

## Topic 16: Graphical Network Inference

by Sai Zhang

## Key points:

**Disclaimer:** The note is built on Prof. *Jinchi Lv*'s lectures of the course at USC, DSO 607, High-Dimensional Statistics and Big Data Problems.

## 16.1 Motivation

Consider a classic question: For  $n$  observations of dimension  $p$ , how can we capture the statistical relationships between the variables of interest? Consider the example of the multivariate Gaussian distribution:

## Example 16.1.1: Multivariate Gaussian Distribution

Suppose we have  $n$  observations of dimension  $p$ ,  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . let  $\mathbf{S}$  be the empirical covariance matrix. Then the probability density

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

define the **inverse covariance matrix** or **precision matrix** as  $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ , then we have

$$f_{\boldsymbol{\mu}, \boldsymbol{\Omega}} = \exp \left\{ \boldsymbol{\mu}' \boldsymbol{\Omega} \mathbf{x} - \left\langle \boldsymbol{\Omega}, \frac{1}{2} \mathbf{x} \mathbf{x}' \right\rangle - \frac{p}{2} \log(2\pi) + \frac{1}{2} \log \det(\boldsymbol{\Omega}) - \frac{1}{2} \boldsymbol{\mu}' \boldsymbol{\Omega} \boldsymbol{\mu} \right\}$$

where  $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}\mathbf{B})$ .

In this example, we know that **every** multivariate Gaussian distribution can be represented by a pairwise **Gaussian Markov Random Field (GMRF)**, which an **undirected graph**  $G = (V, E)$

- representing the collection of variables  $\mathbf{x}$  by a vertex set  $\mathcal{V} = \{1, \dots, p\}$
- encoding correlations between variables by a set of edges  $\mathcal{E} = \{(i, j) \in \mathcal{V} \mid i \neq j, \Omega_{ij} \neq 0\}$

For simplicity, we normalize  $\boldsymbol{\mu} = \mathbf{0}$ . If we draw  $n$  i.i.d. samples  $\mathbf{x}_1, \dots, \mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ , then the log-likelihood is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\Omega}) &= \frac{1}{n} \sum_{i=1}^n \log f(\mathbf{x}_i) = \frac{1}{2} \log \det(\boldsymbol{\Omega}) - \frac{1}{2n} \sum_{i=1}^n \mathbf{x}_i' \boldsymbol{\Omega} \mathbf{x}_i \\ &= \frac{1}{2} \log \det(\boldsymbol{\Omega}) - \frac{1}{2} \left\langle \boldsymbol{\Omega}, \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i' \right\rangle \end{aligned}$$

**What's the goal?** We want to estimate a **sparse** graph structure given  $n \ll p$  i.i.d. observations. But what does sparsity means in this context? A sparse graph is **equivalent** to a sparse precision matrix: the precision

matrix should have many 0s.

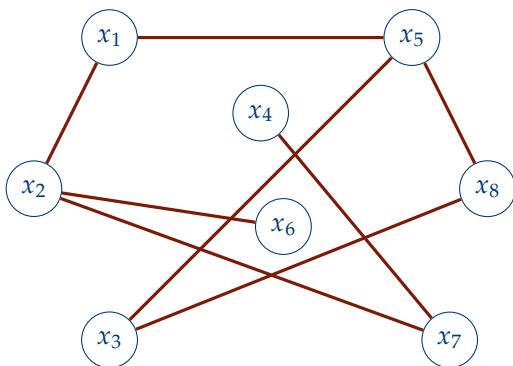
**Sparse precision matrix** for the Gaussian vector mentioned above  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ , we have  $\forall u, v$

$$x_u \perp x_v \mid \mathbf{x}_{V \setminus \{u, v\}} \Leftrightarrow \Omega_{u, v} = 0$$

that is, sparsity of the precision matrix is equivalent to **conditional independence**<sup>1</sup>. Consider a graph, where  $x_1$  and  $x_4$  are only connected through other nodes, that is  $x_1$  and  $x_4$  are conditional independent, then we can have the precision matrix be something like:

$$\Theta = \begin{bmatrix} * & * & 0 & 0 & * & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 & * & * & 0 \\ 0 & 0 & * & 0 & * & 0 & 0 & * \\ 0 & 0 & 0 & * & 0 & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 & 0 & * \\ 0 & * & 0 & 0 & 0 & * & 0 & 0 \\ 0 & * & 0 & * & 0 & 0 & * & 0 \\ 0 & 0 & * & 0 & * & 0 & 0 & * \end{bmatrix}$$

where 0 captures precisely the conditional independence.



$x_1$  and  $x_4$  are connected



$x_1$  and  $x_4$  are NOT connected, conditionally

Intuitively, a sparse graph is much simpler, which is why conditional independence is desired. So how to achieve sparsity? We can again use a L-1 regularization when maximizing the log-likelihood  $\mathcal{L}(\Omega)$ . Denote the sample covariance matrix as  $\mathbf{S} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i'$ , then the problem becomes the so-called **Graphical Lasso**

$$\max_{\Omega \geq 0} \log \det(\Omega) - \text{tr}(\mathbf{S}\Omega) - \rho \|\Omega\|_1$$

which is equivalent to

$$\min_{\Omega \geq 0} -\log \det(\Omega) + \text{tr}(\mathbf{S}\Omega) + \rho \|\Omega\|_1$$

## 16.2 Graphical Lasso

The graphical lasso method is developed by (Friedman et al., 2008). For the optimization problem

$$\min_{\Omega \geq 0} -\log \det(\Omega) + \text{tr}(\mathbf{S}\Omega) + \rho \|\Omega\|_1 \quad (16.1)$$

<sup>1</sup>Meanwhile, for independence:  $\Sigma_{u, v} = 0 \Leftrightarrow x_u \perp x_v$

The first-order optimality condition gives

$$\mathbf{0} \in \mathbf{\Omega}^{-1} - \mathbf{S} - \lambda \mathbf{\Gamma}$$

where  $\mathbf{\Gamma}$  is a matrix of component-wise signs of  $\mathbf{\Omega}$

$$\mathbf{\Gamma} = \partial \|\mathbf{\Omega}\|_1 \Rightarrow \gamma_{jk} \begin{cases} = \text{sign}(\omega_{jk}), & \omega_{jk} \neq 0 \\ \in [-1, 1], & \omega_{jk} = 0 \end{cases}$$

since in a graph, we always have that, following the global stationary conditions,  $\omega_{jj} > 0$ , which implies that

$$w_{ii} = s_{ii} + \lambda \quad i = 1, \dots, p \quad (16.2)$$

where we denote a working version of  $\mathbf{\Omega}^{-1}$  as  $\mathbf{W}$ .

The idea is to repeatedly cycle through all columns-rows and in each step optimize only a single column-row. Consider the following partition where all matrices are partitioned into one column/row versus the rest

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Omega}_{11} & \boldsymbol{\omega}_{12} \\ \boldsymbol{\omega}'_{12} & \omega_{22} \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}'_{12} & s_{22} \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}'_{12} & w_{22} \end{pmatrix} \quad \mathbf{\Gamma} = \begin{pmatrix} \mathbf{\Gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}'_{12} & \gamma_{22} \end{pmatrix}$$

apply this partition to the optimality condition, get

$$\mathbf{\Omega}^{-1} = \mathbf{S} - \lambda \mathbf{\Gamma}$$

$$\begin{pmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}'_{12} & w_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}'_{12} & s_{22} \end{pmatrix} + \lambda \begin{pmatrix} \mathbf{\Gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}'_{12} & \gamma_{22} \end{pmatrix}$$

where  $\mathbf{\Omega}_{11}$  is  $(p-1) \times (p-1)$ ,  $\boldsymbol{\omega}_{12}$  is  $(p-1) \times 1$ ,  $\omega_{22}$  is a scalar.

Consider a **blockwise** step: suppose we fix all but the last row/column, then using properties of inverses of block-partitioned matrices, we have

$$\begin{pmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}'_{12} & w_{22} \end{pmatrix} = \begin{pmatrix} \left( \mathbf{\Omega}_{11} - \frac{\boldsymbol{\omega}_{12} \boldsymbol{\omega}'_{12}}{\omega_{22}} \right)^{-1} & -\mathbf{W}_{11} \frac{\boldsymbol{\omega}_{12}}{\omega_{22}} \\ \frac{1}{\omega_{22}} - \frac{\boldsymbol{\omega}'_{12} \mathbf{W}_{11} \boldsymbol{\omega}_{12}}{\omega_{22}^2} & \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{\Omega}_{11}^{-1} + \frac{\mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12} \boldsymbol{\omega}'_{12} \mathbf{\Omega}_{11}^{-1}}{\omega_{22} - \boldsymbol{\omega}'_{12} \mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12}} & -\frac{\mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12}}{\omega_{22} - \boldsymbol{\omega}'_{12} \mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12}} \\ \frac{1}{\omega_{22} - \boldsymbol{\omega}'_{12} \mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12}} & \end{pmatrix}$$

then, by the partitioned optimality condition, we have<sup>2</sup>:

$$\mathbf{0} = -\mathbf{w}_{12} + \mathbf{s}_{12} + \lambda \boldsymbol{\gamma}_{12} = \mathbf{W}_{11} \frac{\boldsymbol{\omega}_{12}}{\omega_{22}} + \mathbf{s}_{12} + \lambda \boldsymbol{\gamma}_{12} \quad (16.3)$$

$$\mathbf{0} = \frac{\mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12}}{\omega_{22} - \boldsymbol{\omega}'_{12} \mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12}} + \mathbf{s}_{12} + \lambda \boldsymbol{\gamma}_{12} = w_{22} \mathbf{\Omega}_{11}^{-1} \boldsymbol{\omega}_{12} + \mathbf{s}_{12} + \lambda \boldsymbol{\gamma}_{12} \quad (16.4)$$

The graphic Lasso algorithm then solves Eq.16.3 for  $\boldsymbol{\beta} = \boldsymbol{\omega}_{12}/\omega_{12}$ , that is

$$\mathbf{W}_{11} \boldsymbol{\beta} + \mathbf{s}_{12} + \lambda \boldsymbol{\gamma}_{12} = \mathbf{0}$$

<sup>2</sup>For Eq.16.4, by Eq.16.2, we know that  $w_{22} = s_{22} + \lambda$ , which is fixed.

where  $\gamma_{12} \in \text{sign}(\beta)$  since  $\omega_{22} > 0$ , which is essentially solving:

$$\min_{\beta \in \mathbb{R}^{p-1}} \left\{ \frac{1}{2} \beta' \mathbf{W}_{11} \beta + \beta' \mathbf{s}_{12} + \lambda \|\beta\|_1 \right\}$$

and  $\mathbf{W}_{11} > 0$  is assumed to be fixed.

This problem is analogous to a lasso regression problem of **the last variable on the rest**, but the cross-product matrix  $\mathbf{S}_{11}$  is replaced by its **current estimation**  $\mathbf{W}_{11}$ . It is relatively easier to solve using elementwise coordinate descent, then

$$\begin{aligned} \mathbf{w}_{12} &= -\mathbf{W}_{11} \frac{\omega_{12}}{\omega_{22}} & \Rightarrow \hat{\mathbf{w}}_{12} &= -\mathbf{W}_{11} \hat{\beta} & \text{Step 1} \\ w_{22} &= \frac{1}{\omega_{22}} - \frac{\omega'_{12} \mathbf{W}_{11} \omega_{12}}{\omega_{22}^2} & \Rightarrow \frac{1}{\hat{\omega}_{22}} &= w_{22} - \hat{\beta}' \hat{\mathbf{w}}_{12} & \text{Step 2} \\ \omega_{12} &= -\mathbf{W}_{11}^{-1} \mathbf{w}_{12} \omega_{22} & \Rightarrow \hat{\omega}_{12} &= -\mathbf{W}_{11}^{-1} \hat{\mathbf{w}}_{12} \hat{\omega}_{22} & \text{Step 3} \end{aligned}$$

notice that after solving for  $\beta$  and updating  $\mathbf{w}_{12}$  in Step 1, the graphic Lasso procedure can move onto the next block, that is, only Step 1 is used in the loop, Step 2 and 3 can be done at the end. The algorithm can be summarized as:

#### Algorithm 16.2.1: SOFAR algorithm with ALM-BCD

- 1 Initialize  $\mathbf{W} = \mathbf{S} + \lambda \mathbf{I}$
- Cycle around the columns repeatedly, performing the following steps till convergence:
  - a rearrange the rows/columns so that the target column is **the last** (implicitly)
  - b solve the lasso problem, starting the solution from the previous round for this column
  - c update the row/column (*off-diagonal*) of the covariance using  $\hat{\mathbf{w}}_{12}$
  - d save  $\hat{\beta}$  for this column in the matrix  $\mathbf{B}$
- 3 after convergence, for every row/column, compute the diagonal entries  $\hat{\theta}_{jj}$ , and covert the  $\mathbf{B}$  matrix to  $\Theta$

## References

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.