Graphic Lasso: Paper Review & Simulation

Jiaxin Hu

December 18, 2020

- Paper review for *Sparse inverse covariance estimation with the graphical lasso.*
- Simulation for GLasso under different sparsity settings.

1 Paper Review

1.1 Motivation

Consider the observations $Y_i = (Y_{i1}, ..., Y_{ip})^T \in \mathbb{R}^p, i = 1, ..., n$ follow a multivariate Gaussian model $\mathcal{N}_p(\mu, \Sigma)$. Let $V_j = (Y_{1j}, ..., Y_{nj})^T \in \mathbb{R}^n$ denote the realization of j-th variable, for $j = 1, ..., p$. If the component of $\Sigma^{-1}_{kl} = 0$, then variables V_k and V_l are conditionally independent given other variables. Under special structures of Σ, the inverse covariance matrix Σ^{-1} becomes sparse, and thereby L_1 penalty is useful in the estimation of Σ^{-1}.

1.2 Estimation

Let $\Theta = \Sigma^{-1}$. The Lasso-based estimator satisfies

$$\hat{\Theta} = \arg\max_{\Theta} \log(\det(\Theta)) - \text{tr}(S\Theta) - \rho \left\|\Theta\right\|_1,$$

where $S = \frac{1}{n}\sum_{i=1}^n (Y_i - \mu)(Y_i - \mu)^T \in \mathbb{R}^{p\times p}$ is the empirical (sample) covariance matrix, ρ is the tuning parameter, and $\left\|\Theta\right\|_1$ is the sum of the absolute value of the all elements in Θ.

Let $W = \hat{\Sigma}$ be the estimate of Σ. We recover the inverse Θ by finding W. Consider the partition

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22}. \end{pmatrix}$$

By estimating w_{12} and permuting the rows and columns in W, we are able to estimate all the off-diagonal elements, which are of our interest, with an "alternative" algorithm.

Banerjee and others (2007) shows the solution for w_{12} to the problem (**??**) satisfies

$$w_{12} = \arg\min_{y} \left\{ y^T W_{11}^{-1} y : \left\|y - s_{12}\right\|_\infty \leq \rho \right\},$$

which is equal to the dual problem

$$\min_{\beta} \left\{ \frac{1}{2} \left\|W_{11}^{-1}\beta - b\right\|^2 + \rho \left\|\beta\right\|_1 \right\},$$

where $b = W_{11}^{-1/2} s_{12}$. If β solves (**??**), then $w_{12} = W_{11}\beta$ solves (**??**). In fact, the proposed algorithm solves problem (**??**) in each updating step.

1.3 Algorithm

Algorithm 1 Graphical Lasso

Input: Empirical Covariance matrix S, tuning parameter ρ.
Output: Estimated inverse covariance matrix Θ.

1: Initialize $W^{(0)} = S + \rho \boldsymbol{I}_p$.
2: **while** Do until convergence, i.e., $\left\| W^{(i)} - W^{(i-1)} \right\|_{\max} < 1e^{-6}$ **do**
3:     **for** $j = p$ to 1 **do**
4:         Permute the j-th row and column to the last row and column, respectively.
5:         Solve $\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{2} \left\| W_{11}^{-1} \beta - b \right\|^2 + \rho \left\| \beta \right\|_1 \right\}$, where the $j, ..., p-1$-th rows and columns in W_{11} are from $W^{(j)}$, and the $1, ..., j-1-$th rows and columns are from $W^{(j-1)}$.
6:         Plug $w_{12} = W_{11} \hat{\beta}$ into $W^{(j)}$.
7:     **end for**
8:     Obtain the estimate $\hat{\Theta} = (\hat{W})^{-1}$, where $\hat{W}$ is the convergent point of $W^{(j)}$.
9: **end while**

In my implementation, I use the function `glmnet` to solve the marginal Lasso problem in step 5. BIC is used to choose a proper tuning parameter ρ.

2 Simulation Results

We use false negative rate (FN) and false positive rate (FP) to evaluate the algorithm. We consider five different settings (structures) of the covariance matrix. The elements with absolute value smaller than $1e^{-16}$ are considered as 0.

1. **Sparse setting:** Let $\Sigma_{ii}^{-1} = 1$, for $i = 1, ..., n$, and $\Sigma_{i.i-1}^{-1} = \Sigma_{i-1,i}^{-1} = 0.5$, for $i = 2, ..., p$.

2. **Dense setting:** Let $\Sigma_{ii}^{-1} = 2, \Sigma_{ij}^{-1} = 1, i \neq j$, for $i, j = 1, ..., p$.

3. **Chain network setting:** Let $\Sigma_{i,j} = \exp\{-\frac{|s_i - s_j|}{2}\}$, for $i, j = 1, ..., p$, where $s_1 < \cdots < s_p$ and $s_k - s_{k-1} \sim U(0.5, 1)$, for $k = 2, ...p$.

4. **M-nearest network setting:**

    - Let (x_i, y_i) for $i = 1, ..., p$ be random points in the square $[0, 1] \times [0, 1]$. Calculate the euclidean distance between each pair of points, i.e., calculate $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where $i \neq j$.
    - For point (x_i, y_i), find the m nearest point based on the distance $d_{ij}, i \neq j$. Let $j_1, ..., j_m$ denote the index of the points which at least have the m-th shortest distance with (x_i, y_i).
    - Let $\Sigma_{i,j_k} \sim U([-1, -0.5] \cup [0.5, 1])$, for $k = 1, ...m$.
    - Repeat the second and third steps for p points.
    - To ensure the covariance matrix Σ is invertible and the diagonal elements dominate, we let $\Sigma_{ii} = \sum_{j=1}^{p} |\Sigma_{ji}| + 1$.

5. **Scale-free (Random) network setting:**

    - Randomly decide 20% of the $p(p-1)/2$ off-diagonal elements in Σ are nonzero. For the nonzero element $\Sigma_{ij} \sim U([-1, -0.5] \cup [0.5, 1])$.

- To ensure the covariance matrix Σ is invertible and the diagonal elements dominate, we let $\Sigma_{ii} = \sum_{j=1}^{p} |\Sigma_{ji}| + 1$.

The false negative rates of the graphical Lasso under different settings and varying p are showed in Figure **??**.

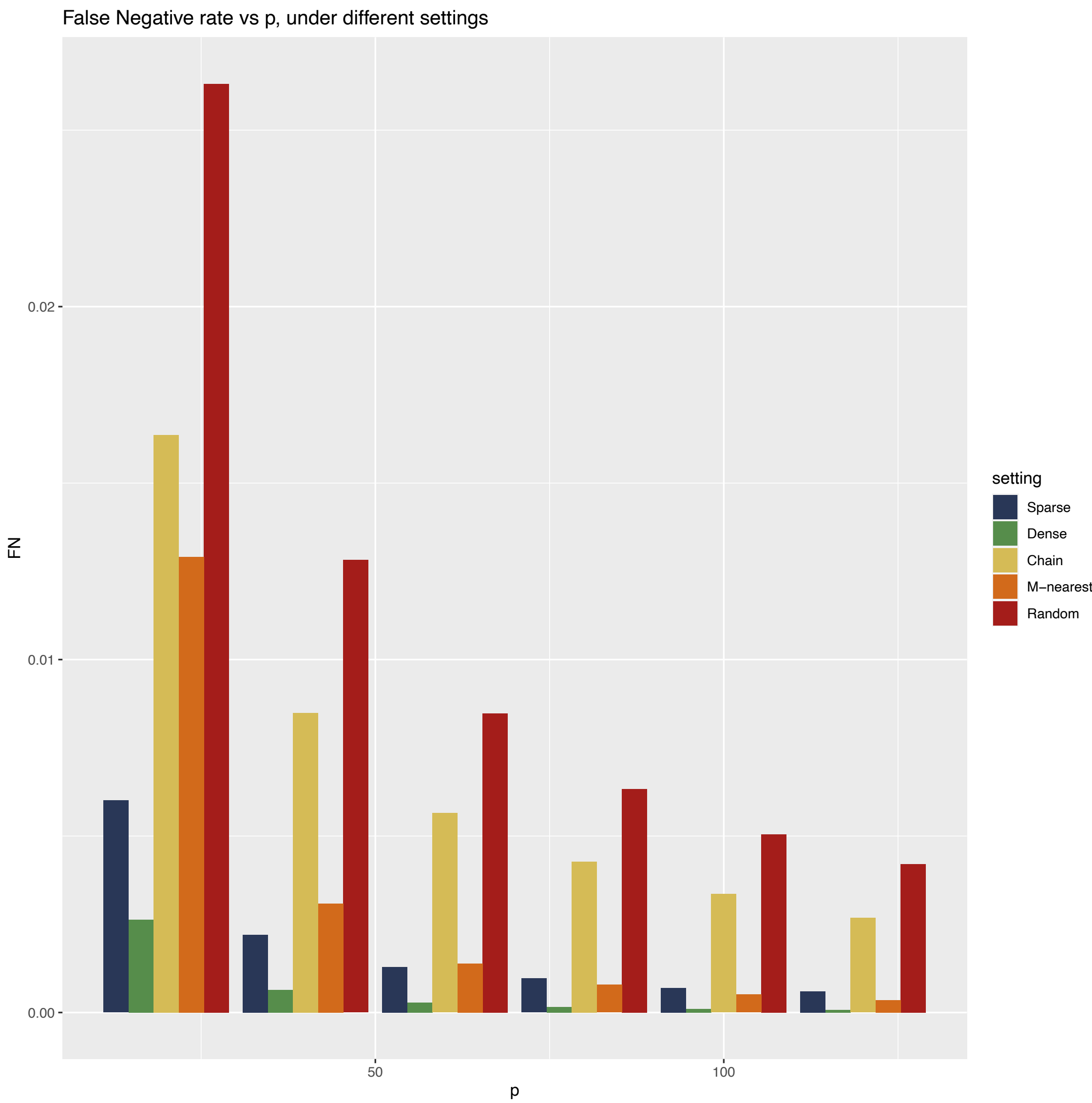

Figure 1: False negative rate of graphical Lasso under different settings. The dimension p varies from 20 to 120. Every value is an average result from 3 duplications.

3 To do list & Issues

- The false positive rates of the graphical Lasso under all settings are equal to 0. I guess it blames to the bad choice of the tuning parameter ρ. If we can choose smaller ρ, the shrinkage may not be so strong, and the false positive rate may not be 0 all the time.
- I did not use the package `glasso`. I just rewrote the matlab codes in the Github. I would love to check whether there are any differences between them.

- I was thinking about how to interpret the false negative rate performance. I guess the performance is related to sparsity (ratio of the zero elements) in each setting.
- I did not fully command all the theoretical parts in the paper. I need to go through the paper again.