dsc-Omega

1. Motivation

Estimation for covariance matrix and its inverse is widely useful in areas, such as multiple test, discriminant analysis and graphical model. In our multiple test procedure, we need estimation of precision matrix to get calculate the likelihood based on multivariate normal assumption. In the real data, we always meet this situation that sample size is around few hundreds and much smaller than number of variables which is around ten to hundred of thousands. And for Gaussian graphical model, each edge corresponding to one element in the precision matrix representing the conditional correlation. In this case the dimension P is much larger then sample size N. For convariance matrix, a natural estimation is sample covariance matrix.

$$\hat{\Sigma} = \frac{1}{n} X^T X \tag{1}$$

This is singular when N is smaller than P. We don't have capability to use N samples to estimate $\frac{P(P+1)}{2}$ parameters. Some special structures of convariance matrix or precision matrix need to be imposed. Assumptions that makes the degree of freedom of parameters smaller than sample size actually provides constraint on number of unknown parameters to estimate. Then we are able to estimate the covariance matrix.

Different Assumptions

There are many possible assumptions to consider. We just consider two of them which are more reasonable to our data set.In the setting that P is much larger than N, the accuracy of estimation of covariance matrix and precision matrix and the computational cost are challenges in practical studies.

- sparse assumption on precision matrix: Ω
- diagonal matrix + low rank matrix

We denote the covariance matrix as Σ and precision matrix as Ω.

The sparse assumption is popular for Gaussian graphical model and there are many methods based on this assumption. This assumption assumes edges of the undirected graph are sparse, which means that just some of the nodes are connected by edges and many of them are disconnected. As we known,

$$i \longleftrightarrow j \iff \Omega_{ij} \neq 0 \tag{2}$$

which means that node i and node j are connected is equivalent to $\Omega_{ij} \neq 0$ for all i and j. So in Gaussian graphical model, precision matrix encode the graph. To assume the sparsity of graph is equivalent to assume sparsity of Ω. In gene expression data, sparsity assumption in Ω introduce the sparsity of conditional independence among different pair of genes.

In gene expression data, gene can not function alone, instead, genes tend to work together to manifest biological function. Imagine we can observe the latent factor (unmeasured, unobserved, confounding factor) in gene expression data, such as bath effects, latent population structure, biological covariates and transcript factor, then this provides a structure of expression data. Genes are co-regulated by these factors and the influence of factors on all genes are sparse sometimes. Usually we impose sparsity on factor loadings to facilitate kind of clustering information: gene with zero loadings not belongs to the “gene cluster” corresponding to specific factor. The factors are sparse sometimes or dense sometimes.

Here we can also divide these two type of sparsity into two categories: sparsity in edges and sparsity in factors.

Dynamic statistical comparisons

It is hard to say which assumption is more reasonable for our data. We need a comprehensive comparison to different methods on our data to test which one is best to apply. “Dynamic statistical comparisons is an attempt to change the way that researchers perform statistical comparisons of methods. When a new statistical method is developed, it is almost inevitable that it will be useful to compare it to other methods for tackling the same problem. However, the way these comparisons are currently (usually) done is suboptimal in so many ways. First, comparisons are usually performed by the research group that developed one of the methods, which almost inevitably favors that method. Furthermore, performing these kinds of comparisons is incredibly time-consuming, requiring careful familiarization with software implementing the methods, and the creation of pipelines and scripts for running and comparing them.”, as Matthew mentions this project in his github repo, “And in fast-moving fields new methods or software updates appear so frequently that comparisons are out of date before they even appear. In summary, the current system results in a large amount of wasted effort, with multiple groups performing redundant and sub-optimal comparisons. A DSC is a public Internet repository that allows methods to be compared with one another in a reproducible and easily-extensible way.” We will use dsc procedure to compare different methods to estimate covariance matrix and precision matrix.

2. Datamakers

In one dsc object, the main parts are datamakers, methods and scores. Datamakers contain all scenarios considered by users. In our dsc-Omega, we focus on covariance matrix and precision matrix. So the different scenarios are set based on the structure of covariance matrix. There are three scenarios included in our simulations: identity, toeplitz and real data case. Datamakers also provide meta which can be used to compare the performance under different scores. The meta can be the real value or test data set. In this project, we generate simulation data and divide it into two groups: training data and testing data. The training data set is used as input to all methods and testing data is used as meta in comparison based on different scores.

3. Methods

We consider the comparison of methods based on those two types of assumptions we mentioned above. Sparse graph assumption is popular and there are many methods in the literature. We choose some typical methods to compare.

Penalize Likelihood Method

Penalized likelihood method is one of the most widely used approaches. We pick glasso as an example.

- Glasso

$$\hat{\Omega} = argmin\{trace(\Sigma_n \Omega) - \log|\Omega| + \rho ||\Omega||_1\} \tag{3}$$

where $\Sigma_n = \frac{1}{n} X^T X$ which is sample covariance matrix. Glasso converts this problem to multiple lasso problem for each column.

Column by Column Methods

Some methods impose sparsity directly on columns of Ω.

- Clime

$$min : ||\Omega||_1$$

subject to

$$||\Sigma_n \Omega - I||_\infty \leq \lambda_n$$

And then decompose this convex problem into P vector minimization problems.

$$min : ||\Omega_{.j}||_1 \tag{4}$$

subject to

$$|\Sigma_n \Omega_{.j} - e_j|_\infty \leq \lambda_n \tag{5}$$

where e_j is vector with j^{th} element being 1 and others being 0. And then make the estimation symmetric. The authors prove that the symmetric $\hat{\Omega}$ from clime is asymptotically positive definite. Similar idea is graphical Dantzig selector by Yuan(2010).

Other methods estimate Ω using column by column regression. This kind methods are more computationally feasible. Column by column method use the relationship conditional distribution of multivariate normal and linear regression to infer the relationship between elements in precision matrix and partial correlation.

$$X_j = \sum_{i \neq j} \beta_i X_i + V_j \tag{6}$$

$$\beta_i = -\Omega_{ij}/\Omega_{jj};\ V_j \sim N(0, \frac{1}{\Omega_{jj}}) \tag{7}$$

$i, j = 1 \cdots P$, X is the N by P data matrix.

Meinshausen, N. and Buhlmann, P(2006) provide a method using lasso for each column and make it symmetric.

Partial Correlation Screening is also a column by column method, which use partial correlation includes the node one by one with a clean step after.

Tuning Insensitive Methods

The methods mentioned above depend on tuning parameters to control bias variance trade-off. Although they provide theoretical choice of the parameters, it is hard to apply those in real data. It is because theoretical tuning parameters always depend on the true value of Ω.

- Tiger

Which is also a column by column method by asymptotically tuning-free. For each column, it uses SQRT-lasso

$$\beta = argmin\{\frac{1}{n}||y - x\beta||_2 + \lambda||\beta||_1\} \tag{8}$$

which is asymptotically tuning-free. And then make the $\hat{\Omega}$ symmetric.

Robust Methods

These methods extend the distribution to non-Gaussian case. Here we take nonparanormal family as example. For X, there exist functions $\{f_j\}_{j=1}^{P}$ such that

$$(f_1(X_{.1}), \cdots, f_P(X_{.P})) \sim N(\mu, \Sigma)$$

and the authors show that

$$\Sigma_{j,k} = 2\sin(\frac{\pi}{6}\rho_{jk}) = \sin(\frac{\pi}{2}\tau_{jk}) \tag{9}$$

we can use Spearman's rho and Kendall's tau pluged in the formula above. And than we can use Clime or Dantzig selector to find the estimation of Ω

Factor Model

One of the limitation of sparse assumption of precision matrix is that the marginal precision matrix is not sparse when the joint precision matrix is sparse

$$(\Sigma_O)^{-1} = \Omega_O - \Omega_{OH}\Omega_H^{-1}\Omega_{HO}$$

where O stand for observed, H means hidden, and the subscript means the corresponding partition of the Ω matrix.

Instead of starting from the precision matrix itself with sparsity assumption, we can start from the data and try to learn the data well, and then turn to the precision matrix. We first put the data into a low-dimension space which hopefully could capture all the variation information in this low-dimensional space in order to estimate the covariance matrix and precision matrix. We use factor model which is easy to extend in dimension to model the expression level data.

$$X_{n\times P} = F_{n\times K}L_{K\times P} + E_{n\times P} \tag{10}$$

where X is gene expression level matrix, F is factor matrix, L is loading matrix and $E_{.i} \sim N(0, \Psi)$

For individual i:

$$X_{i1} = L_{11}F_{i1} + L_{21}F_{i2} + \cdots + L_{K1}F_{iK} + E_{i1}$$
$$X_{i2} = L_{12}F_{i1} + L_{22}F_{i2} + \cdots + L_{K2}F_{iK} + E_{i2}$$
$$X_{iP} = L_{1P}F_{i1} + L_{2P}F_{i2} + \cdots + L_{KP}F_{iK} + E_{iP}$$

F_{ik} stands for influence of factor k on individual i, $L_{k,j}$ stands for influence of factor k on gene j.

The estimation of covariance matrix is

$$\hat{\Sigma} = \Psi + L^T\Lambda_F L \tag{11}$$

where $\Lambda_F = \frac{1}{n}F^TF$

For the precision matrix:

$$\begin{aligned} \hat{\Omega} &= \hat{\Sigma}^{-1} && (12)\\ &= \Psi^{-1} - \Psi^{-1}L^T(\Lambda_F + L\Psi^{-1}L^T)^{-1}L\Psi^{-1} && (13) \end{aligned}$$

- Both $\hat{\Sigma}$ and $\hat{\Omega}$ are diagonal plus low rank.
- Inverse $\hat{\Sigma}$ is not computationally expensive.
- Model is statistically interpretable.

Fan, J. et al(2008) use this model to estimate the covariance matrix using:

$$\Sigma = L^T cov(F) L + \Sigma_E \tag{14}$$

They assume that Σ_E is diagonal corresponding to "strict factor model". This might be restrictive in practice. The column of data matrix could be also correlated given factors, but the correlations are weak, since the low rank part is just an approximation of data matrix. It is possible to relax the diagonal assumption for Σ_E to non-diagonal or sparse. Fan, j. et al(2015) provide a method to achieve this conditional sparsity (given factors).

- find loadings by solve a regression problem
- apple threshold method on sample covariance matrix of residuals.

Here we need $\hat{\Sigma}$ to be positive definite. There are two ways to make it.

- choose tuning parameters such that the smallest eigenvalue being positive.

In practice we can use C-V.

- "nearPD"

which use

$$min||R^* - A||_F^2$$

such that

$$A > 0, diag(A) = I$$

where R^* is correlation matrix after threshold. But the sparsity can't be guaranteed

- add one more constraint to the objective function

$$\hat{\Sigma} = argmin_{\lambda_{\min}(\Sigma) > \tau}\{||\Sigma - \Sigma_n||_F^2 + P(\sigma_{ij})\}$$

The formula (11) is not identifiable if factor is latent and Σ_E is not diagonal, where some assumption is needed.

There are several methods for latent factor model, such as Robust PCA and Principal Orthogonal complEment Thresholding.

For gene expression data, we know that genes cannot manifest their function alone and genes tend to work together to achieve biological functions instead. Some factors like transcription factor co-regulate a subnetwork of genes. "In molecular biology and genetics, a transcription factor (sometimes called a sequence-specific DNA-binding factor) is a protein that binds to specific DNA sequences, thereby controlling the rate of transcription of genetic information from DNA to messenger RNA. Transcription factors perform this function alone or with other proteins in a complex, by promoting (as an activator), or blocking (as a repressor) the recruitment of RNA polymerase (the enzyme that performs the transcription of genetic information from DNA to RNA) to specific genes" and "Transcription factors bind to either enhancer or promoter regions of DNA adjacent to the genes that they regulate. Depending on the transcription factor, the transcription of the adjacent gene is either up- or down-regulated. Transcription factors use a variety of mechanisms for the regulation of gene expression."- from wikipedia. And other latent factors also work in this way.

The method we use for sparse factor analysis is SFAmix, which assumes that the loadings comes from mixture of dense and sparse prior called three parameters beta distribution.

The hierarchical structure of elements of loading as follow:

$$\begin{aligned}
\gamma &\sim G(f,\nu) && (15)\\
\eta &\sim G(e,\gamma) && (16)\\
\tau_k &\sim G(d,\eta) && (17)\\
\phi_k &\sim G(c,\tau_k) && (18)\\
\delta_{jk} &\sim G(b,\phi_k) && (19)\\
\theta_{jk} &\sim G(a,\delta_{jk}) && (20)\\
l_{jk} &\sim N(0,\theta_{jk}) && (21)
\end{aligned}$$

this is for sparsity of loading. The authors also consider the dense loading:

$$\begin{aligned}
\pi &\sim Beta(\alpha,\beta) && (22)\\
Z_k &\sim Bern(\pi) && (23)\\
l_{jk}|Z_k &\sim \begin{cases} P(l_{jk}|\theta_{jk},\delta_{jk},\phi_k) \\ P(l_jk|\phi_k) = N(0,\phi_k) \end{cases} && (24)
\end{aligned}$$

SParse + Low-rank ASH?

$$X_{n\times P} = F_{n\times K}L_{K\times P} + E_{n\times P} \tag{25}$$

From the woodbury matrix identity:

$$\hat{\Sigma} = \Psi + L^T\Lambda_F L \tag{26}$$

where $\Lambda_F = \frac{1}{n}F^TF$

For the precision matrix:

$$\begin{aligned}
\hat{\Omega} &= \hat{\Sigma}^{-1} && (27)\\
&= \Psi^{-1} - \Psi^{-1}L^T(\Lambda_F + L\Psi^{-1}L^T)^{-1}L\Psi^{-1} && (28)\\
&= \Psi^{-1}_{P\times P} - \Psi^{-1}_{P\times P}L^T_{P\times K}(\Lambda_F + L\Psi^{-1}L^T)^{-1}_{K\times K}L_{K\times P}\Psi^{-1}_{P\times P} && (29)
\end{aligned}$$

We are interested in Ψ^{-1} rather then Ψ itself. We want the inverse of the corvariance matrix is sparse rather than itself. So Ω^{-1} is a sparse plus low rank matrix.

Algorithm

- Step 1:
  Estimate L and F

  using SFAmix FLASH or any sparse factor analysis.

- Step 2:
  $R = X - FL$

  R is the residual matrix Given R: estimate $\hat{\Psi}^{-1}$

  apply estimation precision matrix method (Glasso, Tiger) to residual matrix R

  $\hat{\Psi}^{-1}$ is symmetric positive definite and truly sparse which make Glasso quick enough.

- Step 3:
  plug estimation in $\Omega = \Psi^{-1} - \Psi^{-1}L^T(\Lambda_F + L\Psi^{-1}L^T)^{-1}L\Psi^{-1}$

  the matrix multiplication is quick

The challenge here is make Ω symmetric and positive definite. The symmetric and positive definite properties of $\hat{\Psi}^{-1}$ can fit this requirement. Glasso and other popular work can achieve this when the true Ψ^{-1} is truly sparse.

4. Output

The output here is just the estimation precision matrix from different methods under different scenarios.

5. Scores

All the scores we use are not depend on the true value.

- Likelihood:
  $$log|\Omega| - tr(\Sigma_n \Omega)$$

- F-norm of error:
  $$||\frac{1}{2}\Sigma_n\Omega + \frac{1}{2}\Omega\Sigma_n - I||_F$$

  which is first order condition for Hyvärinen score for Gaussian

- prediction error
  $$\sum_j (X_j - (\sum_{i \neq j} -\Omega_{ij}/\Omega_{jj}X_i))^2$$

  this is for one observation, we have n observation, so just add them up.