} \log |\boldsymbol{\Sigma}^{-1}| - \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}) \\ &\propto \log |\boldsymbol{\Sigma}^{-1}| - tr[\boldsymbol{\Sigma}^{-1} \boldsymbol{S}], \end{aligned}$$

where

$$\boldsymbol{S} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^T.$$

Estimating $\boldsymbol{\Sigma}^{-1}$

- Based on $L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \frac{n}{2} \log |\boldsymbol{\Sigma}^{-1}| - \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu})$, it is easy to obtain the estimate of μ as

$$\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{y}} = \frac{1}{2} \sum_{i=1}^{n} \boldsymbol{y}_i.$$

- To estimate $\boldsymbol{\Sigma}$, take the first derivative with respect to $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{\Omega}} \{ - \log |\boldsymbol{\Omega}| + tr[\boldsymbol{\Omega} \boldsymbol{S}] \} \\ &= -\boldsymbol{\Sigma} + \boldsymbol{S} = 0 \end{aligned}$$

Therefore, the estimate of $\boldsymbol{\Sigma}$ is

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T.$$

Graphical Lasso Approach

- Denote $\mathbf{\Sigma}^{-1} = (c_{ij})$. Note that $c_{ij} = 0$ implies that Y_i and Y_j are independent conditional on the other variables.
- Graphical Lasso (Yuan and Lin, 2007; Friedman et al., 2009) is to estimate $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ by

$$\min_{\mathbf{\Omega}} -\log|\mathbf{\Sigma}^{-1}| + tr[\mathbf{\Sigma}^{-1}\boldsymbol{S}] + \lambda\|\mathbf{\Sigma}^{-1}\|_1,$$

  where λ is a tuning parameter. Here $\|\cdot\|_1$ is defined as $\|\mathbf{\Sigma}^{-1}\|_1 = \sum_{i\neq j}|c_{ij}|$.
- Graphical Lasso can be obtained by recursively solving and updating the lasso regression (R package *glasso*), or coordinate descent method (Witten et al., 2011).

Modified Cholesky Decomposition (MCD)

- From matrix algebra, various matrix decompositions are used for computing matrix inversion, determinant, etc.
- The $\boldsymbol{\Sigma}^{-1}$ can be obtained from the modified Cholesky decomposition.
  - Advantage: the estimation is transformed into a series of regressions.
- Let us define

$$\begin{aligned} X_j &= \sum_{t=1}^{j-1} a_{jt} X_t + \epsilon_j \\ &= \boldsymbol{Z}_j^T \boldsymbol{a}_j + \epsilon_j, \end{aligned}$$

where $\boldsymbol{Z}_j = (X_1, \ldots, X_{j-1})'$, and $\boldsymbol{a}_j = (a_{j1}, \ldots, a_{j,j-1})'$ are corresponding regression coefficients. Then the vector $\boldsymbol{a}_j$ can be obtained from

$$\boldsymbol{a}_j = (Cov(\boldsymbol{Z}_j, \boldsymbol{Z}_j))^{-1} Cov(X_j, \boldsymbol{Z}_j).$$

MCD Approach for $\boldsymbol{\Sigma}^{-1}$

- The variance of ϵ_j is defined by

$$d_j^2 = Var(X_j - \hat{X}_j) = Var(\epsilon_j).$$

- Then we can compose a lower triangle matrix $\boldsymbol{A}$ such that

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ a_{21} & 0 & 0 & \dots & 0 \\ a_{31} & a_{32} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{p1} & a_{p2} & \dots & a_{p,p-1} & 0 \end{pmatrix},$$

  which contains all regression coefficients a_{ij}.

- Let $\boldsymbol{D} = diag(d_1^2, \dots, d_p^2)$ as a diagonal matrix. The modified Cholesky decomposition implies that

$$\begin{aligned} \boldsymbol{\Sigma}^{-1} &= (\boldsymbol{I} - \boldsymbol{A})' \boldsymbol{D}^{-1} (\boldsymbol{I} - \boldsymbol{A}) \\ &= \boldsymbol{T}' \boldsymbol{D}^{-1} \boldsymbol{T}. \end{aligned} \tag{1}$$

Computational Flexibility

- The straightforward estimate $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{D}}$ can be obtained from the least squares estimates of the coefficients a_{ij} and the corresponding residual variances d_{ij}.
- The above formulation can be viewed as a multiple response regression under a given order of random variables.

$$
\begin{aligned}
x_1 &= \epsilon_1, \quad \epsilon_1 \sim N(0, d_1^2); \\
x_2 &= a_{21}x_1 + \epsilon_2, \quad \epsilon_2 \sim N(0, d_2^2); \\
x_3 &= a_{31}x_1 + a_{32}x_2 + \epsilon_3, \quad \epsilon_3 \sim N(0, d_3^2); \\
&\vdots \\
x_p &= a_{p1}x_1 + a_{p2}x_2 + \cdots, + a_{p,p-1}x_{p-1} + \epsilon_p, \quad \epsilon_p \sim N(0, d_p^2).
\end{aligned}
$$

- The penalized regression can be applied to encourage the sparsity on $\boldsymbol{T}$, consequently leading to the sparsity of $\boldsymbol{\Sigma}^{-1}$.

Some Remarks

- The assumption of modified Cholesky decomposition is that ϵ_j is independent.
- It can be used to get a banded estimate of $\boldsymbol{\Sigma}^{-1}$.
- Can such an approach be extended for the estimation of sparse $\boldsymbol{\Sigma}$?
- Recall that the negative log-likelihood function of $\boldsymbol{\Sigma}$ given the sample, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, is proportional to

$$L_n(\boldsymbol{\Sigma}) = -\log|\boldsymbol{\Sigma}^{-1}| + \mathrm{tr}[\boldsymbol{\Sigma}^{-1}\boldsymbol{S}],$$

where $\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{'}/n$ be the sample covariance matrix.

MCD for Estimating $\boldsymbol{\Sigma}$

- The modified Cholesky decomposition (MCD) can be used to estimate $\boldsymbol{\Sigma}$ from a latent variable regression model.
- Define $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)'$. Note that $\boldsymbol{\epsilon} = \boldsymbol{X} - \hat{\boldsymbol{X}}$, and hence,

$$\begin{aligned} Var(\boldsymbol{\epsilon}) &= Var(\boldsymbol{X} - \hat{\boldsymbol{X}}) = Var(\boldsymbol{T}\boldsymbol{X}) \\ &= \boldsymbol{D} = \boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}'. \end{aligned}$$

Therefore, we can write $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{\epsilon}$, which leads to

$$\begin{aligned} Var(\boldsymbol{X}) &= Var(\boldsymbol{L}\boldsymbol{\epsilon}) \\ &= \boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}'. \end{aligned} \tag{2}$$

MCD for Estimating $\boldsymbol{\Sigma}$ (Con't)

- It implies a modified Cholesky decomposition for $\boldsymbol{\Sigma}$ itself.
- we can get an interpretation of latent variable regressions.
  - The $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)'$ is unobserved.
  - The $\boldsymbol{L}$ is lower triangular, and each variable X_j regresses on the previous latent variables $\epsilon_1, \ldots, \epsilon_{j-1}$.
- It gives a sequence of regression. For $j = 2, \ldots, p$, we get

$$\begin{aligned} X_j &= \sum_{t=1}^{j-1} l_{jt}\epsilon_t + \epsilon_j \\ &= \boldsymbol{U}_j^T \boldsymbol{l}_j + \epsilon_j, \end{aligned}$$

where $\boldsymbol{U}_j = (\epsilon_1, \ldots, \epsilon_{j-1})'$, and $\boldsymbol{l}_j = (l_{j1}, \ldots, l_{j,j-1})'$ is the vector of regression coefficients.

Sparse Estimate for $\boldsymbol{\Sigma}$

- With data matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$, define its jth column to be $\boldsymbol{x}^{(j)}$. The residual for $\boldsymbol{x}^{(j)}$ is defined as $\boldsymbol{e}^{(j)}$, and $\boldsymbol{Z}^{(j)} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{j-1})$.
- Set $\boldsymbol{e}^{(1)} = \boldsymbol{x}^{(1)}$. we obtain the estimate of $\boldsymbol{l}_j$ by least squares

$$\hat{\boldsymbol{l}}_j = \arg\min_{\boldsymbol{l}_j} \|\boldsymbol{x}^{(j)} - \boldsymbol{Z}^{(j)}\boldsymbol{l}_j\|_2^2, \ \ j = 2, \ldots, p,$$

  where $\boldsymbol{e}^{(j)} = \boldsymbol{x}^{(j)} - \boldsymbol{Z}^{(j)}\boldsymbol{l}_j$ constructing the residuals used in $\boldsymbol{Z}^{(j+1)}$. Get $\hat{\boldsymbol{D}} = diag(\hat{d}_1, \ldots, \hat{d}_p)$ where $\hat{d}_j = var(\boldsymbol{e}^{(j)})$ is sample variance of residuals.
- Encourage the sparsity on $\hat{\boldsymbol{L}}$ by using the penalized regression,

$$\hat{\boldsymbol{l}}_j = \arg\min_{\boldsymbol{l}_j} \|\boldsymbol{x}^{(j)} - \boldsymbol{Z}^{(j)}\boldsymbol{l}_j\|_2^2 + \lambda\|\boldsymbol{l}_j\|_1, \ \ j = 2, \ldots, p.$$

- Then $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{L}}\hat{\boldsymbol{D}}\hat{\boldsymbol{L}}'$ will be a sparse covariance matrix estimate.

Partial Correlation Approach

- Assume that $\boldsymbol{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{X} = (X_1, \ldots, X_p)$. Denote $\boldsymbol{\Sigma}^{-1} = (c_{ij})$.
- **Lemma.** *Suppose X_i is expressed as*

$$X_i = \sum_{j \neq i} \beta_{ij} X_j + \epsilon_i,$$

Then we can have

$$\beta_{ij} = -c_{ij}/c_{ii}, \text{ and } var(\epsilon_i) = 1/c_{ii}.$$

Moreover, $cov(\epsilon_i, \epsilon_j) = c_{ij}/(c_{ii}c_{jj})$.

- The $cor(\epsilon_i, \epsilon_j) = \sqrt{c_{ij}/(c_{ii}c_{jj})}$ is the **partial correlation** between X_i and X_j.
- Now one can consider to estimate $\boldsymbol{\Sigma}^{-1} = (c_{ij})$ through a joint regression effort.
- Remark: connection to the modified Cholesky decomposition approach?

Partial Correlation Approach: Estimation

- Let $\boldsymbol{X}_{(k)}$ be the kth column of the data matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$.
- The joint regression objective function is rewritten as

$$\sum_{i=1}^{p} w_i \|\boldsymbol{X}_{(i)} - \sum_{j \neq i} \beta_{ij} \boldsymbol{X}_{(j)}\|^2$$

$$= \sum_{i=1}^{p} \frac{1}{c_{ii}} \|c_{ii} \boldsymbol{X}_{(i)} + \sum_{j \neq i} c_{ij} \boldsymbol{X}_{(j)}\|^2,$$

  where $w_i = 1/var(\epsilon) = c_{ii}$.
- *Individual penalized regression for estimation* VS *Joint penalized regressions for estimation*
- Consider regularization on c_{ij} can lead to the penalized loss function as

$$\min \sum_{i=1}^{p} \frac{1}{c_{ii}} \|c_{ii} \boldsymbol{X}_{(i)} + \sum_{j \neq i} c_{ij} \boldsymbol{X}_{(j)}\|^2 + \lambda \sum_{i \neq j} |c_{ij}|.$$

CLIME Approach

- The partial correlation approach, Glasso method, MCD approaches are mostly likelihood based methods.
- CLIME (constrained l_1-minimization for inverse matrix estimation, Cai et al., 2011) is to estimate $\mathbf{\Sigma}^{-1} = (c_{ij})$ by solving

$$\min_{\mathbf{\Sigma}^{-1}} \sum_{i,j} |c_{ij}|$$
$$\text{s.t. } |\boldsymbol{S}\mathbf{\Sigma}^{-1} - \boldsymbol{I}|_\infty \leq \lambda,$$

- The motivation is from the score function of log-likelihood $-\log|\mathbf{\Sigma}^{-1}| + tr[\mathbf{\Sigma}^{-1}\boldsymbol{S}]$.
- Such an estimate may not be symmetric, also may not be positive definite.
- The advantages are relied on few assumptions and computation is convenient.

Computation for CLIME Approach

- Note that it is a convex optimization. We can further decompose it into p minimization problems with respect to vectors.
- Denote by $\boldsymbol{e}_j$ the jth column of $\boldsymbol{I}$ and $\boldsymbol{\beta}_j$ the jth column of $\boldsymbol{\Sigma}^{-1}$.
- Then solve a series of minimizations for $j = 1, \ldots, p$,

$$\min_{\boldsymbol{\beta}_j} \|\boldsymbol{\beta}_j\|_1$$
$$\text{s.t. } |\boldsymbol{S}\boldsymbol{\beta}_j - \boldsymbol{e}_j|_\infty \leq \lambda.$$

- The above optimization can be solved by linear programming through the linear relaxation technique.

Computation for CLIME Approach (Con't)

- Because of the convex optimization, it can be decomposed into p minimization problems with respect to vectors.
- Denote $\boldsymbol{r}_j'$ to be the jth row of $\boldsymbol{S}$. The objective is

$$\min_{\boldsymbol{\beta}_j} \|\boldsymbol{\beta}_j\|_1 \text{ s.t. } |\boldsymbol{R}\boldsymbol{\beta}_j - \boldsymbol{e}_j|_\infty \leq \lambda,$$

- We consider a relaxation of the optimization for $\boldsymbol{\beta}_j$ as

$$\begin{aligned} &\min \sum_{k=j}^{p} t_k \\ &\text{s.t. } -t_k \leq \beta_j^{(k)} \leq t_k, \ k = j, \ldots, p, \\ &\quad -\lambda \leq \boldsymbol{r}_j' \boldsymbol{\beta}_j - 1 \leq \lambda, \\ &\quad -\lambda \leq \boldsymbol{r}_k' \boldsymbol{\beta}_j \leq \lambda, \ k \neq j. \end{aligned}$$

- Remark: this linear relaxation technique is also used in Candès and Tao (2007) for the Dantzig selector problem.

Factor Model Approach

- Denote $\boldsymbol{x} = (x_1, \ldots, x_p)^T$, and $\boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$.
- The factor model is to consider

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{z} + \boldsymbol{\epsilon},$$

  where $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p)^T$, and the factors (latent variables) are $\boldsymbol{z} = (z_1, \ldots, z_m)^T$.
- Assume $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = diag\{\sigma_1^2, \ldots, \sigma_p^2\}$. Then the joint distribution of $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ is

$$\begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{x} \end{pmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{B}' \\ \boldsymbol{B} & \boldsymbol{B}\boldsymbol{B}' + \boldsymbol{\Psi} \end{bmatrix} \right).$$

- The underlying assumption is that $cov(\boldsymbol{z}, \boldsymbol{\epsilon}) = \mathbf{0}$.

Factor Model Approach for $\boldsymbol{\Sigma}$

- It is easy to see that $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}' + \boldsymbol{\Psi}$.
- The sparse (parsimonious) estimation of $\boldsymbol{B}$ can imply the sparsity of $\boldsymbol{\Sigma}$.
- With the data matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$, and the latent data matrix $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T$, the joint log-likelihood $\ell(\boldsymbol{B}, \boldsymbol{\Psi})$ can be written as

$$-\frac{n}{2}\log|\boldsymbol{\Psi}| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{z}_i)'\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{z}_i) - \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{z}_i^T\boldsymbol{z}_i,$$

  up to some constant.
- Note that the sparsity is to impose on $\boldsymbol{B}$, which is to consider the format of

$$\min -\frac{2}{n}\ell(\boldsymbol{B}, \boldsymbol{\Psi}) + \lambda J(\boldsymbol{B}),$$

  where $J(\boldsymbol{B})$ is a penalty function.

Factor Model Approach: Computation

- However, the latent data points z_i is not observed. The EM-type algorithm can be considered for facilitating the computation.
- From the joint likelihood of $(\boldsymbol{x}, \boldsymbol{z})$, it is known that

$$\begin{aligned} E(\boldsymbol{z}|\boldsymbol{x}) &= \boldsymbol{B}'(\boldsymbol{B}\boldsymbol{B}' + \boldsymbol{\Psi})^{-1}\boldsymbol{x} \\ &= (\boldsymbol{B}'\boldsymbol{\Psi}^{-1}\boldsymbol{B} + \boldsymbol{I}_m)^{-1}\boldsymbol{B}'\boldsymbol{\Psi}^{-1}\boldsymbol{x} \\ Var(\boldsymbol{z}|\boldsymbol{x}) &= \boldsymbol{I}_m - \boldsymbol{B}'(\boldsymbol{B}\boldsymbol{B}' + \boldsymbol{\Psi})^{-1}\boldsymbol{B} \\ &= \boldsymbol{I}_m - \boldsymbol{B}'\boldsymbol{\Psi}^{-1}\boldsymbol{B}(\boldsymbol{B}'\boldsymbol{\Psi}^{-1}\boldsymbol{B} + \boldsymbol{I}_m)^{-1} \end{aligned}$$

- For the M-step, it is to solve

$$\min_{\boldsymbol{B}} -\frac{2}{n} E\left[\ell(\boldsymbol{B}, \boldsymbol{\Psi})\right] + \lambda J(\boldsymbol{B})$$

$$\Leftrightarrow \min_{\boldsymbol{B}} \frac{1}{n}\sum_{i=1}^{n} E\left[(\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{z}_i)'\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{z}_i)\right] + \lambda J(\boldsymbol{B})$$

Factor Model Approach: Computation (Con't)

- For the estimation of $\boldsymbol{\Psi}$ in M-Step, it is easy to obtain that

$$\hat{\boldsymbol{\Psi}} = \text{diag}\left\{\frac{1}{n}\sum_{i=1}^{n} E\left[(\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{z}_i)(\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{z}_i)'\right]\right\}$$

- When $\lambda = 0$, the solution of $\boldsymbol{B}$ becomes the LS estimator that is

$$\hat{\boldsymbol{B}} = \left\{\sum_{i=1}^{n} \boldsymbol{x}_i E(\boldsymbol{z}_i')\right\}\left\{\sum_{i=1}^{n} E(\boldsymbol{z}_i \boldsymbol{z}_i')\right\}^{-1}$$

- Note that the above formulation requires to compute $E(\boldsymbol{z}\boldsymbol{z}'|\boldsymbol{x}) = E(\boldsymbol{z}|\boldsymbol{x})E(\boldsymbol{z}|\boldsymbol{x})' + Var(\boldsymbol{z}|\boldsymbol{x})$.
- Remark: Some constraints on the model identification may be needed in the estimation of $\boldsymbol{B}$.

Application: Portfolio Optimization

- Consider n stocks whose returns are distributed with mean μ and covariance matrix $\mathbf{\Sigma}$.
- Markowitz defines the portfolio selection problem as:

$$\min_{w} w'\Sigma w$$

$$\text{s.t. } w'1 = 1, w'\mu = q$$

where q is the required expected return.

- Using the Lagrange multipliers, the solution is:

$$w = \frac{c - qb}{ac - b^2}\Sigma^{-1}1 + \frac{qa - b}{ac - b^2}\Sigma^{-1}\mu$$

where $a = 1'\Sigma^{-1}1, b = 1'\Sigma^{-1}\mu, c = \mu'\Sigma^{-1}\mu$

Log Covariance Matrix Estimation

- Motivation:
  - In univariate case: we can model $\log(\sigma^2)$.
  - In multivariate case, can we still do something as

$$\log(\boldsymbol{\Sigma})$$

Background

- Covariance matrix estimation is important in multivariate analysis and many statistical applications.
- Suppose $x_1, \ldots, x_n$ are i.i.d. p-dimensional random vectors $\sim N(0, \Sigma)$. Let $S = \sum_{i=1}^{n} x_i x_i^{'}/n$ be the sample covariance matrix. The negative log-likelihood function is proportional to

$$L_n(\Sigma) = -\log|\Sigma^{-1}| + \mathrm{tr}[\Sigma^{-1} S]. \tag{1}$$

- Recent interests on p large or $p \approx n$. S is not a stable estimate.
  - The largest eigenvalue of S overly estimate the true eigenvalue.
  - When $p > n$, S is singular and the smallest eigenvalue is zero. How to estimate Σ^{-1}?

Motivation

- Estimate of Σ or Σ^{-1} needs to be positive definite.
  - The mathematical restriction makes the covariance matrix estimation problem challenging.
- Matrix transformation/decomposition approaches can relax the restriction: modified Cholesky decomposition, spherical parametrization, Givens parametrizarion, etc.
- Matrix logarithm: any positive definite Σ can be expressed as a matrix exponential of a real symmetric matrix A.

$$\Sigma = \exp(A) = I + A + \frac{A^2}{2!} + \cdots$$

  - Expressing the likelihood function in terms of $A \equiv log(\Sigma)$ releases the mathematical restriction.

Matrix Logarithm Parameterization

- Consider the spectral decomposition of $\Sigma = TDT'$ with $D = diag(d_1, \ldots, d_p)$ where T is an orthonormal matrix consisting of eigenvectors of Σ.
- Then we have

$$A = TMT'$$

with $M = diag(\log(d_1), \ldots, \log(d_p))$.

- In terms of the log-likelihood function,

$$-\log|\Sigma^{-1}| = \mathrm{tr}(A),$$

$$\mathrm{tr}[\Sigma^{-1}S] = \mathrm{tr}[\exp(-A)S].$$

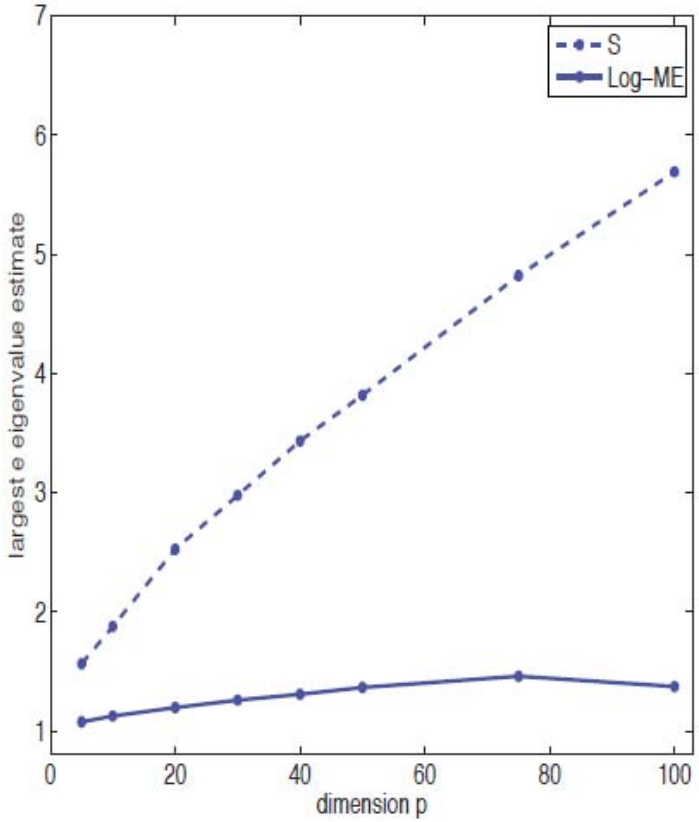
Idea of Proposed Method

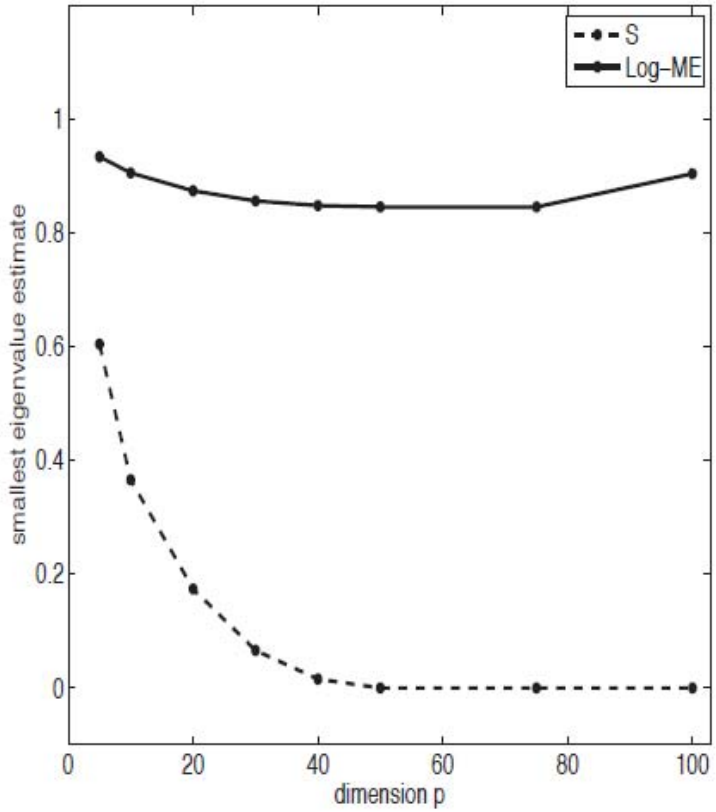
- Leonard and Hsu (1992) used this log-transformation method to estimate Σ by approximating the likelihood using Volterra integral equation.
  - Their approximation based on on S being nonsingular $\Rightarrow$ not applicable when $p \geq n$.
- We extend the likelihood approximation to the case of singular S.
- Regularize the largest and smallest eigenvalues of Σ *simultaneously*.
- An efficient iterative quadratic programming algorithm to estimate A ($\log \Sigma$).
- The resulting estimate is called "Log-ME", Logarithm-transformed Matrix Estimate.

A Simple Example

- Experiment: simulate x_i's from $N(0,I)$, $i = 1,\ldots,n$ where $n = 50$.
- For each p varying from 5 to 100, consider the the largest and smallest eigenvalues of the covariance matrix estimate.
- For each p, repeat the experiment 100 times and compute the average of the largest eigenvalues and the average of the smallest eigenvalues for
  - The sample covariance matrix.
  - The Log-ME covariance matrix estimate.

Example Illustration





The averages of the largest and smallest eigenvalues of covariance matrix estimates over the dimension p. The true eigenvalues are all equal to 1.

The Transformed Log-Likelihood

- In terms of the covariance matrix logarithm A, the negative log-likelihood function in (1) becomes

$$L_n(A) = \mathrm{tr}(A) + \mathrm{tr}[\exp(-A)S]. \qquad (2)$$

- The problem of estimating a positive definite matrix Σ now becomes a problem of estimating a real symmetric matrix A.
- Because of the matrix exponential term $\exp(-A)S$, estimating A by directly minimizing $L_n(A)$ is nontrivial.
- **Our approach:** Approximate $\exp(-A)S$ using the Volterra integral equation (valid even for S singular case).

The Volterra Integral Equation

- The Volterra integral equation (Bellman, 1970, page 175) is

$$\exp(At) = \exp(A_0 t) + \int_0^t \exp(A_0(t-s))(A-A_0)\exp(As)ds. \qquad (3)$$

- Repeatedly applying (3) leads to

$$\begin{aligned}\exp(At) = \exp(A_0 t) &+ \int_0^t \exp(A_0(t-s))(A-A_0)\exp(A_0 s)ds \\ &+ \int_0^t \int_0^s \exp(A_0(t-s))(A-A_0)\exp(A_0(s-u))(A-A_0)\exp(A_0 u)duds \\ &+ \text{cubic and higher order terms}, \qquad (4)\end{aligned}$$

  where $A_0 = \log(\Sigma_0)$ and Σ_0 is an initial estimate of Σ.

- One can get the expression of $\exp(-A)$ by letting $t = 1$ in (4) and replacing A, A_0 in (4) with $-A, -A_0$.

Approximation to the Log-Likelihood

- The term $\mathrm{tr}[\exp(-A)S]$ can be written as

$$\begin{aligned}\mathrm{tr}[\exp(-A)S] =& \mathrm{tr}(S\Sigma_0^{-1}) - \int_0^1 \mathrm{tr}[(A-A_0)\Sigma_0^{-s}S\Sigma_0^{s-1}]ds \\ &+ \int_0^1\int_0^s \mathrm{tr}[(A-A_0)\Sigma_0^{u-s}(A-A_0)\Sigma_0^{-u}S\Sigma_0^{s-1}]duds \\ &+ \text{cubic and higher order terms.} \qquad (5)\end{aligned}$$

- By leaving out the higher order terms in (5), we approximate $L_n(A)$ by using $l_n(A)$:

$$\begin{aligned}l_n(A) =& \mathrm{tr}(S\Sigma_0^{-1}) - \left[\int_0^1 \mathrm{tr}[(A-A_0)\Sigma_0^{-s}S\Sigma_0^{s-1}]ds - \mathrm{tr}(A)\right] \\ &+ \int_0^1\int_0^s \mathrm{tr}[(A-A_0)\Sigma_0^{u-s}(A-A_0)\Sigma_0^{-u}S\Sigma_0^{s-1}]duds. \qquad (6)\end{aligned}$$

Explicit Form of $l_n(A)$

- The integrations in $l_n(A)$ can be analytically solved through the spectral decomposition of $\Sigma_0 = T_0 D_0 T_0'$.
- **Some Notation:**
  - Here $D_0 = diag(d_1^{(0)}, \ldots, d_p^{(0)})$ with $d_i^{(0)}$'s as the eigenvalues of Σ_0.
  - $T_0 = (t_1^{(0)}, \ldots, t_p^{(0)})$ with $t_i^{(0)}$ as the corresponding eigenvector for $d_i^{(0)}$.
  - Let $B = T_0'(A - A_0)T_0 = (b_{ij})_{p\times p}$, and $\tilde{S} = T_0' S T_0 = (\tilde{s}_{ij})_{p\times p}$.
- In the integration,

$$\begin{aligned}
&\mathrm{tr}[(A - A_0)\Sigma_0^{-s} S \Sigma_0^{s-1}] = \mathrm{tr}[(A - A_0)T_0 D_0^{-s} T_0' S T_0 D_0^{s-1} T_0'], \\
&\mathrm{tr}[(A - A_0)\Sigma_0^{u-s}(A - A_0)\Sigma_0^{-u} S \Sigma_0^{s-1}] \\
&= \mathrm{tr}[(A - A_0)T_0 D_0^{u-s} T_0'(A - A_0)T_0 D_0^{-u} T_0' S T_0 D_0^{s-1} T_0'].
\end{aligned}$$

Explicit Form of $l_n(A)$ (Con't)

- The $l_n(A)$ can be written as a function of b_{ij}:

$$l_n(A) = \sum_{i=1}^{p} \frac{1}{2}\xi_{ii} b_{ii}^2 + \sum_{i<j} \xi_{ij} b_{ij}^2 + 2\sum_{i=1}^{p}\sum_{j\neq i} \tau_{ij} b_{ii} b_{ij} + \sum_{k=1}^{p} \sum_{i<j, i\neq k, j\neq k} \eta_{kij} b_{ik} b_{kj}$$
$$- \left[\sum_{i=1}^{p} \beta_{ii} b_{ii} + 2\sum_{i<j} \beta_{ij} b_{ij} \right], \qquad (7)$$

  up to some constant.

- Since $B = T_0'(A - A_0)T_0$, getting $B \leftrightarrow$ getting A.

Some Details

- For the linear term,

$$\beta_{ii} = \frac{\tilde{s}_{ii}}{d_i^{(0)}} - 1,\ \beta_{ij} = \frac{\tilde{s}_{ij}(d_i^{(0)} - d_j^{(0)})/(d_i^{(0)} d_j^{(0)})}{(\log d_i^{(0)} - \log d_j^{(0)})}.$$

- For the quadratic term,

$$\xi_{ii} = \frac{\tilde{s}_{ii}}{d_i^{(0)}},$$

$$\xi_{ij} = \frac{\tilde{s}_{ii}/d_i^{(0)} - \tilde{s}_{jj}/d_j^{(0)}}{\log d_j^{(0)} - \log d_i^{(0)}} + \frac{(d_i^{(0)}/d_j^{(0)} - 1)\tilde{s}_{ii}/d_i^{(0)} + (d_j^{(0)}/d_i^{(0)} - 1)\tilde{s}_{jj}/d_j^{(0)}}{(\log d_j^{(0)} - \log d_i^{(0)})^2},$$

$$\tau_{ij} = \left[\frac{1/d_j^{(0)} - 1/d_i^{(0)}}{(\log d_j^{(0)} - \log d_i^{(0)})^2} + \frac{1/d_i^{(0)}}{\log d_j^{(0)} - \log d_i^{(0)}}\right]\tilde{s}_{ij},$$

$$\eta_{kij} = \left[\frac{1/d_i^{(0)} - 1/d_j^{(0)}}{\log(d_k^{(0)}/d_j^{(0)})\log(d_j^{(0)}/d_i^{(0)})} + \frac{1/d_j^{(0)} - 1/d_i^{(0)}}{\log(d_k^{(0)}/d_i^{(0)})\log(d_i^{(0)}/d_j^{(0)})} + \frac{2/d_k^{(0)} - 1/d_i^{(0)} - 1/d_j^{(0)}}{\log(d_k^{(0)}/d_i^{(0)})\log(d_k^{(0)}/d_j^{(0)})}\right]\tilde{s}_{ij}$$

The Log-ME Method

- Propose a regularized method to estimate Σ by using the approximate log-likelihood function $l_n(A)$.
- Consider the penalty function $\|A\|_F^2 = \mathrm{tr}(A^2) = \sum_{i=1}^{p} (\log(d_i))^2$, where d_i is the eigenvalue of the covariance matrix Σ.
  - If d_i goes to zero or diverges to infinity, the value of $\log(d_i)$ goes to infinity in both cases.
  - Such a penalty function can *simultaneously* regularize the largest and smallest eigenvalues of the covariance matrix estimate.
- Estimate Σ, or equivalently A, by minimizing

$$l_{n,\lambda}(B) \equiv l_{n,\lambda}(A) = l_n(A) + \lambda \mathrm{tr}(A^2), \tag{8}$$

where λ is a tuning parameter.

The Log-ME Method (Con't)

- Note that $\mathrm{tr}(A^2) = \mathrm{tr}((T_0 B T_0' + A_0)^2)$ is equivalent to $\mathrm{tr}(B^2) + 2\mathrm{tr}(B\Gamma)$ up to some constant, where $\Gamma = (\gamma_{ij})_{p\times p} = T_0' A_0 T_0$.
- In terms of B, the function $l_{n,\lambda}(A)$ becomes

$$
\begin{aligned}
l_{n,\lambda}(B) = & \sum_{i=1}^{p} \frac{1}{2}\xi_{ii} b_{ii}^2 + \sum_{i<j} \xi_{ij} b_{ij}^2 + 2\sum_{i=1}^{p}\sum_{j\neq i} \tau_{ij} b_{ii} b_{ij} + \sum_{k=1}^{p} \sum_{i<j, i\neq k, j\neq k} \eta_{kij} b_{ik} b_{kj} \\
& - \left(\sum_{i=1}^{p} \beta_{ii} b_{ii} + 2\sum_{i<j} \beta_{ij} b_{ij} \right) \\
& + \lambda \left[\frac{1}{2}\sum_{i=1}^{p} b_{ii}^2 + \sum_{i<j}^{p} b_{ij}^2 + \sum_{i=1}^{p} \gamma_{ii} b_{ii} + 2\sum_{i<j} \gamma_{ij} b_{ij} \right].
\end{aligned} \tag{9}
$$

- The $l_{n,\lambda}(B)$ is still a quadratic function of $B = (b_{ij})$.

An Iterative Algorithm

- The $l_{n,\lambda}(B)$ depends on an initial estimate Σ_0, or equivalently, A_0.
- Propose to iteratively use $l_{n,\lambda}(B)$ to obtain its minimizer $\hat{B}$:

  Algorithm:

  Step 1: Set an initial covariance matrix estimate Σ_0, a positive definite matrix.

  Step 2: Use the spectral decomposition $\Sigma_0 = T_0 D_0 T_0'$, and set $A_0 = \log(\Sigma_0)$.

  Step 3: Compute $\hat{B}$ by minimizing $l_{n,\lambda}(B)$. Then obtain $\hat{A} = T_0 \hat{B} T_0' + A_0$, and update the estimate of Σ by

  $$\hat{\Sigma} = \exp(\hat{A}) = \exp(T_0 \hat{B} T_0' + A_0).$$

  Step 4: Check if $\|\hat{\Sigma} - \Sigma_0\|_F^2$ is less than a pre-specified positive tolerance value. Otherwise, set $\Sigma_0 = \hat{\Sigma}$ and go back to **Step 2**.

- Set an initial Σ_0 in **Step 1** to be $S + \varepsilon I$.

Properties of the Log-ME Method

- The iterative algorithm improves $\hat{A}$ step-by-step as the initial estimate Σ_0 is updated in each iteration.
- Recall that $L_n(A) = \mathrm{tr}(A) + \mathrm{tr}[\exp(-A)S]$ and $L_{n,\lambda}(A) \equiv L_n(A) + \lambda \mathrm{tr}(A^2)$.

  Proposition. *Suppose $\hat{A}$ is the solution of the proposed iterative algorithm when it converges. Then the value of $l_n(\hat{A})$ in (7) is exactly the same as the value of $L_n(\hat{A})$, i.e., $l_n(\hat{A}) = L_n(\hat{A})$.*
- When the proposed iterative algorithm converges at $\hat{A}$, $l_n(A)$ can provide a good approximation to the likelihood function $L_n(A)$ around $\hat{A}$.
- It indicates that $l_{n,\lambda}(A)$ can approximate $L_{n,\lambda}(A)$ well in a neighborhood region of $\hat{A}$.

Simulation Study

- Consider four different covariance models of $\Sigma = (\sigma_{ij})_{p\times p}$ for comparison,
  - Model 1: Σ is constructed from an MA(2) model with $\sigma_{ii} = 1, \sigma_{i,i-1} = \sigma_{i-1,i} = 0.6, \sigma_{i,i-2} = \sigma_{i-2,i} = 0.3$, and all the other σ_{ij} are zeros.
  - Model 2: $\Sigma = PMP'$, where M is the covariance matrix defined in model 2, and P is a permutation matrix by randomly permuting the rows of the identity matrix.
  - Model 3: $\Sigma^{-1} = (c_{ij})_{p\times p}$ with $c_{ii} = 1$ and $c_{ij} = 0.3$ if $i \neq j$.
  - Model 4: $\Sigma = GG$ where G is the covariance matrix defined in model 2.
- Compare with five covariance matrix estimates: (1) the LW estimate in Ledoit and Wolf (2006), (2) the banding estimate in Rothman et al. (2010), (3) the graphical Lasso estimate (Glasso) in Yuan and Lin (2007), (4) the estimate with a condition number constraint (denoted by CN) in Won et al. (2009), and (5) the CLIME estimate in Cai et al. (2011).

The CN Method

- The CN method is to estimate Σ with a constraint on its condition number (Won et al., 2009).
- They consider $\hat{\Sigma} = T\,\mathrm{diag}(\hat{u}_1^{-1}, \ldots, \hat{u}_p^{-1})T'$, where T is from the spectral decomposition of $S = T\,\mathrm{diag}(l_1, \ldots, l_p)T'$.
- The $\hat{u}_1, \ldots, \hat{u}_p$ are obtained by solving the constraint optimization

$$\min_{u, u_1, \ldots, u_p} \quad \sum_{i}^{p} (l_i u_i - \log u_i)$$

$$s.t. \quad u \le u_i \le \kappa_{max} u, \; i = 1, \ldots, p,$$

where κ_{max} is a tuning parameter.

The CLIME Method

- For the CLIME method, Cai et al. (2011) estimate the inverse covariance matrix $\Sigma^{-1} = (c_{ij})$ by solving

$$\min_{\Sigma^{-1}} \sum_{i,j} |c_{ij}| \text{ s.t. } |S\Sigma^{-1} - I|_\infty \leq \lambda,$$

  where λ is a tuning parameter.

- For all six methods in comparison, the tuning parameter in each method is selected through the independent validation set under the likelihood loss.

Simulation Study (Con't)

- Consider five loss functions to evaluate the performance of each method,
- The first three loss functions are the Kullback-Leibler (KL) loss, the entropy loss (EN), and the Frobenius norm (Fnorm),

$$KL = -\log|\hat{\Sigma}^{-1}| + \mathrm{tr}(\hat{\Sigma}^{-1}\Sigma) - (-\log|\Sigma^{-1}| + p),$$

$$EN = \mathrm{tr}(\Sigma^{-1}\hat{\Sigma}) - \log|\Sigma^{-1}\hat{\Sigma}| - p,$$

and

$$Fnorm = \|\hat{\Sigma} - \Sigma\|_F = \sqrt{\sum_{i,j}(\hat{\sigma}_{ij} - \sigma_{ij})^{ij}}.$$

- The other two loss functions are

$$\Delta_{1/p} = |\hat{d}_1/\hat{d}_p - d_1/d_p| \text{ and } \Delta_1 = |\hat{d}_1 - d_1|.$$

Here d_1 is the largest eigenvalue and d_p is the smallest eigenvalue of the covariance matrix Σ.

Simulation Results

Averages and standard errors from 100 runs for $n = 50, p = 100$.

	Model 1					Model 2				
Method	*KL*	*EN*	*Fnorm*	$\Delta_{1/p}$	Δ_1	*KL*	*EN*	*Fnorm*	$\Delta_{1/p}$	Δ_1
Log-ME	42.59	46.29	7.80	13.09	0.11	42.89	46.74	7.81	21.95	0.18
	(0.05)	(0.30)	(0.01)	(0.03)	(0.01)	(0.07)	(0.06)	(0.01)	(0.02)	(0.01)
Glasso	28.87	59.59	8.90	11.25	3.41	28.83	59.99	8.89	11.19	3.51
	(0.08)	(0.25)	(0.06)	(0.19)	(0.05)	(0.08)	(0.29)	(0.05)	(0.15)	(0.05)
LW	43.13	86.18	7.92	22.91	0.39	43.11	86.86	7.91	22.94	0.38
	(0.04)	(0.32)	(0.01)	(0.05)	(0.02)	(0.05)	(0.42)	(0.01)	(0.05)	(0.03)
CN	44.70	109.27	8.04	24.69	0.46	45.33	102.76	9.45	22.75	0.73
	(0.04)	(0.40)	(0.03)	(0.03)	(0.01)	(0.05)	(0.39)	(0.02)	(0.01)	(0.01)
Banding	10.12	8.49	3.34	32.35	0.60	57.40	140.69	9.64	24.30	1.19
	(0.10)	(0.07)	(0.02)	(1.15)	(0.03)	(0.05)	(0.50)	(0.00)	(0.16)	(0.02)
Clime	26.13	49.65	6.29	20.72	0.83	25.86	49.34	6.25	20.52	0.81
	(0.12)	(0.46)	(0.02)	(0.08)	(0.01)	(0.11)	(0.39)	(0.02)	(0.08)	(0.01)

Simulation Results (Con't)

Averages and standard errors from 100 runs for $n = 50, p = 100$.

	Model 3					Model 4				
Method	*KL*	*EN*	*Fnorm*	$\Delta_{1/p}$	Δ_1	*KL*	*EN*	*Fnorm*	$\Delta_{1/p}$	Δ_1
Log-ME	9.34	30.86	4.29	32.63	0.24	80.32	479.83	19.89	750.95	1.26
	(0.00)	(0.02)	0.01	0.04	0.04	(0.29)	(5.08)	(0.04)	(0.28)	(0.05)
Glasso	7.05	54.22	6.00	41.12	1.83	77.17	344.15	24.84	679.05	9.97
	(0.04)	(0.21)	(0.04)	(0.04)	(0.03)	(0.18)	(2.31)	(0.14)	(0.87)	(0.14)
LW	2.99	36.49	1.64	42.62	0.22	130.51	1377.90	18.98	759.12	2.08
	(0.01)	(0.10)	(0.02)	(0.01)	(0.01)	(0.10)	(6.87)	(0.03)	(0.13)	(0.09)
CN	3.72	41.54	2.43	42.85	0.20	158.69	3005.30	29.57	760.32	10.51
	(0.03)	(0.57)	(0.04)	(0.01)	(0.01)	(0.17)	(12.68)	(0.15)	(0.07)	(0.15)
Banding	5.14	39.84	3.15	41.07	0.76	175.20	2797.80	26.43	753.26	3.74
	(0.04)	(0.09)	(0.02)	(0.03)	(0.02)	(0.10)	(10.56)	(0.10)	(0.74)	(0.07)
Clime	2.90	40.53	1.54	42.68	0.11	85.23	282.74	21.42	747.04	4.16
	(0.01)	(0.02)	(0.02)	(0.01)	(0.03)	(0.17)	(2.51)	(0.03)	(0.29)	(0.03)

Classification of Ionosphere Data

- The data contains 351 observations with 34 variables with binary response 1 for "Good" and -1 for "Bad" of radar returns (Sigillito et al., 1989).
- Apply linear discriminant analysis (LDA) for classification under various covariance matrix estimate.
- Examine how different covariance matrix estimates can affect the classification performance of LDA.
- Randomly divide the data into a training set (of size 40), a validation set (of size 40) and a test set.

Misclassification Errors

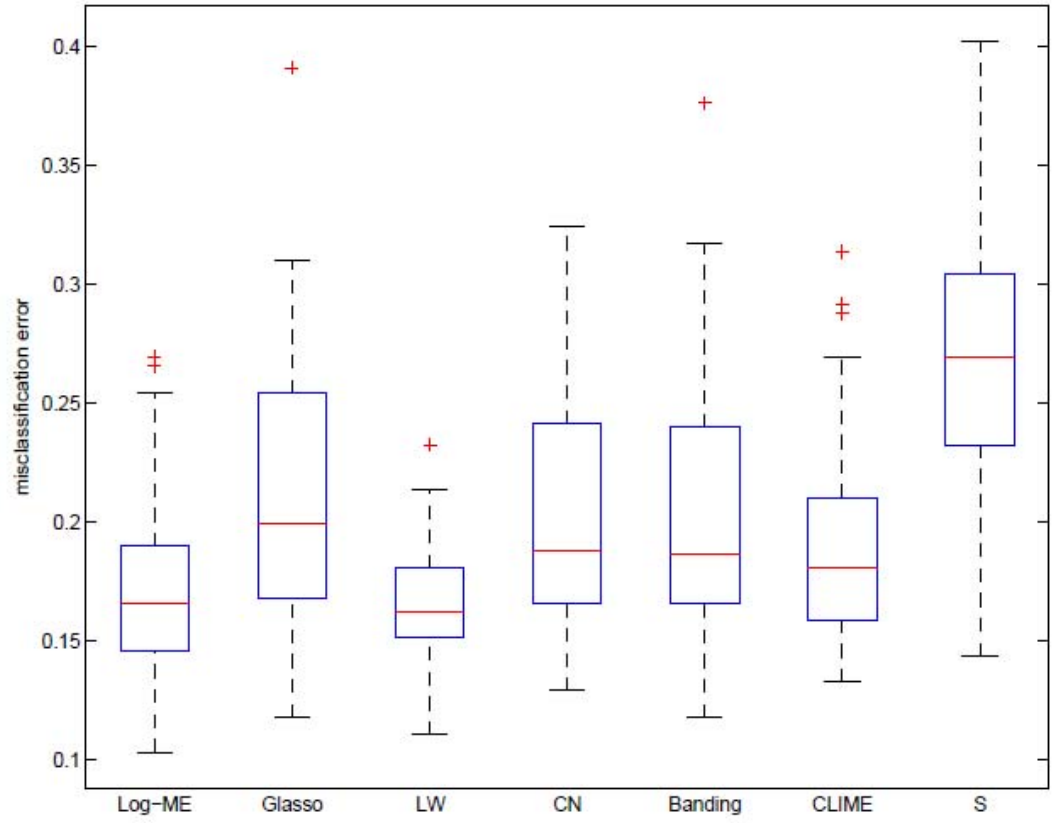


Figure 1. The boxplots of test misclassification errors from 100 replications.

Portfolio Optimization of Stock Data

- Apply the Log-ME method in the portfolio optimization.
- In mean-variance optimization, the risk of a portfolio $w = (w_1, \ldots, w_p)$ is measured by the standard deviation $\sqrt{w^T \Sigma w}$, where $\sum_i^p w_i = 1$.
- The estimated minimum variance portfolio optimization problem is

$$\min_w w^T \hat{\Sigma} w \qquad (10)$$
$$\text{s.t.} \sum_i^p w_i = 1,$$

  where $\hat{\Sigma}$ is an estimate of the true covariance matrix Σ.
- An accurate covariance matrix estimate $\hat{\Sigma}$ can lead to a better portfolio strategy.

The Setting-up

- Consider the weekly returns of $p = 30$ components of the Dow Jones Industrial Index from January 8th, 2007 to June 28th, 2010.
- Use the first $n = 50$ observations as a training set, the next 50 observations as a validation set, and *the remaining* 83 observations as a test set.
- Let X_{ts} be the test set and S_{ts} be the sample covariance matrix of X_{ts}. The performance of a portfolio w is measured by the *realized return*

$$R(w) = \sum_{x \in X_{ts}} w^T x,$$

and the *realized risk*

$$\sigma(w) = \sqrt{w^T S_{ts} w}.$$

- The optimal portfolio $\tilde{w}$ is computed with $\hat{\Sigma}$ from the Log-ME method and the other five methods, separately.

The Comparison Results

Table 1. The comparison of the realized return and the realized risk.

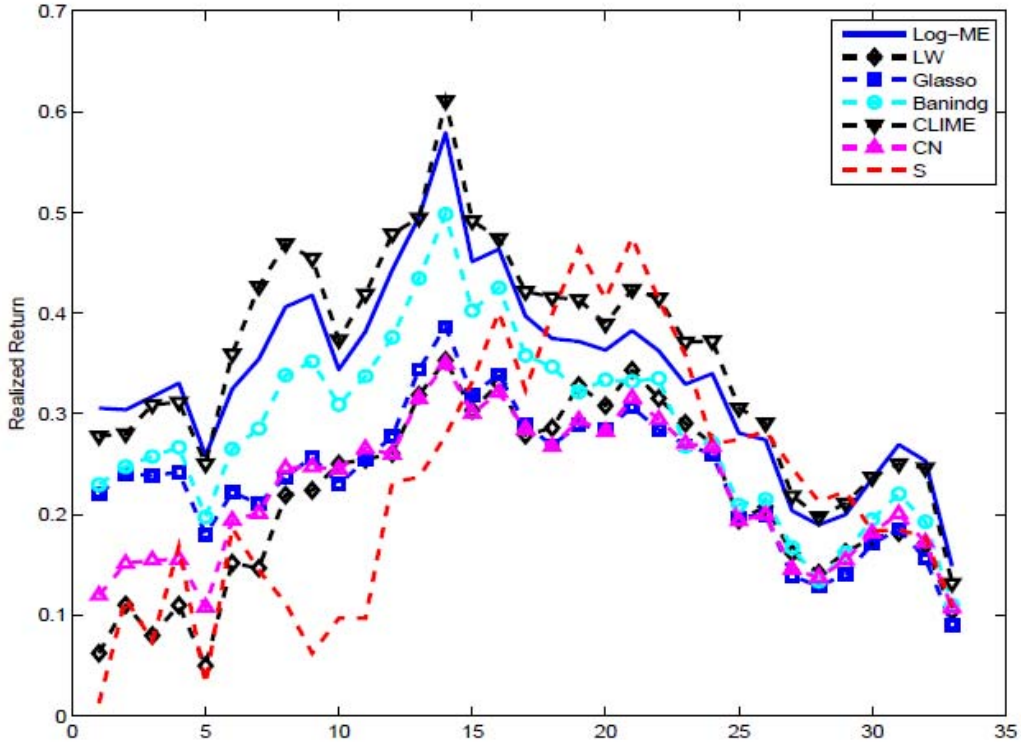
	Log-ME	CN	S	LW	Glasso	Banding	CLIME
Realized return $R(\tilde{w})$	0.218	0.128	0.059	0.062	0.193	0.192	0.211
Realized risk $\sigma(\tilde{w})$	0.029	0.024	0.035	0.025	0.028	0.029	0.029

- The Log-ME method produced a portfolio with a larger realized return but smaller realized risk.

Comparison in Different Periods

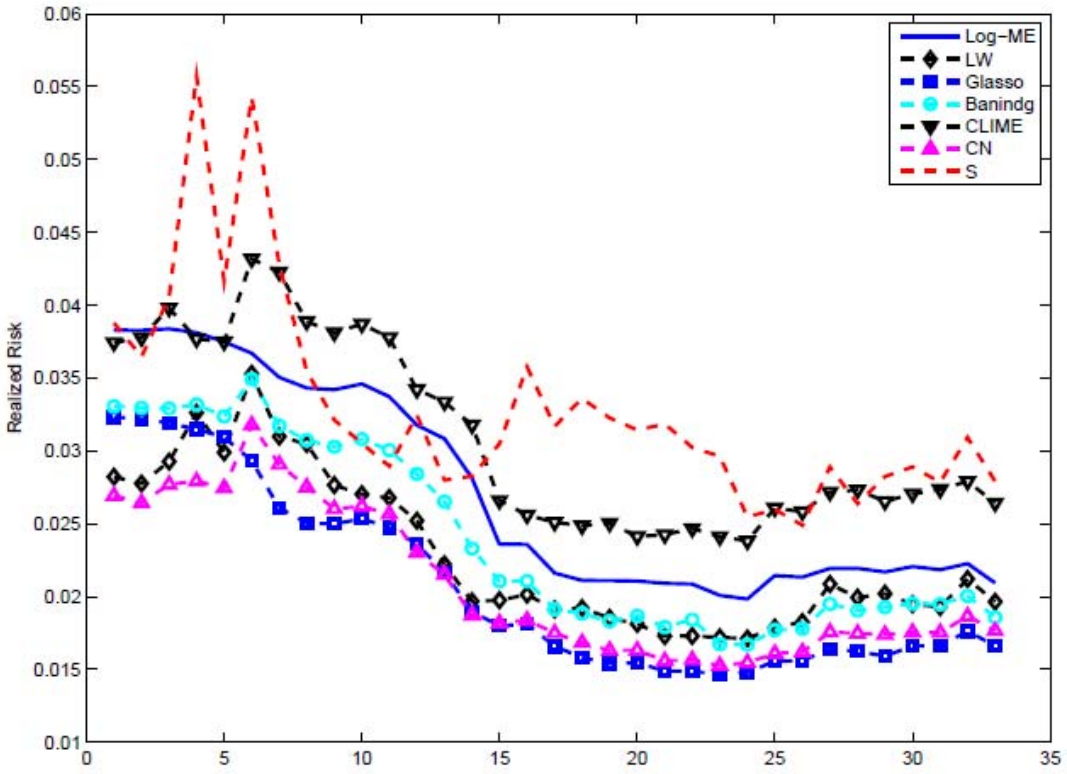
- Given a starting week, use the first 50 observations as the training set, the next 50 observations as a validation set, and *the third* 50 *observations* as a test set.
- Shift the starting week one ahead every time, and evaluate the portfolio strategy of 33 different consecutive test periods.
- The optimal portfolio $\tilde{w}$ is computed with $\hat{\Sigma}$ obtained from the Log-ME method and other methods in comparison.

The Realized Returns



The proposed Log-ME covariance matrix estimate can lead to higher returns.

The Realized Risks



The log-ME method has relatively higher risks than other methods, but it provides larger realized returns.

Extension:
Other Types of Penalties

- One can consider other types of penalties for regularization.
- Consider L_1 penalty on $A = (a_{ij})$ such that

$$L_{n,\lambda}(A) = \mathrm{tr}(A) + \mathrm{tr}[\exp(-A)S] + \lambda \sum_{i,j} |a_{ij}|.$$

- Need to be more careful on the interpretation of L_1 penalty.
- Alternatively, one can consider a penalty term with respect to a target,

$$L_{n,\lambda}(A) = \mathrm{tr}(A) + \mathrm{tr}[\exp(-A)S] + \lambda \|A - A^*\|_F^2.$$

Extension: Bayesian Perspective

- The approximated log-likelihood function $l_n(A)$ is a quadratic function of A.
- Denote $a = \mathrm{vec}(A) = (a_{11}, \ldots, a_{pp}, a_{12}, \ldots, \alpha_{p-1,p}, \ldots, a_{1,p})'$, Then we have

$$l_n(a) = Const + (a - a^{(0)} - Q^{-1}\beta)^T Q(a - a^{(0)} - Q^{-1}\beta). \qquad (11)$$

where

$$Q = \frac{1}{2}\sum_{i=1}^{p} \xi_{ii} f_{ii} f_{ii}^T + \sum_{i<j} \xi_{ij} f_{ij} f_{ij}^T + 2\sum_{i=1}^{p}\sum_{j\neq i} \tau_{ij} f_{ii} f_{ij}^T + \sum_{k=1}^{p}\sum_{i<j, i\neq k, j\neq k} \eta_{kij} f_{ik} f_{kj}^T,$$

and

$$\beta = \sum_{i=1}^{p} \frac{1}{2}\beta_{ii} f_{ii} + \sum_{i<j} \beta_{ij} f_{ij}.$$

Here $f_{ij} = t_i^{(0)} * t_j^{(0)}$ the product of the eigenvalues $t_i^{(0)}$ and $t_j^{(0)}$ such that $(a - a^{(0)})^T (t_i^{(0)} * t_j^{(0)}) = (t_i^{(0)})^T (A - A_0) t_j^{(0)} = b_{ij}$.

Bayesian Covariance Matrix Estimation (Con't)

- The $l_n(a)$ can be viewed as the log-density function of a multivariate normal distribution with mean vector $\alpha^{(0)} + Q^{-1}\beta$ and covariance matrix Q^{-1}.
- The proposed framework enables to apply flexible prior structures.
- Conjugate priors: $a = \text{vec}(A) \sim N(\gamma, R)$.
  - The posterior distribution of a is a multivariate normal with mean a^* and covariance R^* as

$$a^* = (Q + R^{-1})^{-1}(Qa^{(0)} + \beta + R^{-1}\gamma),$$
$$R^* = (Q + R^{-1})^{-1}.$$

Hierarchical Prior Structures

- Consider a hierarchical structure on the different bands of A. We set the prior of $a = \mathrm{vec}(A)$ as

$$
\begin{aligned}
a_{11}, a_{22}, \ldots, a_{pp} &\sim N(\mu_1, \sigma_1^2), \\
a_{12}, a_{23}, \ldots, a_{p-1,p} &\sim N(\mu_2, \sigma_2^2), \\
&\vdots \\
a_{1,p-1}, a_{2p} &\sim N(\mu_{p-1}, \sigma_{p-1}^2), \\
a_{1,p} &\sim N(\mu_p, \sigma_p^2).
\end{aligned}
$$

- The hierarchical structure lies in that $\mu = (\mu_1, \ldots, \mu_p)$ and $\sigma^2 = (\sigma_1^2, \ldots, \sigma_p^2)$ with prior distribution $p(\mu, \sigma^2)$ as

$$
\begin{aligned}
\mu_1, \mu_2, \ldots, \mu_p &\sim N(\mu_0, \tau_0^2), \\
\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2 &\sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2),
\end{aligned}
$$

  where μ_i's and σ_i^2's are independent each other.

Hierarchical Prior Structures (Con't)

- The posterior distribution of $a = \text{vec}(A)$ is $f(a|x_1,\ldots,x_n) \propto$

$$\exp\{-\frac{1}{2}l_n(a)\} \times \prod_{k=1}^{p} \prod_{i-j+1=k} N(a_{ij}|\mu_k,\sigma_k^2) \times \prod_{k=1}^{p} N(\mu_k|\mu_0,\tau_0^2) \prod_{k=1}^{p} \text{Inv-}\chi^2(\sigma_k^2|\nu_0,\sigma_0^2).$$

- The conditional posteriors have close forms:
  - The conditional posterior distribution $p(a|u,\sigma^2,X) \propto N(a^*,R^*)$ with

$$a^* = (Q+R^{-1})^{-1}(Qa^{(0)}+\beta+R^{-1}\gamma), \quad R^* = (Q+R^{-1})^{-1}.$$

    where $\gamma = (\mu_1 1_p,\ldots,\mu_{p-1}1_1)^T$ and $R = diag(\sigma_1 1_p,\ldots,\sigma_{p-1}1_1)$.
  - The conditional posterior distribution $(\mu_k|a,\sigma^2,X) \propto N(\mu_k^*,(\tau_k^2)^*)$, where

$$\mu_k^* = \frac{\frac{\sum_{i-j+1=k} a_{ij}}{\sigma_k^2} + \frac{\mu_0}{\tau_0^2}}{\frac{p-k+1}{\sigma_k^2} + \frac{1}{\tau_0^2}}, (\tau_k^2)^* = \frac{1}{\frac{\sum_{i-j+1=k} a_{ij}}{\sigma_k^2} + \frac{1}{\tau_0^2}}.$$

  - The conditional posterior of $(\sigma^2|a,\mu,X)$ follows scaled inverse χ^2 distributions.

Remarks

- Estimate the covariance matrix through its matrix logarithm based on a penalized likelihood function approximation.
- The Log-ME method regularizes the largest and smallest eigenvalues simultaneously by imposing a convex penalty.
- Can be extended to Bayesian covariance matrix estimation with flexible prior structures.

Multiple Graphical Models

- Suppose there is a heterogeneous data set with p variables and K categories. The k-th category contains n_k observations $(\boldsymbol{x}_1^{(k)}, \ldots, \boldsymbol{x}_{n_k}^{(k)})'$, where each $\boldsymbol{x}_i^{(k)} = (x_{i,1}^{(k)}, \ldots, x_{i,p}^{(k)})$ is a p-dimensional row vector.
- Assume $\boldsymbol{x}^{(k)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}^{(k)})$. It means there are multiple graphical models for estimation
- Denote $\boldsymbol{\Omega}^{(k)} = (\boldsymbol{\Sigma}^{(k)})^{-1}$. The negative log-likelihood can be written as

$$\min_{\boldsymbol{\Omega}^{(1)},\ldots,\boldsymbol{\Omega}^{(K)}} \sum_{k=1}^{K} n_k \left[-\log |\boldsymbol{\Omega}^{(k)}| + tr(\boldsymbol{\Omega}^{(k)} \boldsymbol{S}^{(k)}) \right],$$

up to come constant, where

$$\boldsymbol{S}^{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{x}_i^{(k)} (\boldsymbol{x}_i^{(k)})^T.$$

Multiple Graphical Model: Joint Estimation

- For a joint estimation, we can impose consistent sparse structure on the $\mathbf{\Omega}^{(k)} = (\omega_{ij}^{(k)})_{p\times p}$.
- Remark: The idea is similar as the group lasso. while now we focus group structure for graphical lasso.
- The penalized log-likelihood estimation can be considered by minimizing

$$\sum_{k=1}^{K} n_k \left[-\log |\mathbf{\Omega}^{(k)}| + tr(\mathbf{\Omega}^{(k)} \boldsymbol{S}^{(k)}) \right] + \lambda \sum_{i\neq j} \sqrt{\sum_{k=1}^{K} |\omega_{ij}^{(k)}|},$$

  where λ is a tuning parameter.
- Other meaningful penalties can also be considers. The reparameterization technique is also an alternative approach to address the structured penalty.

Joint Estimation Tools

- To solve the above optimization efficiently, we can apply local linear approximation to the penalty term.
- Denote $(\omega_{ij}^{(k)})_{(m)}$ the estimate of $\omega_{ij}^{(k)}$ in $f_k^{-1} = (\omega_{ij}^{(k)})_{p\times p}$ at mth iteration. Then using local linear approximation, we write

$$\sqrt{\sum_{k=0}^{n} |\omega_{ij}^{(k)}|} = \frac{\sum_{k=0}^{n} |\omega_{ij}^{(k)}|}{\sqrt{\sum_{k=0}^{n} |(\omega_{ij}^{(k)})_{(m)}|}} \equiv \alpha_{ij}^{(m)} \sum_{k=0}^{n} |\omega_{ij}^{(k)}|,$$

  where $\alpha_{ij}^{(m)} = 1/\sqrt{\sum_{k=0}^{n} |(\omega_{ij}^{(k)})_{(m)}|}$.
- Now the glasso (Friedman et al., 2008) can be used for efficient computation. At the $(m+1)$th iteration, we minimize

$$\sum_{k=1}^{K} n_k \left[-\log |\mathbf{\Omega}^{(k)}| + tr(\mathbf{\Omega}^{(k)} \boldsymbol{S}^{(k)}) \right] + \lambda \sum_{i\neq j} \alpha_{ij}^{(m)} |\omega_{ij}^{(k)}|.$$

- The above is to estimate each $\mathbf{\Omega}^{(k)}$ with positive-definiteness.

Some References

- Huang, J. Z., Liu, N., Pourahmadi, M., and Liu, L. (2006). Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, **93**, 85–98.
- Bickel, P. J. and Levina., E. (2008). Regularized estimation of large covariance matrices. *Annals of Statistics*, **36**, 199–227.
- Rothman, A. J., Levina, E., and Zhu, J. (2010). A new approach to Cholesky-based estimation of high-dimensional covariance matrices. *Biometrika*, **97**, 539–550.
- Cai, T., Liu, W., and Luo, X. (2011). A constrained l_1 minimization approach to sparse precision matrix estimation. *Journal of American Statistical Association*, **106**, 594–607.
- Candès, E., and Tao, T. (2007), The Dantzig Selector, Statistical Estimation When p is Much Larger Than n. *The Annals of Statistics*, **35**, 2313–2351.

Remarks

Thank you!

- **Questions and Comments?**