Introduction to sparse Gaussian Graphical Models

julien.chiquet@gmail.com

M2 ISG - summer 2016 -
network modelling - practical 3

1 Context

We consider a random vector $X = (X_1, \ldots, X_p) \in \mathbb{R}^p$ whose joint distribution is described by a multivariate Gaussian distribution. Without loss of generality, we assume that $\mathbb{E} = \mathbf{0}_p$. The variance-covariance matrix is denoted by $\mathbf{\Sigma}$ and we have $\mathcal{N}(\mathbf{0}_p, \mathbf{\Sigma})$.

We consider the conditional dependency graph $G = (V, E)$ such that $V = \{1, \ldots, p\}$ and $(i, j) \in E$ whenever there is a significant conditional dependency between variable X_i and X_j. Thanks to the Gaussian assumption, this boils down to unraveling significant non zero entries in the inverse covariance matrix, *a.k.a.* the precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$. In other words, as seen in the course,

$$G: \ (i, j) \notin E \Leftrightarrow \Omega_{ij} = 0.$$

We would like to infer the conditional dependency graph G from a data set that possibly enters the high-dimensional data setup, that is, $n < p$. More formally, we consider a sample $(X^1, \ldots, X^n)$ of n independent copies of X. We denote by $\boldsymbol{X}$ the $n \times p$ data matrix, the ith row of which contains the data associated with individual i, that is, X^i.

2 Questions

2.1 Preliminaries

1. *Multivariate Gaussian log-likelihood.* Derive the data log-likelihood as a function of the parameter $\mathbf{\Omega}$ and the MLE estimator.
2. *Gaussian vector and linear regression.* Write the conditional distribution of $X_j | X_{\setminus j}$. Show that X_j can be expressed as a linear combination of the $\{X_k, k \neq j\}$ plus some Gaussian noise, that is, a linear regression model.
3. *GGM and linear regression.* In this model, show that the regression coefficients depends on $\mathbf{\Omega}$ only. Recast the network reconstruction problem as p independent Lasso problems.

2.2 Simulating GGM

1. *Multivariate Gaussian sample.* Write a function `rmvnorm(n,mu,Sigma)` that draws n samples of a Gaussian vector with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and sends back a $n \times p$ matrix X. To this end, remark that $\mathbb{V}(Z\boldsymbol{\Sigma}^{1/2})$ where $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ has the same covariance as X.
2. *From adjacency to precision matrices.* Write a function `getPrecision(A)` that takes as an argument a binary symmetric adjacency matrix and computes a symmetric positive-definite matrix with the same sparsity pattern. You can use the property of positive-definiteness own by diagonal dominant matrices and draw $\boldsymbol{\Omega}$ such that

$$\boldsymbol{\Omega} = A \times \min(\text{eig}(A)) + \mathbf{I}_p \times (\text{cst.} + |\min(\text{eig}(A))|).$$

3. *From adjacency matrices to multivariate Gaussian samples.* By means of the two previous questions, write a function `rggm(n,A)` that returns a matrix of Gaussian data.

2.3 Learning GGM

1. *Optional, depending on the timing.* Thanks to Section 2.1, write a function `neighborhood.selection(X,lambda)` that learns the sparsity pattern of the precision matrix of X by solving p independent Lasso problems [1]. The post-symmetrization can be done either by means of a 'AND' or a 'OR' rule.
2. Make some experiments to assess the performances of the neighborhood selection method and the graphical Lasso by computing ROC curve. You may use the implementations provided by the package **huge**.

2.4 Application to E. coli regulatory network

Consider the network and expression data found in the **Ecoli.data** dataset from the **sand** package. Symmetrize the network and remove the isolated nodes. Then, infer the network from the expression data. Compare the inferred network to the reference network.

1. Use the package **glmnet** to solve a Lasso problem.